# Acoustic Chord Transcription and Key Extraction From Audio Using Key-Dependent HMMs Trained on Synthesized Audio

Kyogu Lee, *Member, IEEE*, and Malcolm Slaney, *Senior Member, IEEE*

***Abstract*—We describe an acoustic chord transcription system that uses symbolic data to train hidden Markov models and gives best-of-class frame-level recognition results. We avoid the extremely laborious task of human annotation of chord names and boundaries—which must be done to provide machine learning models with ground truth—by performing automatic harmony analysis on symbolic music files. In parallel, we synthesize audio from the same symbolic files and extract acoustic feature vectors which are in perfect alignment with the labels. We, therefore, generate a large set of labeled training data with a minimal amount of human labor. This allows for richer models. Thus, we build 24 key-dependent HMMs, one for each key, using the key information derived from symbolic data. Each key model defines a unique state-transition characteristic and helps avoid confusions seen in the observation vector. Given acoustic input, we identify a musical key by choosing a key model with the maximum likelihood, and we obtain the chord sequence from the optimal state path of the corresponding key model, both of which are returned by a Viterbi decoder. This not only increases the chord recognition accuracy, but also gives key information. Experimental results show the models trained on synthesized data perform very well on real recordings, even though the labels automatically generated from symbolic data are not 100% accurate. We also demonstrate the robustness of the tonal centroid feature, which outperforms the conventional chroma feature.**

***Index Terms*—Acoustic chord transcription, hidden Markov model (HMM), key-dependent models, key extraction, symbolic music files.**

## I. Introduction

A MUSICAL key and a chord are important attributes of Western tonal music. A key defines a referential point or a tonal center upon which we arrange musical phenomena such as melody, harmony, cadence, etc. A musical chord is a set of simultaneous tones. A succession of chords over time, or a chord progression, forms the harmony core 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 and deriving harmonic functions, based on the key.

Finding the key and labeling the chords automatically from audio are of great use for performing harmony analysis of music. Once the harmonic content of a piece is known, we can use a sequence of chords for further higher level structural analysis to define themes, phrases, or forms.

Chord sequences and the timing of chord boundaries are a compact and robust mid-level representation of musical signals; they have many potential applications, such as music identification, music segmentation, music-similarity finding, and audio thumbnailing. Chord sequences have been successfully used as a front end to an audio cover-song identification system [1]. For these reasons and others, automatic chord recognition is attractive to researchers in the music information retrieval field.

Most chord-recognition systems use a chroma vector or its variation as the feature set. A chroma vector is often a 12-dimensional vector, each dimension representing spectral energy in a pitch class in a chromatic scale. We also describe the use of the tonal centroid vector, which is a six-dimensional feature obtained from a 12-dimensional chroma feature as introduced by Harte *et al.* [2], and compare it with the conventional chroma vector. We show that the tonal centroid features are more robust and outperform the conventional chroma features in chord recognition [3], [4].

Research for the last 20 years shows that HMMs are very successful for speech recognition. A hidden Markov model [5] is an extension of a discrete Markov model, in which the states are *hidden* in the sense that we cannot directly observe the underlying stochastic process, but can only observe it through another set of stochastic processes. The three parameters that define an HMM are the observation probability distribution, the state transition probability distribution, and the initial state distribution; we can accurately estimate these parameters from the labeled training data.

Much progress in speech recognition has been made with gigantic databases with labels. Such a huge database not only enables researchers to build richer models, but also helps estimate the model parameters precisely, resulting in improved performance. However, there are very few such databases 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. Consider the huge acoustical differences among: a C major chord in root position played by a piano, the same chord in first inversion played by a rock band, and the same chord in second inversion played by a full orchestra. All of these sounds must be transcribed as the same C major chord;

Manuscript received September 23, 2007; revised November 13, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Vesa Valimaki.

K. Lee is with the Center for Computer Research in Music and Acoustics, Stanford University, Stanford, CA 94305 USA and also with Gracenote, Inc., Emeryville, CA 94608 USA.

M. Slaney is with the Center for Computer Research in Music and Acoustics, Stanford University, Stanford, CA 94305 USA and also with Yahoo! Research, Santa Clara, CA 95054 USA.

