Highlighting Root Notes in Chord Recognition using Cepstral Features and Multi-task Learning

Mu-Heng Yang, Li Su, and Yi-Hsuan Yang
Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan
E-mail: dandelion2413@iis.sinica.edu.tw
E-mail: lisu@citi.sinica.edu.tw
E-mail: yang@citi.sinica.edu.tw

***Abstract*—A musical chord is usually described by its root note and the chord type. While a substantial amount of work has been done in the field of music information retrieval (MIR) to automate chord recognition, the role of root notes in this task has seldom received specific attention. In this paper, we present a new approach and empirical studies demonstrating improved accuracy in chord recognition by properly highlighting the information of the root notes. In the signal level, we propose to combine spectral features with features derived from the cepstrum to improve the identification of low pitches, which usually correspond to the root notes. In the model level, we propose a multi-task learning framework based on the neural nets to jointly consider chord recognition and root note recognition in training. We found that the improved accuracy can be attributed to better information about the sub-harmonics of the notes, and the emphasis of root notes in recognizing chords.**

I. Introduction

A musical chord is a collection of three or more notes with specific harmonic relations that is heard as if sounding simultaneously. Many musical performances use chords to provide the harmonic context of the melody. In consequence, automatic chord recognition from audio is an important step in characterizing the content of music.

A chord can usually be defined by the pitch class of its *root note* (which is mostly the lowest note) and the *type* (e.g., major or minor) of the harmonic relations among the notes. To capture such information from the audio, many audio feature representations have been proposed in the literature, the most famous one being the *chromagram* [1], a dimension-reduced feature that aggregates the spectral information of a time-frequency representation into the 12 pitch classes {`C`, `C#`, `D`, ... `B`}. As the chromagram is insensitive to the tone height (i.e., octave number) of the pitches, it is useful in modelling the type of the harmonic relations. Accordingly, the chromagram has been used extensively in chord recognition [1], [2], [3]. To make it further invariant to timbre or tempo variations, techniques that *postprocess* the chromagram have also been proposed [4], [5].

Because the information about the tone height is discarded in the chromagram, however, the ability to resolve the root note will highly depend on the *saliency* of the root note (i.e., relative energy as compared with other pitches in the same time frame) in the given time-frequency representation. If the saliency of the root note is low, it is likely that the pitch classes of other notes can dominate the chromagram, thereby misleading the chord recognizer. To deal with this issue, one may use an additional chromagram that represents the chroma over low frequencies (e.g., MIDI notes 26–49) [6], or use a separate model to first estimate the bass notes and then modify the score for every chord according to the bass pitch [7], [8], [9]. Another existing approach is to decompose the problem and train a bass note recognizer and a chord type recognizer either separately or jointly [10], [11]. All these attempts have reported improved accuracy in chord recognition.

A fundamental issue that is nevertheless seldom addressed in the literature is the inherent limits of a time-frequency representation such as the short-time Fourier transform (STFT) in representing the *low pitches*, which usually correspond to the root notes. We argue in this paper that it is possible to *preprocess* the time-frequency representation to enhance the saliency of the bass notes across time before calculating the chromagram, thereby highlighting the pitch classes of the bass notes in our chroma-based representation.

In a pilot study, we considered an artificial scenario in which the root notes are known in the feature extraction stage (but not in the chord recognition stage) and are employed to double the energy of the corresponding pitch classes in the chromagram. Using such enhanced chromagrams in replacement of the original chromagrams in a standard chord recognition system can already significantly improve the accuracy in disambiguating 24 chords from 66.2% to 77.3%, confirming the potential benefit of highlighting the root notes.

To move the simple idea to something practical and usable, we investigate in this paper both signal processing and machine learning techniques to improve the saliency of root notes, without assuming that the root notes are known in the test data. In the signal level, we propose to combine spectral features derived from a time-frequency representation with features derived from the *cepstrum* to improve the identification of the low pitches. We discuss the rational of applying such cepstral features from the perspectives of both signal processing and music theory, and validate through experiments on their effectiveness in chord recognition. While cepstral features have been employed in other music information retrieval tasks (MIR) [12], [13], its use in chord recognition has been limited to date. In fact, cepstral features are especially suitable for chord recognition because most of the time, bass

