Published in the Proceedings of the 2007 ISMIR.

A UNIFIED SYSTEM FOR CHORD TRANSCRIPTION AND KEY EXTRACTION USING HIDDEN MARKOV MODELS

Kyogu Lee
Center for Computer Research in Music and Acoustics
Stanford University
kglee@ccrma.stanford.edu

Malcolm Slaney
Yahoo! Research
Sunnyvale, CA94089
malcolm@ieee.org

ABSTRACT

A new approach for acoustic chord transcription and key extraction is presented. We use a novel method of acquiring a large set of labeled training data for automatic key/chord recognition from the raw audio without the enormously laborious process of manual annotation. To this end, we first perform harmonic analysis on symbolic data to extract the key information and the chord labels with precise segment boundaries. In parallel, we synthesize audio from the same symbolic data whose harmonic progression are in perfect alignment with the automatically generated annotations. We then estimate the model parameters directly from the labeled training data, and build 24 key-specific hidden Markov models for 24 different keys. The experimental results show that the proposed model not only successfully estimates the key, but also yields higher chord recognition accuracy than a universal, key-independent model.

1 INTRODUCTION

A musical key and a chord are among important attributes of Western tonal music. A key defines a referential point or a tonal center upon which other musical phenomena such as melody, harmony, and cadence are arranged. Particularly, a key and succession of chords over time, or chord progression, based on the key forms the core of harmony in a piece of music. Hence analyzing the overall harmonic structure of a musical piece often starts with labeling every chord at every beat or measure based on the key.

Finding the key and labeling the chords automatically from audio are of great use for those who want to do harmonic analysis of music. Once the harmonic content of a piece is known, a sequence of chords can be used for further higher-level structural analysis where themes, phrases or forms can be defined.

Chord sequences and the timing of chord boundaries are also a compact and robust mid-level representation of musical signals, and have many potential applications such as music identification, music segmentation, music similarity finding, audio summarization, and mood classification. Chord sequences have been successfully used as a front end to the audio cover song identification system [1]. For these reasons and others, automatic chord recognition has recently attracted a number of researchers in the Music Information Retrieval field. Some systems use a simple pattern matching algorithm [2, 3, 4] while others use more sophisticated machine learning techniques such as hidden Markov models or Support Vector Machines [5, 6, 7, 8, 9, 10].

Hidden Markov models (HMMs) are very successful for speech recognition, whose high performance is largely attributed to gigantic databases with labels have been accumulated over decades. Such a huge database not only helps estimate the model parameters appropriately, but also enables researchers to build richer models, resulting in better performance. However, there is very few such database available for music. Furthermore, the acoustical variance in music is far greater than that in speech in terms of its frequency range, timbre due to different instrumentations, dynamics, and/or duration, and thus a even more data is needed to build generalized models.

It is very difficult to obtain a large set of training data for music, however. First of all, it is very hard for researchers to acquire a large collection of music. Secondly, hand-labeling the chord boundaries in a number of recordings is not only an extremely time consuming and tedious task but also is subject to errors made by humans.

In this paper, we propose a novel method of automating the extremely laborious task of obtaining labeled training data for supervised learning algorithm. To this end, we use symbolic data such as MIDI files to generate chord names and precise time boundaries, as well as to create audio. Audio and chord-boundary information generated from symbolic files are in perfect alignment, and we can use them to directly estimate the model parameters. In doing so, we build 24 key-specific models, one for each for 24 major/minor keys. The training process is illustrated in Figure 1.