Digital Object Identifier 10.1109/TASL.2007.914399

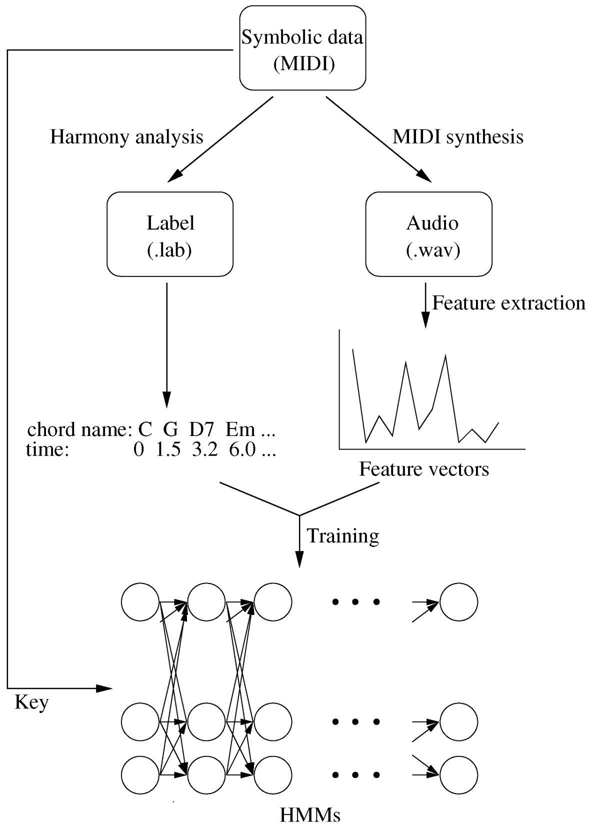

Fig. 1. Training the chord transcription system. Labels obtained through harmony analysis on symbolic music files and feature vectors extracted from audio synthesized from the same symbolic data are used to train HMMs.

this in turn means even more data are needed to train the models so they generalize.

However, it is very difficult to obtain a large set of training data for music. First of all, the annotator must have a certain level of expertise in music theory or musicology to perform harmony analysis. Second, 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 method of automating the daunting task of providing the machine-learning models with labeled training data. To this end, we use symbolic music files, such as MIDI files, to generate chord names and precise corresponding boundaries, as well as to create audio. Instead of a digitized audio signal like a pulse code modulation (PCM) waveform, MIDI files contain a set of event messages such as pitch, velocity, and note duration, along with clock signals from which we can synthesize audio. Audio and chord-boundary information generated this way are in perfect alignment, and we can use them to directly estimate the model parameters. The overall process of training is illustrated in Fig. 1.

There are several advantages to this approach. First, a great number of symbolic music 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 richer models so that we can include more chord types such as a seventh, augmented, or diminished. Lastly, by using a sample-based synthesis technique, we can generate harmonically rich audio as in real acoustic recordings. Although there may be noticeable differences in sonic quality between real acoustic recording and synthesized audio, we do not believe that the lack of human touch, which makes a typical MIDI performance dry, affects our training program.

This paper continues with a review of related work in Section II. In Section III, we describe the feature set we used as a front end to the system. In Section IV, we explain the method of obtaining the labeled training data and describe the procedure of building our models. In Section V, we describe our evaluation method and present empirical results with discussions. We draw conclusions in Section VI, followed by directions for future work.

## II. Related Work

Several systems describe chord recognition from raw audio. Some systems use a simple pattern-matching algorithm based on predefined chord templates [6]–[8], while others use more sophisticated machine-learning techniques, such as hidden Markov models (HMMs) or support vector machines (SVMs) [9]–[11]. Our work is closest to two previous works described below that used machine-learning techniques to recognize chords.

Sheh and Ellis propose a statistical learning method for chord segmentation and recognition using the chroma features as a front end [9]. They use the HMMs trained by the expectation-maximization (EM) algorithm, and treated the chord labels as hidden values within the EM framework. They use only the sequence of chord names, without chord boundaries, as an input to the models, and apply the forward–backward algorithm to estimate the model parameters. The frame-level accuracy they obtain is about 76% for segmentation and about 22% for recognition. The poor performance for recognition may be due to insufficient training data for 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 use the chroma features and HMMs with the EM algorithm to find the crude transition probability matrix for each input [10]. What is novel in their approach is that they incorporate musical knowledge into the models by defining a state transition matrix based on the key distance in a circle of fifths, and avoid random initialization of a mean vector and a covariance matrix of observation distribution. In addition, in training the model's parameter, they selectively update 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. They test their system on the Beatles' two full albums. The frame-level accuracy is about 67%, and it increases up to about 75% with beat-synchronous segmentation. We believe this is the state-of-the-art, and thus we compare our system's performance to theirs. In particular, they argue that the accuracy increases as much as 32% when observation distribution parameters are held constant. They focus on extracting from the raw waveform a robust mid-level representation, however, and thus use a much smaller set of chord types (24 major/minor triads only) compared with 147 chord types defined by Sheh and Ellis.