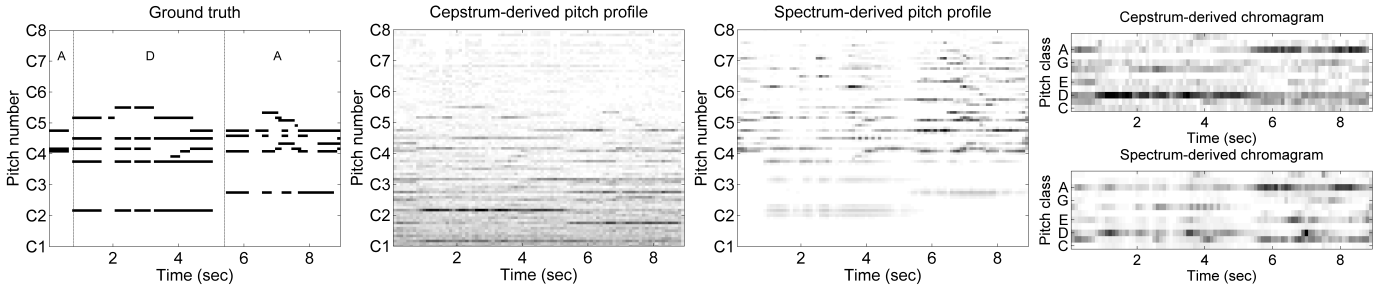


Fig. 1. A 9-second excerpt of piano quintet with chord progression A−maj→D−maj→A−maj: (a) ground truth piano roll, (b) cepstral pitch profile, (c) spectral pitch profile, (d) upper: cepstral chromagram, lower: spectral chromagram.

notes are exactly root notes, meaning that cepstral features can strengthen the energy of root notes and improve the performance of chord recognition.[1] As to those inverted chords [14], whose root notes are not bass notes, cepstral features can still help highlight the root notes because their sub-harmonics tend to focus on the root notes. The detail of harmonics music theory will be illustrated in Section II-B.

In the model level, we propose to use *multi-task learning* (MTL) to jointly consider chord recognition and root note recognition in the training phase. By doing so, the model can learn the relationship between root notes and chords (as also attempted in [10], [11]) and thereby further highlight the root notes while recognizing the chords in the test data. We approach this via multi-task *neural network* of both *shallow* and *deep* structures [15]. The former one is simpler and more transparent, whereas the latter one can be considered as an advanced feature representation learning method and thereby rendering MTL as another feature-level improvement.

In short, our contribution to this field is mainly the introduction of the cepstral features, coined as *cepstral chromagram*, to improve the saliency of the root notes for chord recognition. In what follows, we introduce cepstral chromagram and the rational of using it in Section II. We then present the MTL framework in Section III, the experimental setup in Section IV, and finally evaluations in Section V.

II. Cepstral Chromagram

Spectral features, carrying the *frequency* information, has long been regarded as the predominant way for constructing the chromagram [2], [3]. However, it has been known that the frequency information is not the only source from which pitches can be detected. Rather, features representing *periodicity*, such as the autocorrelation function (ACF) or the cepstrum, have been shown useful for both single-pitch and multi-pitch detection [12], [13], [16], [17], [18].[2]

[1]The bass notes are the notes with the lowest pitch, while the root notes are the notes where other notes derived from. If the bass note of a chord is not the root note, it is called a inverted chord. For example, a normal C−maj chord consists of C, E and G in ascending order. However, the first inversion of a C−maj chord would be E, G and C. Similarly, the second inversion of a C−maj chord would be texttttG, C and E.

[2]The famous Mel-frequency cepstral coefficients (MFCC) can also be viewed as a cepstrum-derived feature but it characterizes timbre rather than periodicity information.

The *real cepstrum* is defined as the inverse discrete Fourier transform (DFT) of a log-magnitude spectrum:

$$\mathcal{F}^{-1}(\log(|\mathcal{F}(\mathbf{x})|))\,, \tag{1}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the DFT and its inverse, respectively. While the spectrum indicates the salience of the fundamental frequencies (F0's) and their multiples (i.e., *harmonics*) in the frequency domain, the cepstrum indicates the fundamental periods (i.e., the inverse of F0's) and their multiples (i.e., *sub-harmonics*) in the lag domain.

Given an input audio signal, we can compute the magnitude spectrum and the real cepstrum from its STFT, and then collect the energy corresponding to each semitone to respectively generate the spectral and cepstral *pitch profiles* for each frame. Pitch profiles are further folded into 12-dimensional *chroma vectors* (a.k.a. pitch class profiles) and normalized to zero mean and unit variance. We refer to the resulting feature representations over time as the *spectral chromagram* (i.e., the usual chromagram) and *cepstral chromagram*, respectively.

A. Its Potentials – Signal Processing Perspective

To demonstrate the potential strength of the cepstral chromagram for chord recognition, we show in Figure 1 the groundtruth piano roll of an audio excerpt and the spectral and cepstral features extracted from the audio. The excerpt is a piano quintet in D−maj key featuring a chord progression of A−maj, D−maj and A−maj. The following observations can be made.

First, from Figures 1(b) and (c), we see that the cepstral and spectral pitch profiles have salient sub-harmonic (i.e., multiple fractions of the F0's) and harmonic components, respectively. The energy distributions in the two profiles are quite different and complementary to each other. For example, due to the insufficient window length for STFT and their weaker energy, we can hardly observe the D2 (1–5.3 seconds) and A2 (5.3–9 seconds) played by the double bass in the spectral pitch profile. However, they are fairly salient in the cepstral pitch profile.