Once we trained 24 key-specific HMMs, key estimation is done by selecting the key model with the highest probability given the observation sequence of input audio;

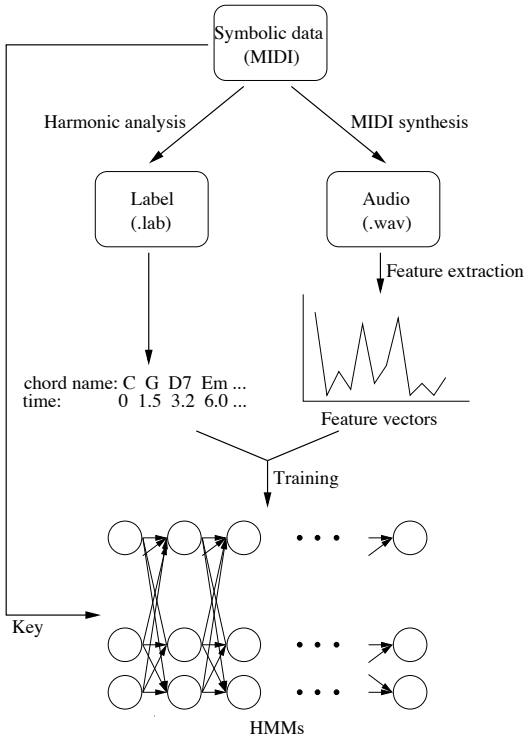


Figure 1. Training stage of the system.

i.e.,

$$key = \underset{k}{\operatorname{argmax}} \Pr(O, Q|\lambda_k), \tag{1}$$

where key is an estimated key, O is an observation sequence, and λ_k is a key model for key k.

Once the key model is selected from Equation 1, we can obtain the chord sequence by taking the optimal state path $Q_{OPT} = Q_1 O_2 \cdots Q_T$ returned by the Viterbi decoder. The overall system for key estimation and chord recognition is shown in Figure 2.

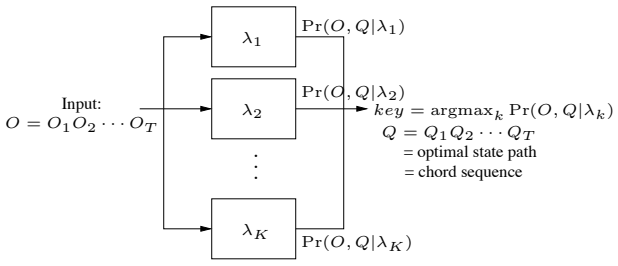


Figure 2. System for key estimation and chord recognition.

There are several advantages to this approach. First, a great number of symbolic files are freely available. Second, we do not need to manually annotate chord boundaries with chord names to obtain training data. Third, we can generate as much data as needed with the same symbolic files but with different musical attributes by changing instrumentation, tempo, or dynamics when synthesizing audio. This helps avoid overfitting the models to a specific type of music. Fourth, sufficient training data enables us to build a model for each key, which not only results in increased performance for chord recognition but also provides key information.

This paper is organized as follows. A review of related work is presented in Section 2; in Section 3, we explain the method of obtaining the labeled training data, and describe the procedure of building our models and the feature set we used to represent the state observation in the models; in Section 4, we present empirical results with discussions, and draw conclusions followed by directions for future work in Section 5.

2 RELATED WORK

Several systems have been previously described for chord recognition from the raw audio waveform. Sheh and Ellis proposed a statistical learning method for chord segmentation and recognition using the chroma features [5]. They used the hidden Markov models (HMMs) trained by the Expectation-Maximization (EM) algorithm, and treated the chord labels as hidden values within the EM framework. In training the models, they used only the chord sequence as an input to the models, and applied the forward-backward algorithm to estimate the model parameters. The frame accuracy they obtained was about 76% for segmentation and about 22% for recognition, respectively. The poor performance for recognition may be due to insufficient training data compared with a large set of classes (20 songs for 147 chord types). It is also possible that the flat-start initialization of training data yields incorrect chord boundaries resulting in poor parameter estimates.

Bello and Pickens also used the chroma features and HMMs with the EM algorithm to find the crude transition probability matrix for each input [6]. What was novel in their approach was that they incorporated musical knowledge into the models by defining a state transition matrix based on the key distance in a circle of fifths, and avoided random initialization of a mean vector and a covariance matrix of observation distribution. In addition, in training the model's parameter, they selectively updated the parameters of interest on the assumption that a chord template or distribution is almost universal regardless of the type of music, thus disallowing adjustment of distribution parameters. The accuracy thus obtained was about 75% using beat-synchronous segmentation with a smaller set of chord types (24 major/minor triads only). In particular, they argued that the accuracy increased by as much as 32% when the adjustment of the observation distribution parameters is prohibited.