The present paper expands our previous work on chord recognition using HMMs trained on synthesized audio [12], [13]. Our system is based on the work of Sheh and Ellis and Bello and Pickens, in that the states in the HMM represent chord types, and the optimal path, i.e., the most probable chord sequence, is found 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, which not only increase the chord recognition accuracy but also provide key information. Finally, we train our models on the data sets of different musical genres and investigate the effect of each parameter set when various types of input are given. We also demonstrate the robustness of the tonal centroid feature because it yields better performance than the chroma feature when tested on different kinds of input.

In evaluating our performance, we compare two different approaches to chord recognition—a data-based model and a knowledge-based model. Our model is a data-based model—all model parameters are learned from the training data. On the other hand, we contrast our approach with Bello and Pickens' approach which is knowledge-based. Here, the output distribution parameters are fixed based on an expert's music theoretical knowledge, although Bello and Pickens adapt their transition probabilities based on their initial transcription [10].

## III. Feature Vector

Our system starts by extracting suitable feature vectors from the raw audio. In this paper, we compare two different feature sets. First, like most chord-recognition systems, a chroma vector or a pitch class profile (PCP) vector is used. The second feature vector we use is called the tonal centroid, and proves to be a better feature set for our chord-recognition system.

### A. Chroma Vector

A chromagram or PCP is the feature set of choice in automatic chord recognition or key extraction since it was first introduced by Fujishima [6]. Perception of musical pitch involves two dimensions—*height* and *chroma*. Pitch height moves vertically in octaves, indicating to which octave a note belongs. On the other hand, the chroma tells where it stands in relation to others within an octave. A chromagram or a PCP is a $B$-dimensional vector representation of a chroma, where $B$ is the number of bins in an octave, and represents the relative intensity in each of 12 semitones in a chromatic scale. Since a chord is composed of a set of tones and its label is only determined by the position of those tones in a chroma, regardless of their heights, the chroma vectors appear to be an ideal feature to represent a musical chord or a musical key. Fujishima develops a real-time chord-recognition system, where he derives a 12-dimensional pitch class profile from the DFT of the audio signal, and performs pattern matching using binary chord type templates [6].

There are some variations when computing a chromagram. We use one based on constant-Q transform (CQT) to compute a 12-dimensional chromagram following these steps. First, the discrete Fourier transform (DFT) of the input signal $X(k)$ is computed, and the constant-Q transform $X_{CQ}$ is calculated from $X(k)$, using logarithmically spaced frequencies to reflect the way humans perceive sound [14]. The frequency resolution of the constant-Q transform follows that of the equal-tempered scale, which is also logarithmically based, and the $k$th spectral component is defined as

$$f_k = (2^{1/B})^k f_{\min} \tag{1}$$

where $f_k$ varies from $f_{\min}$ to an upper frequency, both of which are set by the user, and $B$ is the number of bins in an octave in the constant-Q transform. Once $X_{CQ}(k)$ is computed, a chromagram vector $CH$ is easily obtained as

$$CH(b) = \sum_{m=0}^{M-1} |X_{CQ}(b+mB)| \tag{2}$$

where $b = 1, 2, \cdots, B$ is the chromagram bin index, and $M$ is the number of octaves spanned in the constant-Q spectrum. Although there are only 12 pitch classes in a chromatic scale, $B = 24$ or $B = 36$ is also used for fine tuning.

In our system, we use a feature vector based on the 12-bin quantized chromagram proposed by Harte and Sandler [7], which compensates for possible mis-tuning present in the recordings by reallocating the peaks based on the peak distribution.

### B. Tonal Centroid

Recently, Harte *et al.* propose a six-dimensional feature vector called tonal centroid and use it to detect harmonic changes in musical audio [2]. 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 or major/minor thirds have smaller Euclidean distances on the plane.

The harmonic network is a theoretically infinite plane, but is wrapped along the dimensions of fifths, minor thirds, and major thirds to create a 3-D hypertorus, assuming enharmonic and octave equivalence. 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 Fig. 2. As shown in Fig. 2, the six dimensions are viewed as three coordinate pairs $(x1, y1)$, $(x2, y2)$, and $(x3, y3)$.