To see how a cepstrum-derived representation can enhance the low-frequency notes whose F0's are the common divisor of the concurrent notes, we take a look at the notes from 0.8 to 1.6 seconds (i.e., D2+A3+D4+F#4+D5) and the notes from 5.4 to 6.1 seconds (A2+C#4+G4+A4). For the first cluster of

	C-maj			C-min		
5th harmonics	E	G#	**B**	E	G	**B**
3rd harmonics	G	**B**	**D**	G	A#	**D**
fundamental tones	**C**	E	G	**C**	D#	G
3rd sub-harmonics	F	A	**C**	F	G#	**C**
5th sub-harmonics	G#	**C**	D#	G#	B	D#

TABLE I
THE PITCH CLASSES OF THE HARMONICS AND SUB-HARMONICS OF C-MAJ (LEFT) AND C-MIN CHORD (RIGHT).

notes, the greatest common divisor of the F0's of the notes is the pitch D2, while for the second cluster it is A1, both of which are salient in the cepstral pitch profile and can therefore correctly reflect the root notes of the underlying chords (i.e., D and A).

Finally, from Figure 1(d), we can also see that the bass notes contribute a lot to the cepstral chromagram, but not to the spectral chromagram.

Because the cepstrum is derived from the spectrum, we can say that the cepstral chromagram is a modified version of the spectral chromagram by pre-processing the STFT time-frequency representation before summing up the pitch coefficients per chroma.

B. Its Potentials – Music Theory Perspective

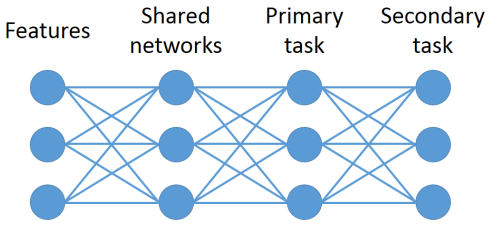
The potential usage of cepstral features in chord recognition can also be discussed from a music-theoretical perspective. Table I shows the pitch classes of the 3rd and 5th harmonics and sub-harmonics of a C-maj chord (C+E+G) and an C-min chord (C+D#+G).[3] From the left hand side, we see that the sub-harmonics tend to highlight the pitch class of the root note (i.e., the bold C), whereas the harmonics tend to enhance the pitch classes of the major 7th and the major 2nd (i.e., the bold B and D), which are not related to the underlying chord. In consequence, using a spectral chromagram to recognize the C-maj chord can be error-prone, if the 3rd and 5th harmonics are strong for the particular instrument(s) playing the notes. In such a case, the C-maj chord can be mistaken as E-min or G-maj, since we may observe salient energy in D, E, G and B in the spectral chromagram. Such a confusion is less likely to happen when using the cepstral chromagram for chord recognition. Similar observation can be found for various types of chords, and extended to all kinds of root notes.

Note that the sub-harmonics introduced by the cepstrum are not free from side effects. For example, a C-maj chord may be mistaken as a F-maj chord because of the 3rd sub-harmonic. However, we have to take into account the energy contributed to the root note. Since F is not in the fundamental tones, simply one 3rd sub-harmonics is not strong enough to disguise as the root note.

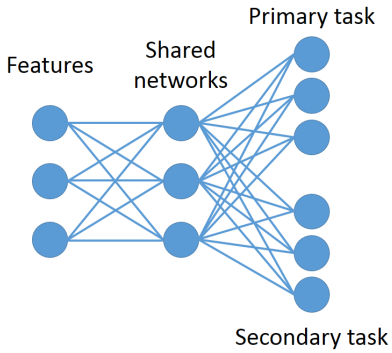
For spectrum, the harmonics often lead to confusion, whose root notes are in the fundamental tones, such as mislabelling C-maj as E-min, C-maj as G-maj, or C-min as D#-maj. Since E, G and D# are already in fundamental tones, their energy will be very salient that we can hardly tell C is the true root note.

In contrast, the cepstrum tends to emphasize the pitch classes of the root notes, thereby avoiding such confusions. Although some confusions may be brought up from the 3rd or 5th sub-harmonic in the cepstral representation, their energy may not be strong enough to be harmful since they are not in the fundamental tones.

[3]Please note that the 1st (sub)harmonic correspond to the fundamental tone, and the 2nd and 4th (sub)harmonics are simply octaves apart from the fundamental tone. Therefore, we do not list them in the table.



(a) series architecture



(b) parallel architecture

Fig. 2. Two possible multi-task learning architectures. The input features can be the 12-dimensional spectral chroma vector, the 12-dimensional cepstral chroma vector, or the concatenation of them. The primary task is chord recognition, whereas the secondary task can be root note recognition or others.