The present paper expands our previous work on chord recognition [8, 9, 10]. It is based on the work of Sheh and Ellis or Bello and Pickens in that the states in the HMM represent chord types, and we try to find the optimal state path, *i.e.*, the most probable chord sequence in a maximum-likelihood sense. The most prominent difference in our approach is, however, that we use labeled training data from which model parameters can be directly

estimated without using an EM algorithm. In addition, we propose a method to automatically obtain a large set of labeled training data, removing the problematic and time consuming task of manual annotation of precise chord boundaries with chord names. Furthermore, this large data set allows us to build key-specific HMMs, and we can obtain at the same time the optimal chord sequence and the most probable key of input audio by running a Viterbi decoder.

3 SYSTEM

Our chord transcription system starts off by performing harmonic analysis on symbolic data to obtain label files with chord names and precise time boundaries. In parallel, we synthesize the audio files with the same symbolic files using a sample-based synthesizer. We then extract appropriate feature vectors from audio which are in perfect sync with the labels, and use them to train our models.

3.1 Obtaining Labeled Training Data

In order to train a supervised model, we need a large number of audio files with corresponding label files which must contain chord names and boundaries. To automate this laborious process, we use symbolic data to generate label files as well as to create time-aligned audio files. To this end, we first convert a symbolic file to a format which can be used as an input to a chord-analysis tool. Chord analyzer then performs harmonic analysis and outputs a file with root information and note names from which complete chord information (*i.e.*, root and its sonority – major, minor, or diminished) is extracted. Sequence of chords are used as pseudo ground-truth or labels when training the HMMs along with proper feature vectors.

We used symbolic files in MIDI (Musical Instrument Digital Interface) format. For harmonic analysis, we used the Melisma Music Analyzer developed by Sleator and Temperley [11]. The Melisma Music Analyzer takes a piece of music represented by an event list, and extracts musical information from it such as meter, phrase structure, harmony, pitch-spelling, and key. By combining harmony and key information extracted by the analysis program, we can generate label files with sequence of chord names and accurate boundaries.

The symbolic harmonic-analysis program was tested on a corpus of excerpts and the 48 fugue subjects from the *Well-Tempered Clavier*, and the harmony analysis and the key extraction yielded an accuracy of 83.7% and 87.4%, respectively [12].

We then synthesize the audio files using Timidity++. Timidity++ is a free software synthesizer, and converts MIDI files into audio files in a WAVE format. [1] It uses a sample-based synthesis technique to create harmonically rich audio as in real recordings. We used Fluid3 sound font for synthesis. The raw audio is downsampled to 11025 Hz, and 12-dimensional chroma features are first extracted from audio, and were projected onto a 6-dimensional space

[1] http://timidity.sourceforge.net/

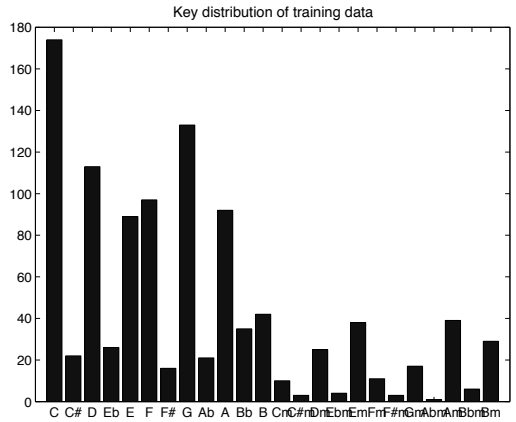


Figure 3. Key distribution from 1,046 MIDI files.

to generate tonal centroid features. We used the frame size of 8192 samples and the hop size of 2048 samples, corresponding to 743 ms and 186 ms, respectively.

The MIDI files we used for harmonic analysis and synthesis were acquired from `http://www.mididb.com`, which were all rock music. The number of MIDI files was 1,046, which correspond to 1,070,752 feature vectors or 55.25 hours of audio.