Using the aforementioned representation, a collection of pitches in a chord is described as a single point in the 6-D space. Harte *et al.* obtain 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. In other words, because each pitch class in a chromatic scale is mapped to a

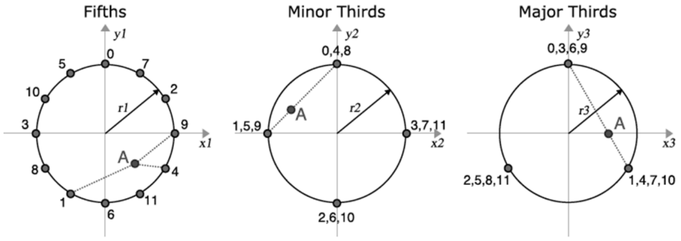

Fig. 2. Visualizing the 6-D tonal space as three circles: fifths, minor thirds, and major thirds. Numbers on the circles correspond to pitch classes and represent nearest neighbors in each circle. Tonal centroid for A major triad (pitch class 9,1, and 4) is shown at point A (adapted from Harte et. al [2]).

point on three circles, we compute the tonal centroid vector by multiplying a 6 × 12 transformation matrix with a 12 × 1 chroma vector. By calculating the Euclidean distance between successive analysis frames of tonal centroid vectors, they successfully detect harmonic changes, such as chord boundaries from musical audio.

We use the tonal centroid feature, as well as the conventional 12-bin chroma vector, and compare their performance. We hypothesize the tonal centroid vector is more efficient and more robust because it has only six dimensions, and puts emphasis on the interval relations such as fifths and major/minor thirds, which are key intervals that comprise most of musical chords in Western tonal music.

## IV. System

### A. Obtaining Labeled Training Data

In order to train a supervised model, we need a large number of audio files with corresponding label files that contain annotated chord boundaries as well as chord names. To automate this laborious process, we use symbolic data to generate label files as well as to create audio data. To this end, we first convert a symbolic file to a format which can be used as an input to a chord-analysis tool. The chord analyzer then performs harmony analysis and outputs a file with root information and note names from which we extract complete chord information (i.e., root and its sonority—major, minor, or diminished triad/seventh). We use sequence of chords as ground truth, or labels, when training the HMMs.

To examine the model's dependency on the training data, we choose two different training data sets with different types of music. For the first parameter set, we use 765 classical symbolic music files as a training data set, which comprise 406 pieces of solo keyboard music and 359 string quartets by J. S. Bach, Beethoven, Haydn, Mozart, and other composers. All classical symbolic music files are in a Humdrum data format from the Center for Computer Assisted Research in the Humanities at Stanford University. Humdrum is a general-purpose software system intended to help music researchers encode, manipulate, and output a wide variety of musically pertinent representations [15]. These files are converted to a format which can be used in the Melisma Music Analyzer, as well as to a MIDI format using the tools developed by Craig Sapp.[1]

For the second training set, we use 158 MIDI files of the Beatles available from [Online]. Available: http://www.mididb.com.

The audio data synthesized from these symbolic music files of the classical and the Beatles data set are 26.73 h long or 517 945 feature frames, and 5.73 h long or 111 108 feature frames, respectively.

We perform harmony analysis to obtain chord labels using the Melisma Music Analyzer developed by Sleator and Temperley [16]. Melisma performs harmony analysis on a piece of music represented by an event list and extracts information about meter, harmony, and key, so on. We configure Melisma so that it outputs a chord name every beat and use it as ground truth. When building key-dependent models, we take the beginning key as a home key for an entire piece.

Temperley tests the symbolic harmony-analysis program on a corpus of excerpts and the 48 fugue subjects from the *Well-Tempered* Clavier; the harmony analysis and the key extraction yields an accuracy of 83.7% and 87.4%, respectively [17].

Fig. 3 shows the normalized distributions of chords and keys extracted from Melisma for each training set.