III. MULTI-TASK LEARNING

Multi-task learning (MTL) has been proposed to learn more than one related task at a time [15], [19], [20]. The model is trained to perform both the *primary* task and the *secondary* task using a shared parameters, and may therefore generalize better to unseen data due to the shared structure of the tasks. For cases where the labeled data for the primary task is scarce, MTL can be employed to take advantage of the labeled data of the related tasks. Moreover, adopting the MTL framework will not increase the runtime in the testing phase, because the additional parameters for the secondary task is only used to optimize the shared parameters while training. After the training phase, we can discard the parameters associated with the secondary task, reducing the model to be a single-task one.

MTL can be applied to many MIR tasks because different attributes of music (e.g., chord, key, or metric position) have various hierarchical or peer relationships. A chord can

be represented by a one-hot vector, the corresponding ideal chroma representation, or by breaking it into the root note and the chord type. By adding different forms of output labels as the secondary tasks, MTL can possibly learn signal-level interrelationships or music-theoretical bindings among the attributes, thereby improving the performance of the primary task. Such an idea has been pursued in prior work, using for the example a multi-chain hidden Markov model (HMM) [11] or a dynamic Bayesian network [10].

In this paper, we use a neural nets based MTL framework for chord recognition. Specifically, using for example root note recognition as the secondary task, we investigate two possible network architectures for MTL. The first one is a series architecture (Figure 2(a)), where the secondary task is stacked after the primary one. During back propagation for training the network parameters, the errors of the secondary task can influence the weights associated with the primary task. The second one is a parallel architecture (Figure 2(b)), where the primary and secondary tasks are in parallel. The errors of the secondary task will only modify the weights in the shared network, leaving some weights for the primary task undisturbed.

We prefer to use a neural nets based framework for it is possible to add multiple hidden layers to the shared network to optimize the feature representation in a data-driven way, which has been shown effective in many tasks, including chord recognition [21]. We consider neural nets of both shallow structure (i.e., only one layer for the shared network, as the one shown in Figure 2) and deep structure in our evaluations.

IV. Experimental Setup

A. Dataset, Chord Labels & Performance Measures

We evaluate the proposed ideas by using the well-known Beatles dataset, which comprises 180 Beatles songs with groundtruth chord labels annotated by Harte [22], [23]. For evaluation, we adopt 5-fold cross validation and randomly select 144 songs for training and 36 songs for testing each time. The cross validation process is repeated for 10 times to get the average result.

Following previous work [24], we consider only the major and minor triad chords with 12 different root notes in this work, plus one additional 'non-chord' label `N`, making it in total 25 classes. Chord types other than major and minor (e.g., augmented, diminished, and suspended) are considered as non-chords, and the seventh note in a major or minor seventh chords will be discarded to map for example a `D-maj7` to `D-maj`. A time frame will be labeled as `N` as well when there is no chord or even no sound at all.

The accuracy of chord recognition is measured by the weighted average overlap ratio (WAOR) [25], which indicates the average recognition rate over all the songs in the test set, weighted by the length of each song. From the result of chord recognition, we can also evaluate how accurate the root notes are predicted, disregarding the chord types. In this way, we can compare the accuracy of chord recognition and root note recognition, and examine whether an improved chord recognition rate is due to the enhanced root notes.

B. Feature extraction

Given an audio signal, we first downsample it to 11,025 Hz, and then compute the STFT with half-overlapping windows of 2,048 samples (i.e., 0.2 second). From the STFT, we compute the magnitude spectrum and the real cepstrum, as described in Section 2, and then accumulate the amplitudes of them around each of the semitones ranging from `A0` to `C7`, generating 76-dimensional spectral and cepstral pitch profiles respectively. Pitch profiles are further folded into 12-dimensional chroma vectors and normalized by z-score. As we need to predict the chord label for each time frame, the input to our chord recognizer is these frame-level feature vectors, instead of the whole chromagrams.

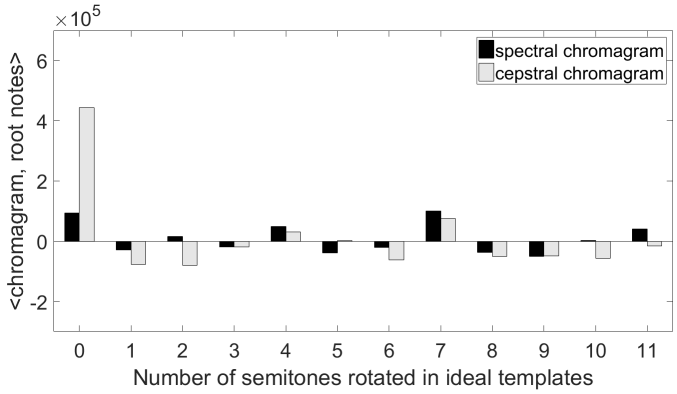
C. Neural Network & Hidden Markov Model

Unless otherwise specified, we use one layer neural network (i.e., a shallow structure) with only 64 neurons for most experiments. Although it may not achieve high accuracy, it is a fairly simple classifier that prevents overfitting. Therefore, the performance comparison of different features can be done without the interference from the complex models. Specifically, we use rectified linear unit (ReLU) as the activation function, cross entropy as the cost function, and the softmax function for the output layer. Due to the softmax function, the range of our output will be in [0,1], which can be regarded as the probability of observing each chord in a given time frame. We finally use the observed probabilities in a typical HMM for post processing. Due to some transient noise or percussions, the features in certain time frames are corrupted. By using HMM, steady chord transition can be assured [2], [3], and those errors can be corrected.