Figure 3 shows the distribution of 24 keys from 1,046 files.

3.2 Feature Vector

Harte *et. al* proposed a 6-dimensional feature vector called *Tonal Centroid*, and used it to detect harmonic changes in musical audio [13]. It is based on the Harmonic Network or *Tonnetz*, which is a planar representation of pitch relations where pitch classes having close harmonic relations such as fifths, major/minor thirds have smaller Euclidean distances on the plane.

The Harmonic Network is a theoretically infinite plane, but is wrapped to create a 3-D Hypertorus assuming enharmonic and octave equivalence, and therefore there are just 12 chromatic pitch classes. If we reference C as a pitch class 0, then we have 12 distinct points on the circle of fifths from 0-7-2-9-$\cdots$-10-5, and it wraps back to 0 or C. If we travel on the circle of minor thirds, however, we come back to a referential point only after three steps as in 0-3-6-9-0. The circle of major thirds is defined in a similar way. This is visualized in Figure 4. As shown in Figure 4, the six dimensions are viewed as three coordinate pairs $(x1, y1)$, $(x2, y2)$, and $(x3, y3)$.

Using the aforementioned representation, a collection of pitches like chords is described as a single point in the 6-D space. Harte *et. al* obtained a 6-D tonal centroid vector by projecting a 12-bin tuned chroma vector onto the three circles in the equal tempered Tonnetz described above. By calculating the Euclidean distance between successive analysis frames of tonal centroid vectors, they successfully detect harmonic changes such as

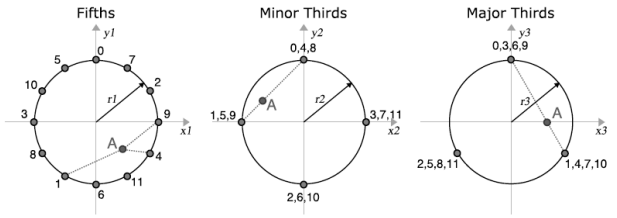


Figure 4. Visualizing the 6-D Tonal Space as three circles: fifths, minor thirds, and major thirds from left to right. Numbers on the circles correspond to pitch classes and represent nearest neighbors in each circle. Tonal Centroid for A major triad (pitch class 9,1, and 4) is shown at point A (adapted from Harte *et. al* [13]).

chord boundaries from musical audio.

While a 12-dimensional chroma vector has been widely used in most chord recognition systems, it was shown that the tonal centroid feature yielded far less errors in [10]. The hypothesis was that the tonal centroid vector is more efficient and more robust because it has only 6 dimensions, and it puts emphasis on the interval relations such as fifths, major/minor thirds, which are key intervals that comprise most of musical chords in Western tonal music.

3.3 Key-Specific Hidden Markov Model

A hidden Markov model [14] is an extension of a discrete Markov model, in which the states are *hidden* in the sense that an underlying stochastic process is not directly observable, but can only be observed through another set of stochastic processes.

We recognize chords using 24-state HMMs. Each state represents a single chord, and the observation distribution is modeled by a single Gaussian with diagonal covariance matrix. State transitions obey the first-order Markov property; *i.e.*, the future is independent of the past given the present state. In addition, we use an ergodic model since we allow every possible transition from chord to chord, and yet the transition probabilities are learned.

In our model, we have defined two chord types – major and minor – for each of 12 chromatic pitch classes, and thus we have 24 classes in total. We grouped triads and seventh chords with the same root into the same category. For instance, we treated E minor triad and E minor seventh chord as just E minor chord without differentiating the triad and the seventh.

With the labeled training data obtained from the symbolic files, we first train our models to estimate the model parameters for each key model. Once the model parameters are learned, we extract the feature vectors from the real recordings, and estimate the key by computing the likelihood of each key model given the observation sequence using the Baum-Welch or forward-backward algorithm. We then apply the Viterbi algorithm to the selected key model to find the optimal state path, *i.e.*, chord sequence, in a maximum likelihood sense.