We synthesize the audio files using Timidity++ (Timidity++ is a free software synthesizer and converts MIDI files into audio files in a WAVE format.[2] It uses a sample-based synthesis technique to create harmonically rich audio as in real recordings.) We use GUS (Gravis Ultra Sound) sound font to synthesize the MIDI files.[3] The set of instruments we use to synthesize classical music are piano, violin, viola, and cello. When rendering the Beatles' MIDI files, we use electric piano, electric guitar, steal string guitar, electric bass, and orchestral strings. The raw audio is downsampled to 11 025 Hz; 12-bin chroma features and 6-D tonal centroid features are extracted from it with the frame size of 8192 samples and the hop size of 2048 samples, which gives the frame rate of approximately 5.4 frames/s. The frame-level chroma vectors or tonal centroid vectors are then used as input to the HMMs along with the label files obtained above.

### B. Hidden Markov Model

We recognize chords using either 24-state (the Beatles music) or 36-state (classical music) HMMs. Each state represents a single chord; the observation distribution is modeled by a single multivariate Gaussian—in 12 dimensions for the chroma feature or in six dimensions for the tonal centroid feature—defined by its mean vector $\mu_i$ and covariance matrix $\Sigma_i$, where $i$ denotes the $i$th state. We assume the dimensions of the features are uncorrelated with each other, and thus use a diagonal-covariance matrix.[4] State transitions obey a 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 define 36 classes or chord types according to their sonorities only—major, minor, and diminished chords for each pitch class. We ignore the augmented chords since they

[1] [Online]. Available: http://www.extras.humdrum.net.

[2] [Online]. Available: http://timidity.sourceforge.net

[3] [Online]. Available: http://www.gravis.com

[4] We tried full-covariance observation matrices, but our recognition was lower, suggesting that we do not have enough data.

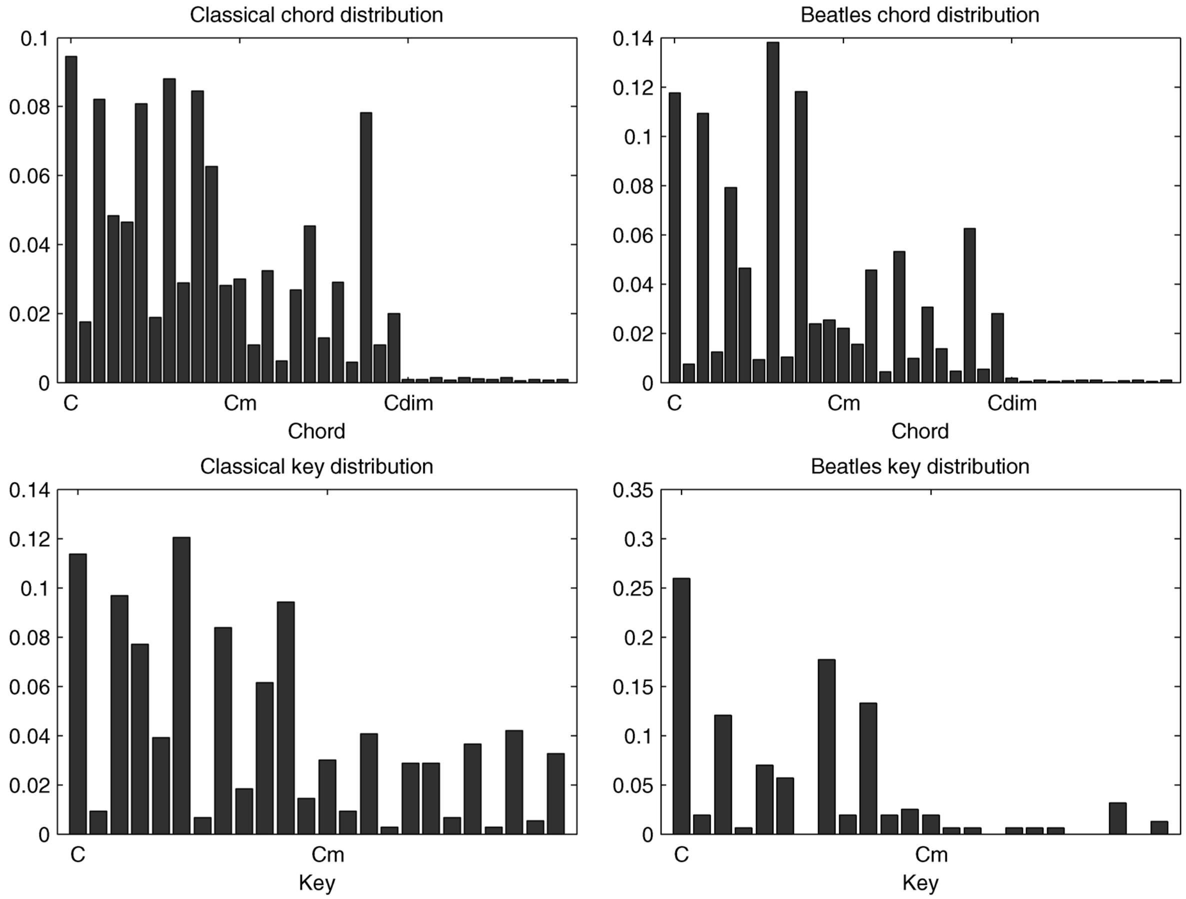

Fig. 3. Chord and key distribution of classical and the Beatles training data.