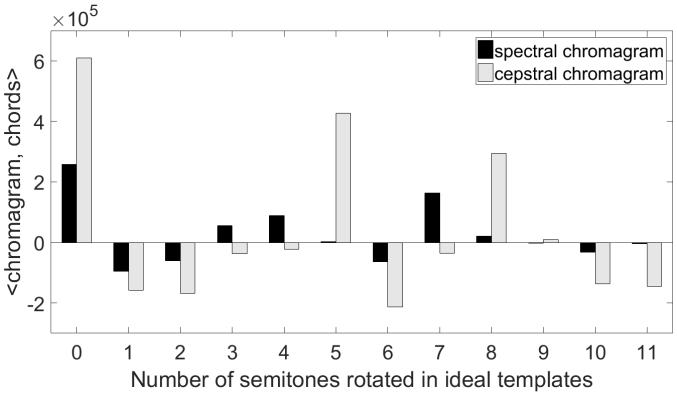
In the end of our evaluation, we will consider the case of deep neural nets for model learning, with two hidden layers, to see whether this can lead to better result.

D. Combining Spectral and Cepstral Features

As the spectral and cepstral chroma features may be complementary to each other, in addition to comparing their performance we are also interested in combining them for chord recognition. We consider two classic fusion strategies: feature concatenation is an *early-fusion* method that combines the two feature representations before feeding them into the classifiers, whereas ensemble is a *late-fusion* method that use the two given feature representations to train two classifiers independently, after which the result of the two classifiers are combined by taking the average. In our implementation, feature concatenation allows a neuron network to learn the interactions between the two given features, whereas ensemble exploits the information embedded in the two features separately and combine them later. In case of feature concatenation, the input to our neural nets will be 24-dimensional feature vectors.



(a) inner product with templates of root notes



(b) inner product with templates of chords

Fig. 3. Accumulated inner product between the spectral (black) and cepstral (light gray) chroma vectors computed from audio and the template chroma vectors for different (a) root notes and (b) chords for the Beatles dataset.

V. Experiments and Discussion

In what follows, we first present two pilot studies supporting the proposed ideas in synthetic experimental settings. After that, we build chord recognizers using different feature representations (spectral, cepstral or fusion) and neural network architectures (single-task, series MTL, or parallel MTL) to investigate the practical gain of using the cepstral chromagram.

Although a large number of features and models have been proposed in the literature for chord recognition [2], [3], our evaluation focuses only on the simplistic setting of comparing the standard STFT-based spectral chromagram and the proposed cepstral chromagram. The goal is not to "beat" existing methods, but rather to show that spectral chromagram is not the only feature representation people can use in their chord recognizers.

A. Pilot Studies

As mentioned in Section I, in a pilot study we consider an artificial scenario in which the root notes are known in the feature extraction stage to double the energy of the corresponding pitch classes in the standard spectral chromagram. Using such enhanced chromagrams as input to the single-task neural network yields 77.3% accuracy in chord recognition for the Beatles dataset, which is 11.1% higher than the case where we use the original chromagrams as the input. The accuracy for root note recognition also significantly improves from 71.2% to 85.1%. These results indicate that enhancing the saliency of the root notes has the potential to help chord recognition.

chords and root notes	ideal templates
C	[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
C#	[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
D	[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
C-maj	[1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0]
C#-maj	[0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0]
D-maj	[0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0]
C-min	[1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0]
C#-min	[0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0]
D-min	[0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0]

TABLE II
IDEAL TEMPLATES OF DIFFERENT CHORDS AND ROOT NOTES.

The second pilot study examines whether the cepstral chroma vectors computed from audio can better match the ideal (theoretical) binary templates of the chroma vectors. This study is therefore done in the feature-level using the Beatles dataset, without really training a classifier. By excluding the factor of classifiers, including fitting majority classes, randomness, and over-fitting, we can better analyze the error distribution, and verify the statements in Section II.[4]

Specifically, we build the ideal templates of chroma vectors for different chords and root notes. As Table II shows, the template for a C-maj chord is $[1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0]$, and for a root note C is simply $[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$. Templates for other root notes are generated by shifting them circularly from left to right. We then frame by frame compute the inner product between the templates and the spectral or cepstral chroma vectors extracted from audio. A large inner product indicates that the features extracted matches the ideal template better.