Figure 5 shows the transition probability matrix of a C major key model and the mean and covariance vectors of a G major chord in the same model. Transition probability matrix is strongly diagonal since the frame rate is usually faster than the rate the chord changes. However, it is also shown that the transitions to dominant or subdominant chords are relatively frequent as is expected in Western tonal music. This is indicated by darker off-diagonal lines 5 or 7 semitones apart from the main diagonal line.

Figure 6 displays the transition probabilities from G major chord in C major and in C minor key models. The obvious difference in transition probabilities shown in Figure 6, in spite of the fact that they are both from the same G major chord, supports our hypothesis that key-specific models help make a correct decision especially when there is great confusion caused by a observation vector. For instance, F major triad and A minor triad share two chord notes – A and C – in common, and thus the observation may look very similar. Given no prior about the key, when the previous chord was G major chord, it will be difficult to make a confident decision about which chord will follow – C major or A minor chord. Assuming it's in a C major key, however, the system will classify it as F major chord with much more confidence since the transition probability from G major to F major chord is far greater than that from G major to A minor chord in C major key (indicated by a dashed circle on the left). For the same reason, it will be recognized as A minor chord with confidence in a C minor key model. We believe such key-specific chord progression characteristics will help decrease confusion seen in observation vectors, resulting in increased performance.

4 EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Evaluation

We tested our models' performance on the two whole albums of Beatles (CD1: *Please Please Me*, CD2: *Beatles For Sale*), which were also used as test data set in [3], [6]. Each album contains 14 tracks, and ground-truth annotations were provided by Harte and Sandler at the Digital Music Center at University of London in Queen Mary. [2]

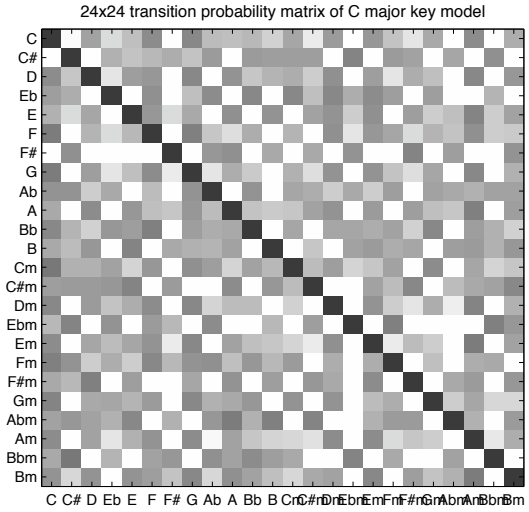
In computing scores, we only counted exact matches as correct recognition. We tolerated the errors at the chord boundaries by having a time margin of one frame, which corresponds approximately to 0.19 second. This assumption is fair since the segment boundaries were generated by human by listening to audio, which cannot be razor sharp.

To examine the validity of the key-specific models, we also built the key-independent, universal model trained on all 1,046 files, and compared the performance.

4.2 Results and Discussion

Table 1 shows the frame-rate accuracy in percentage for each model.

[2] http://www.elec.qmul.ac.uk/digitalmusic/



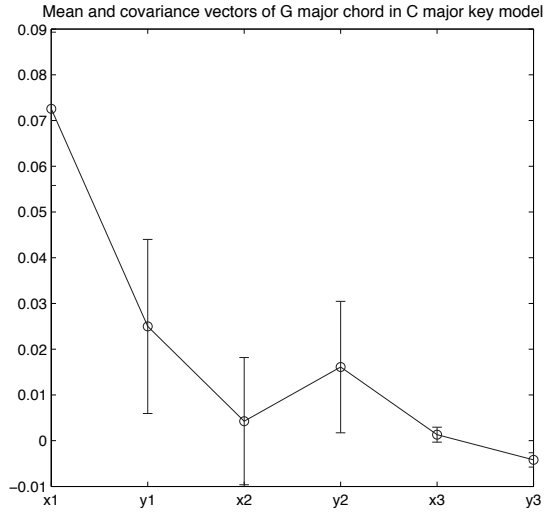


Figure 5. (a) 24×24 transition probability matrix of a C major key model. For viewing purpose, logarithm was taken of the original matrix. Axes are labeled in the order of major and minor chords. (b) Mean and covariance vectors of a G major chord in the same model.