rarely appear in Western tonal music. We group triads and seventh chords with the same sonority into the same category. For instance, we treat E minor triad and E minor seventh chord as just E minor chord without differentiating the triad and the seventh. Most pop or rock music, as in the Beatles, makes use of only 24 major/minor chords, so for our experiments with popular music we recognized only 24 chords, as done by Bello and Pickens [10].

With the labeled training data we obtain from the symbolic files, we first train our models to estimate the model parameters. Once we learn the model parameters—initial state probabilities, state transition probability matrix, and mean vector and covariance matrix for each state—we recognize chords for an unknown input by extracting the feature vectors from the raw audio and applying the Viterbi algorithm to the appropriate model to find the optimal path, i.e., chord sequence, in a maximum-likelihood sense.

### C. *Parameter Estimates*

Fig. 4 shows transition probability matrices estimated from each training data set. The transition matrices are strongly diagonal since a chord's duration is usually longer than the frame rate, and thus the state does not change for several frames, which makes a transition probability to itself highest.

As further illustrated in Fig. 5, however, the chord progression observed in the transition probabilities is rooted in music theory. The C major chord has the largest probability of staying within the same state, i.e., within a C major chord, because of faster frame rate than the rate of chord changes. However, it has comparably higher probabilities for making a transition to specific chords such as an F major, G major, F minor, or A minor chord than to others. F major and G major have subdominant-tonic and dominant-tonic relationships with C major, respectively, and transitions between them happen very often in Western tonal music. A C major chord is also a dominant chord of an F minor, and therefore a C major to F minor transition is frequent as well. Finally, an A minor chord is a relative minor of the C major chord, and a C-to-Am transition also occurs quite often. This tonic-dominant-subdominant relationship is also shown in Fig. 4 as off-diagonal lines with five and seven semitone offsets with respect to their tonics.

Fig. 6 shows the observation distribution parameters for chroma feature estimated from each training data set for C major chord. On the left are the mean chroma vector and diagonal covariance vector for the HMM trained on classical music, and those for the Beatles' music are on the right. It is obvious, as expected, that they both have three large peaks at chord tones or at C, E, and G. In addition, we can also see relatively large peaks at D and B, which come from the third harmonics of chord tones G and E. Mean vectors and covariance matrices of tonal centroid feature are also shown in Fig. 7 for each data set.

Although we can estimate the model parameters for observation distribution for each chord, the number of feature samples