The values of the inner product are accumulated over the Beatles dataset, and the resulting histograms for root notes and chords are shown in Figures 3(a) and (b), respectively.[5]

The x-axes of the figures are indexed by the number of semitones rotated from left to right in ideal templates. Template with 0 semitones rotated is the same as shown in Table II. Template with 1 rotations will make root note C become $[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$, D-maj become $[0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0]$, and so on. Therefore, we should expect high values for the one with 0 semitones rotated. We can see that cepstral chroma vectors outperform spectral ones in both 3(a) and (b).

By analyzing the results of rotated templates, we can get lots of insights about music theory and cepstral features. For example, a high value in template with 7 semitones rotated means that chroma vector of I chord resemble the ideal template of V chord. This can serve as a form of error analysis

[4]Note that the second pilot study is different from the first pilot study. The energy is not artificially doubled in second pilot study.

[5]Note that the chroma vectors are normalized to zero mean and unit variance. Therefore, there are some negative values of inner product.

like confusion table. In addition, the observations here match perfectly with the music theory perspectives in Section II-B.

From Figure 3(a), we see that the magnitude of the root note (0 semitones rotated), major 3rd (4 semitones rotated) and perfect 5th (7 semitones rotated) are relatively higher, mainly because these three notes constitute a major chord. We also see higher magnitude for the root notes for the cepstral features than spectral features, as we have expected. Moreover, the magnitudes for unwanted harmonics component major 2nd (2 semitones rotated) and major 7th (11 semitones rotated) are high for the spectral features.

From Figure 3(b), we can also see that the proposed cepstral features can better match the ideal templates of the correct chords (i.e., 0 semitones rotated). For confusions, spectral features are more likely to confuse a chord `I` with its `V` chord (7 semitones rotated) and `III` chord (4 semitones rotated), as can be guessed from Figure 3(a), whereas cepstral features will more easily confuse `I` with its `IV` chord and `V#` chord. It might be a good idea to fuse the two features as they seem to be complementary.

B. Chord Recognition in Practical Settings

Table III shows the accuracy of chord and root note recognition using different features and learning models. From the result of the first four rows, we see that cepstral features outperform spectral features by 4.1% in chord recognition and 5.0% in root note recognition. Combining the two features via early fusion can further improve the accuracy to 72.8% for chord recognition and to 78.1% for root note recognition. In comparison, late fusion cannot effectively exploit the interaction among the two feature representations. These results demonstrate the effectiveness of the proposed cepstral chroma representation and its ability to complement spectral features.

Rows 5–10 of Table III display the result when we use MTL as the learning model, considering chord recognition as the primary task and root note recognition as the secondary one. When we fix the input feature to either spectral or cepstral features, it seems that the series MTL architecture can more effectively benefit from the secondary task, improving the accuracy of chord recognition by 1.6%. The parallel MTL does not bring gains, possibly because the original input features are already relevant to the two tasks and there is little room for the framework to modify one for the other. Another practical challenge in using the parallel MTL architecture is to properly tune the weights for the two tasks, otherwise one of them may overkill the other, but in our work we simply set equal weights for the primary and secondary tasks. Finally, using the 24-dimensional combined spectral and cepstral chroma features (i.e., early fusion) in the series MTL model can further improve the accuracy of chord recognition to 73.9%, which is 1.1% higher than that achieved by the best single-task model, as shown in row 9 of Table III.

To gain more insights, we show in Figure 4 the distributions of chord recognition errors using different methods, all using shallow neural nets. The errors are counted with respect to the true chords, breaking down according to major or minor chords. For example, if a `C-maj` is mistaken as a `C-min` it will be counted in the '`i`' bin in Figure 4(a). We see that, when cepstral features are used, the recognition rate of major chords (i.e., `I` in Figure 4(a)) is improved by 5.1%, and the recognition rate of minor chords (i.e., `i` in Figure 4(b)) is increased by 6.8%. We can see that cepstral features have fewer errors in most transitions, except for the transition from `i` to `I`. Figure 4(b) also shows that the errors of mis-labeling `i` as `I` can be reduced by combining spectral and cepstral features.

Feature	Model	Chord	Root
spectral	single-task	66.2%	71.2%
cepstral	single-task	70.3%	76.2%
early fusion	single-task	**72.8%**	**78.1%**
late fusion	single-task	69.6%	75.2%
spectral	series MTL	67.8%	72.8%
spectral	parallel MTL	66.3%	71.4%
cepstral	series MTL	71.5%	77.5%
cepstral	parallel MTL	70.3%	76.3%
early fusion	series MTL	**73.9%**	**79.3%**
early fusion	parallel MTL	72.5%	78.1%
spectral	single-task & deep	71.9%	75.4%
early fusion	series MTL & deep	**78.3%**	**81.6%**

TABLE III
CHORD AND ROOT NOTE RECOGNITION ACCURACY USING DIFFERENT FEATURES AND LEARNING MODELS. WE USE BOLD FONTS TO INDICATE THE BETTER RESULTS.