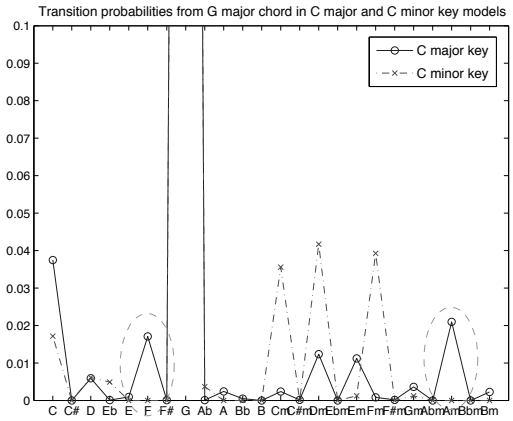


Figure 6. Transition probabilities from G major chord in C major key (solid) and in C minor key model (dash-dot). Note that the probability to itself (G major chord) is very high in both cases.

Model	Key-independent	Key-specific	Increase
CD1	61.026	63.978	4.837
CD2	84.482	84.746	0.312
Total	72.754	74.362	2.210

Table 1. Chord recognition results (% accuracy)

The results shown in Table 1 proves our hypothesis on key-specific models although they don't make a significant improvement over a key-independent model. Particularly, the increase in performance is greater with CD1 for which overall performance is much lower than that for CD2. A possible explanation for this is using key-specific models is of greater help when there are more ambiguities in observation vectors.

Our results compare favorably with other state-of-the-art systems proposed by Harte and Sandler [3] or by Bello and Pickens [6]. Using the same test data set, Harte and Sandler obtained 53.9% and 70.8% of frame-rate accuracy for CD1 and CD2, respectively. They defined 48 different triads including augmented triads, and used a pattern matching algorithm for chord identification, followed by median filtering for smoothing. Using the HMMs with 24 states for just major/minor chords, Bello and Pickens' system yielded the performance of 68.55% and 81.54% for CD1 and CD2, respectively. However, they went through a pre-processing stage of beat detection to perform a tactus-based analysis. Without a beat-synchronous analysis, their accuracy drops down to 58.96% and 74.78% for each CD, which is lower than our results which are 63.98% and 84.75%.

As to key estimation, 22 out of 26 tracks were correctly identified, corresponding to 84.62% of accuracy. Two tracks were disregarded in key estimation because of their ambiguities. Table 2 shows a confusion matrix for the key estimation task.

We can observe from the confusion matrix in Table 2 that all mis-recognized keys are in fifth relations with their original keys (G-C and D-A). It is probably due to such keys not only share many chord in common but also the chord progression pattern is similar to each other.

Key	C	D	E	G	A	B♭	Accuracy (%)
C(4)	3	0	0	1	0	0	75.00
D(5)	0	2	0	0	3	0	40.00
E(8)	0	0	8	0	0	0	100.00
G(4)	1	0	0	3	0	0	75.00
A(4)	0	0	0	0	4	0	100.00
B♭(1)	0	0	0	0	0	1	100.00

Table 2. Confusion matrix for key estimation task

5 CONCLUSION

In this paper, we describe a system for automatic chord transcription and key extraction from the raw audio. The main contribution of this work is the demonstration of automatic generation of labeled training data for machine learning models which allows for richer models like key-specific HMMs, resulting in very promising results in both musical tasks.

Using symbolic music files such as MIDI was a key to avoiding extremely laborious process of manual annotation. In order to achieve this goal, we first perform harmonic analysis on the symbolic data, which contain noise-free pitch and time information, to generate label files with chord names and precise time boundaries. In parallel, by using a sample-based synthesizer, we could create audio files which have harmonically rich spectra as in real acoustic recordings. The label files and audio generated from the same symbolic files are thus in perfect alignment, and were used to train our models.