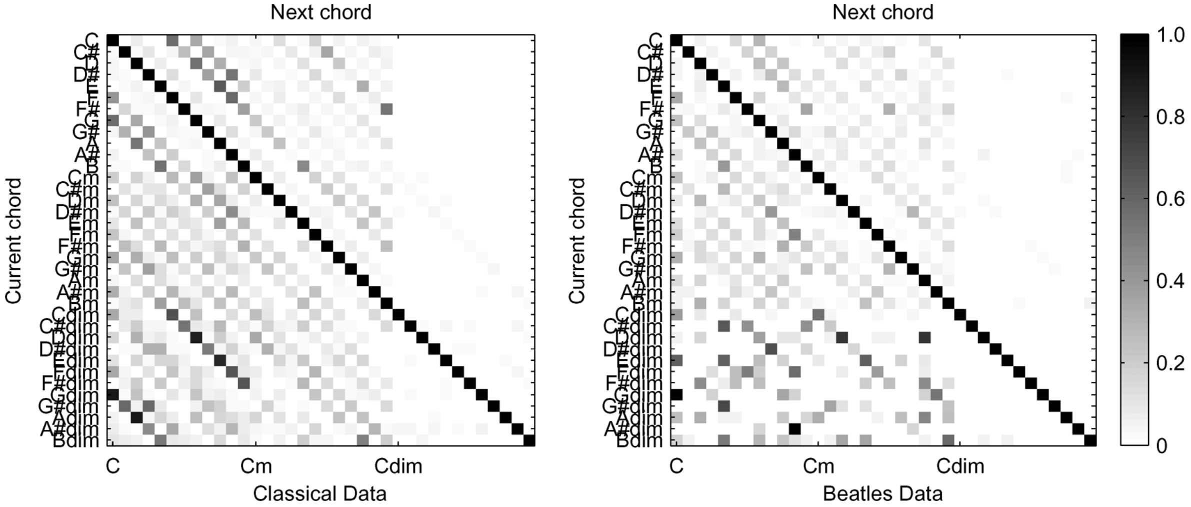

Fig. 4. 36 × 36 transition probability matrices obtained from 765 pieces of classical music and from 158 pieces of the Beatles' music. For viewing purpose, logarithm is taken of the original matrices. Axes are labeled in the order of major, minor, and diminished chords. The right third of these matrices are mostly zero because these musical pieces are unlikely to transition from a major or minor chord to a diminished chord, and once in a diminished chord, the music is likely to transition to a major or minor chord again.

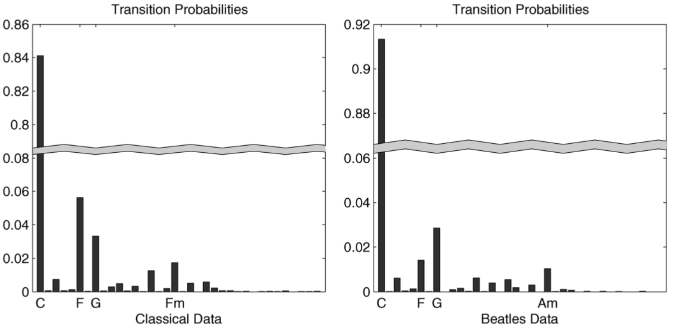

Fig. 5. Transition probabilities from C major chord estimated from classical and from the Beatles' data. The $X$ axis is labeled in the order of major, minor, and diminished chords.

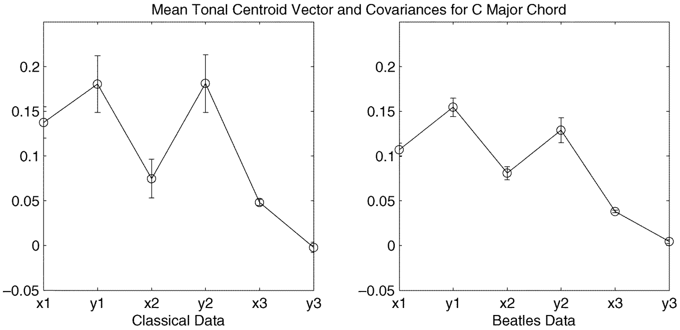

Fig. 7. Mean tonal centroid vector and covariances for C major chord estimated from classical and from the Beatles' data. Because we use diagonal covariance, the variances are shown with "error" bars on each dimension.

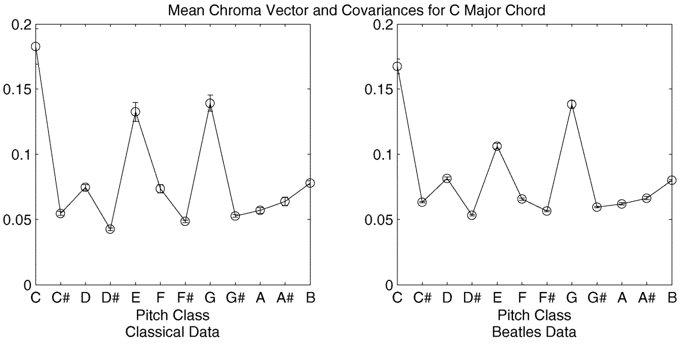

Fig. 6. Mean chroma vector and covariances for C major chord estimated from classical and from the Beatles' data. Because we use diagonal covariance, the variances are shown with "error" bars on each dimension.

in training data not only varies to a great degree from chord to chord but also is limited for some chords, as shown in the chord distributions in Fig. 3. This is likely to cause class statistics errors when estimating the mean vectors and covariance matrices from the available training samples, which may lead to overfitting. We therefore transpose all the major chords to a single major chord of no tonal center and then estimate its probability distribution.