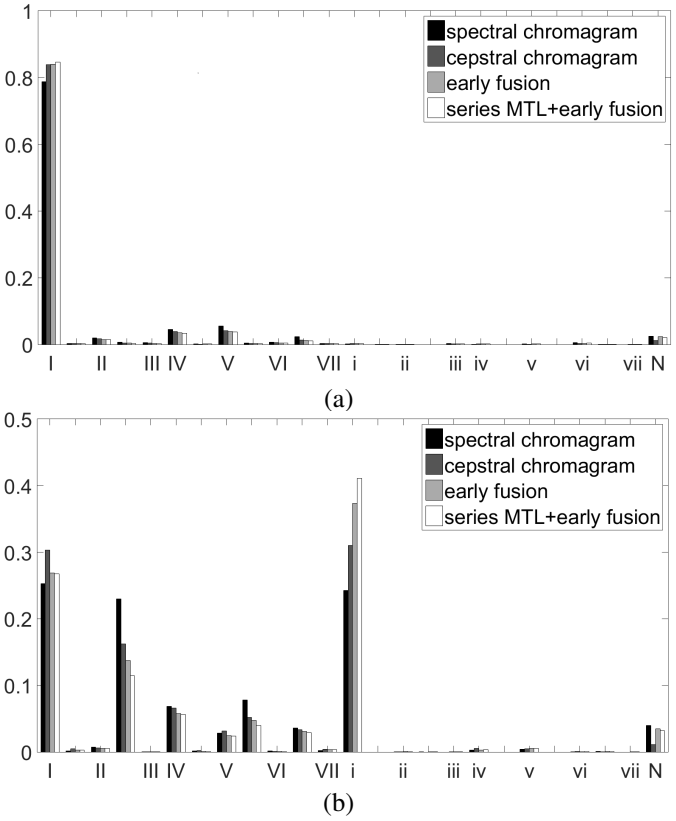


Fig. 4. Distribution of chord recognition errors for (a) major chords and (b) minor chords by different methods, all using shallow structure neural nets.

Finally, we consider a more advanced setting where we use two layers neural nets (i.e., a deep one), dropout rate of 0.7, 3rd order time splicing for capturing temporal contexts [26], and 1,024 neurons per hidden layer. From the last two rows of Table III, we can see that using the combined spectral and cepstral features with this model can further lead to 78.3% accuracy of chord recognition. The accuracy of root note recognition is also improved.

VI. Related work

There are some previous works that attempt to capture notes to improve the accuracy of chord recognition. For example, Yoshioka *et al*. [7] use different weightings for chord notes and non-chord notes. Ni *et al*. [11] use bass chroma to detect the bass notes. Mauch [10] captures the temporal bass line by using dynamic Bayesian networks which can learn the relations among root note, chord, key, and metric position. However, none of the approaches, to our best of knowledge, have explored the use of cepstral features.

VII. Conclusion and future work

In this paper, we have presented an application of the cepstral features in chord recognition. The rational of using the cepstral features was first discussed from theoretical viewpoints and then validated in both synthetic and practical settings. From the evaluation result using the Beatles dataset, we have shown the superiority of the cepstral chromagram over the spectral chromagram, and also the advantage of fusing the two complementary types of features. Moreover, we proposed a multi-task learning framework based on the neural nets to highlight the root notes for chord recognition, finding that a series architecture can perform better than a parallel architecture. From the error analysis, we have shown that the root notes are significantly enhanced by the proposed methods, and this in turn helps improve chord recognition.

Future work can be directed towards investigating advanced deep neural network algorithms, especially using the raw spectral or cepstral features instead of the chroma vectors as inputs to the neural nets. Following the idea of MTL, other related tasks such as multi-pitch estimation or key detection can be jointly considered. We are also interested in considering more chord types and datasets in our evaluation, and rigorously comparing our method with existing ones. The effect of chord inversion can also be an interesting subject for future research.

References

[1] T. Fujishima, "Real time chord recognition of musical sound: a system using Common Lisp Music," *Proc. Int. Computer Music Conf.*, pp. 464–467, 1999.

[2] T. Cho, and J. P. Bello, "On the Relative Importance of Individual Components of Chord Recognition Systems," *IEEE Trans. Audio, Speech, Lang. Proc.*, vol. 22, no. 2, pp. 477–492, 2014.

[3] M. McVicar, R. Santos-Rodriguez, Y. Ni, and T. De Bie, "Automatic Chord Estimation from Audio: A Review of the State of the Art," *IEEE Trans. Audio, Speech, Lang. Proc.*, vol. 22, no. 2, pp. 556–575, 2014.