As feature vectors, we used a 6-dimensional feature called Tonal Centroid, which was proved to outperform a conventional chroma feature in previous work by the same authors.

Each state in HMMs was modeled by a multivariate, single Gaussian completely represented by its mean vector and covariance matrix. We have defined 24 classes or chord types in our models, which include for each pitch class major and minor chords. We treated seventh chords as their corresponding root triads, and disregarded diminished and augmented chords since they very rarely appear in Western tonal music, especially in rock music.

Based on the close relationship between key and chord in Western tonal music, we have built 24 key-specific HMMs, one for each key. We then applied the same approach as used in isolated word recognition systems. That is, given the observation sequence, we computed the likelihood of each key model to estimate the key, and the Viterbi decoder returned the optimal state path which is identical to the frame-rate chord sequence.

Experiments showed a slightly higher performance with the key-specific model than with the key-independent model trained on all data regardless of keys.

In this paper, we trained our models only on rock music, and the test data was of the same kind. It was shown in [15] that the genre also has a great impact in model's performance. We therefore plan to build genre-specific models and combine them with key-specific models to develop a model for each genre and key. A smoothing technique must be accompanied in such models due to a data sparsity problem.

In addition, we consider higher-order HMMs in the future because chord progressions based on Western tonal music theory reveal such higher-order characteristics. Therefore, knowing two or more preceding chords will help make a correct decision. We also plan to build richer models using Gaussian mixture models or Support Vector Machines in order to better represent the emission probabilities as we increase the size of training data even more.

6 REFERENCES

[1] K. Lee, "Identifying cover songs from audio using harmonic representation," in *extended abstract submitted to Music Information Retrieval eXchange task*, BC, Canada, 2006.

[2] T. Fujishima, "Realtime chord recognition of musical sound: A system using Common Lisp Music," in *Proceedings of the International Computer Music Conference*. Beijing: International Computer Music Association, 1999.

[3] C. A. Harte and M. B. Sandler, "Automatic chord identification using a quantised chromagram," in *Proceedings of the Audio Engineering Society*. Spain: Audio Engineering Society, 2005.

[4] K. Lee, "Automatic chord recognition using enhanced pitch class profile," in *Proceedings of the International Computer Music Conference*, New Orleans, USA, 2006.

[5] A. Sheh and D. P. Ellis, "Chord segmentation and recognition using EM-trained hidden Markov models," in *Proceedings of the International Conference on Music Information Retrieval*, Baltimore, MD, 2003.

[6] J. P. Bello and J. Pickens, "A robust mid-level representation for harmonic content in music signals," in *Proceedings of the International Conference on Music Information Retrieval*, London, UK, 2005.

[7] J. Morman and L. Rabiner, "A system for the automatic segmentation and classification of chord sequences," in *Proceedings of Audio and Music Computing for Multimedia Workshop*, Santa Barbar, CA, 2006.

[8] K. Lee and M. Slaney, "Automatic chord recognition using an HMM with supervised learning," in *Proceedings of the International Conference on Music Information Retrieval*, Victoria, Canada, 2006.

[9] ——, "Automatic chord recognition from audio using a supervised HMM trained with audio-from-symbolic data," in *Proceedings of Audio and Music Computing for Multimedia Workshop*, Santa Barbar, CA, 2006.

[10] ——, "Automatic chord transcription from audio using key-dependent HMMs trained on audio-from-symbolic data," 2007, in review.

[11] D. Sleator and D. Temperley, "The Melisma Music Analyzer," http://www.link.cs.cmu.edu/music-analysis/, 2001.

[12] D. Temperley, *The cognition of basic musical structures*. The MIT Press, 2001.

[13] C. A. Harte, M. B. Sandler, and M. Gasser, "Detecting harmonic change in musical audio," in *Proceedings of Audio and Music Computing for Multimedia Workshop*, Santa Barbara, CA, 2006.

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

[15] K. Lee, "A system for automatic chord recognition from audio using genre-specific hidden Markov models," in *International Workshop on Adaptive Multimedia Retrieval*, Paris, France, 2007, in review.