[4] M. Müller, and S. Ewert, "Chroma Toolbox: MATLAB implementations for extracting variants of chroma-based audio features," *Proc. Int. Soc. Music Information Retrieval Conf.*, Miami, USA, 2011.

[5] D. Ellis, "Identifying 'Cover Songs' with Beat-Synchronous Chroma Features," *Proc. Music Information Retrieval Evaluation eXchange*, 2006.

[6] M. P. Ryynänen, and A. Klapuri, "Automatic Transcription of Melody, Bass Line, and Chords in Polyphonic Music," *Computer Music Journal*, vol. 32, no. 3, pp. 72–86, 2008.

[7] T. Yoshioka, T. Kitahara, K. Komatani, T. Ogata, and H. G. Okuno, "Automatic chord transcription with concurrent recognition of chord symbols and boundaries," *Proc. Int. Soc. Music Information Retrieval Conf.*, pp. 100–105, 2004.

[8] K. Sumi, K. Itoyama, K. Yoshii, K. Komatani, T. Ogata, and H.G. Okuno, "Automatic chord recognition based on probabilistic integration of chord transition and bass pitch estimation," *Proc. Int. Soc. Music Information Retrieval Conf.*, 2008.

[9] M. Mauch, and S. Dixon, "A Discrete Mixture Model for Chord Labelling," *Proc. Int. Soc. Music Information Retrieval Conf.*, pp. 45–50, 2008.

[10] M. Mauch, "Automatic Chord Transcription from Audio Using Computational Models of Musical Context," PhD thesis, University of London, Queen Mary, 2010.

[11] Y. Ni, M. Mcvicar, R. Santos-Rodriguez, and T. De Bie, "An End-to-End Machine Learning System for Harmonic Analysis of Music," *IEEE Trans. Audio, Speech, Lang. Proc.*, vol. 20, no. 6, pp. 1771–1782, 2012.

[12] A. De Cheveigné, and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[13] G. Peeters, "Music pitch representation by periodicity measures based on combined temporal and spectral representations," *Proc. IEEE Int. Conf. Acoust. Speech Signal Proc.*, 2006.

[14] M. Mauch, and S. Dixon, "Symbolic Representation of Musical Chords: A Proposed Syntax for Text Annotations," *IEEE Trans. Audio, Speech, Lang. Proc.*, vol. 18, no. 6, pp. 1280–1289 2010.

[15] X. Liu, J. Gao, X. He, L. Deng, K. Duh, and Y.-Y. Wang, "Representation Learning Using Multi-Task Deep Neural Networks for Semantic Classification and Information Retrieval," *Proc. Conf. the North American Chapter of the Association for Computational Linguistics-Human Language Technologies*, 2015.

[16] L. Su, and Y.-H. Yang, "Combining spectral and temporal representations for multipitch estimation of polyphonic music," *IEEE Trans. Audio, Speech, Lang. Proc.*, vol. 23, no. 10, pp. 1600–1612, 2015.

[17] L. Rabiner, M. J. Cheng, A. E. Rosenberg, C. McGonegal, and others, "A comparative performance study of several pitch detection algorithms," *Proc. IEEE Int. Conf. Acoust. Speech Signal Proc.*, vol. 24, no. 5, pp. 399–418, 1976.

[18] T. Tolonen, and M. Karjalainen, "A computationally efficient multipitch analysis model," *IEEE Speech Audio Processing*, vol. 8, no. 6, pp. 708–716, 2000.

[19] R. K. Ando, and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *J. Machine Learning Research*, no. 6, pp. 1817–1853, 2005.

[20] T. Evgeniou, C. A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *J. Machine Learning Research*, no. 6, pp. 615–637, 2005.

[21] E. J. Humphrey, T. Cho, and J. P. Bello, "Learning a robust tonnetz-space transform for automatic chord recognition," *Proc. IEEE. Int. Conf. Acoustics, Speech & Signal Processing*, pp. 453–456, 2012.

[22] C. Harte, "Towards automatic extraction of harmony information from music signals," PhD thesis, Department of Electronic Engineering, Queen Mary, University of London, 2010.

[23] C. Harte, M. B. Sandler, S. A. Abdallah, and E. Gómez, "Symbolic Representation of Musical Chords: A Proposed Syntax for Text Annotations.," *Proc. Int. Soc. Music Information Retrieval Conf.*, vol. 5, pp. 66–71, 2005.

[24] N. Steenbergen, "Chord Recognition with Stacked Denoising Autoencoders," PhD thesis, University of Amstersdam, 2014.

[25] S. Sigtia, N. Boulanger-Lewandowski, and S. Dixon, "Audio Chord Recognition with a Hybrid Recurrent Neural Network," *Proc. Int. Soc. Music Information Retrieval Conf.,* 2015.

[26] X.-Q. Zhou, and A. Lerch, "Chord Detection Using Deep Learning.," *Proc. Int. Soc. Music Information Retrieval Conf.*, pp. 52–58, 2015.