For example, if we wish to estimate the parameters for the C major chord, we downshift the chroma vectors of C♯ major chord by 1, those of D major chord by 2, $\cdots$ and those of B major chord by 11, respectively; we now have more feature samples than we had for the original C major chord. Such a transposition method is valid because in a 12-dimensional chroma representation, only the relative spacing between pitch classes is important, not the absolute location. Similarly, we estimate the distribution parameters for 12 minor and 12 diminished chords. This simple transposition method increases the number of feature samples per class to give more accurate parameter estimates reducing the need for regularization. We use this method to obtain the distribution parameter estimates shown in Figs. 6 and 7.

### D. Key-Dependent HMMs

In Western tonal music, a key and chords are very closely related; thus, knowing the key of a piece provides very valuable information about the chords as well. For instance, if a musical piece is in the key of C major, then we can expect frequent appearances of chords such as C major, F major, and G

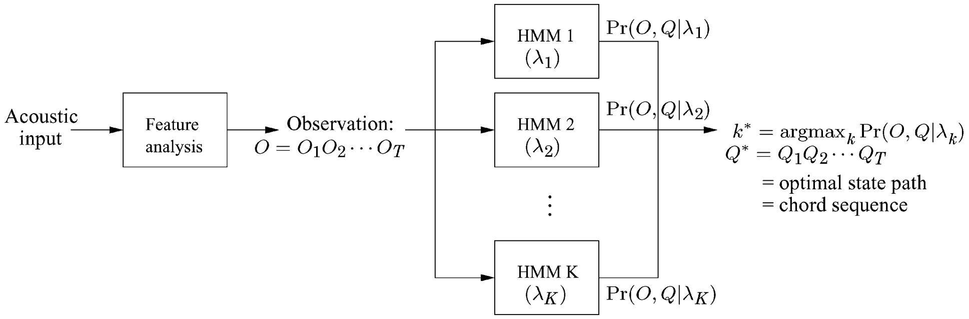

Fig. 8. System for key estimation and chord recognition using key-dependent models ($K = 24$).

major, which correspond to the tonic, subdominant, and dominant chord, respectively. On the other hand, F♯ minor or A♭ major chord do not appear, since neither has any harmonic function in a C major key.

Another great advantage of using symbolic music files is that other information such as the key or the tempo comes for free. We therefore build key-dependent models using the key information already contained in the symbolic data. We define major/ minor keys for each pitch class, resulting in 24 different HMMs. After building 24 key models, $\lambda_k, 1 \leq k \leq 24$, we simultaneously perform key estimation and chord recognition of unknown input as follows: first, given acoustic input, we extract the observation sequence $O = \{O_1O_2 \cdots O_T\}$ of appropriate feature; then, we calculate the model likelihoods for all 24 key-dependent models, $P(O,Q|\lambda_k), 1 \leq k \leq 24$; we then estimate the key by selecting the key model whose likelihood is highest, i.e.,

$$k^* = \underset{1 \leq k \leq 24}{\arg\max}\, P(O,Q|\lambda_k). \tag{3}$$

By using the Viterbi algorithm in (3), however, we not only estimate the key $k^*$, but we also obtain the optimal state path $Q^* = \{Q_1Q_2 \cdots Q_T\}$, which is our estimate of the frame-level chord sequence. This process is illustrated in Fig. 8.

In the same manner we estimate observation distribution parameters for each chord, we also transpose all major or minor keys to a key of interest before estimating the model parameters because the number of instances in each key in the training sets varies significantly from key to key, as is shown in Fig. 3. This key-transposition technique helps us fit more accurate models because we have more data.

Fig. 9 contrasts the transition probabilities from a C major chord for HMMs based on C major and C minor keys for classical data. They share some general properties like the high transition probability to the same chord, and the relatively high probability to the harmonically close chords such as dominant (G) or subdominant (F) chord. The most prominent distinction is found in the higher transition probability to F minor chord in the C minor key HMM than in the C major key HMM. This is because the fourth degree (F) or the subdominant degree is defined as a minor chord in the C minor key context, and therefore it is much more likely to occur than in the C major key context where the fourth degree is defined as a major chord. Similar distinctions are found in the Beatles' training data set.

Such key-specific transition probability matrices help us make a correct decision, particularly in situations where the observation feature vector can cause confusion. For example, those chords in relations, such as parallel major/minor or relative major/minor, share two notes in common, and thus their observation vectors look quite similar, which may cause great confusion. Discriminating the transition probabilities even from the same chord by using key-specific HMMs helps avoid mis-recognition caused by the confusion described above.

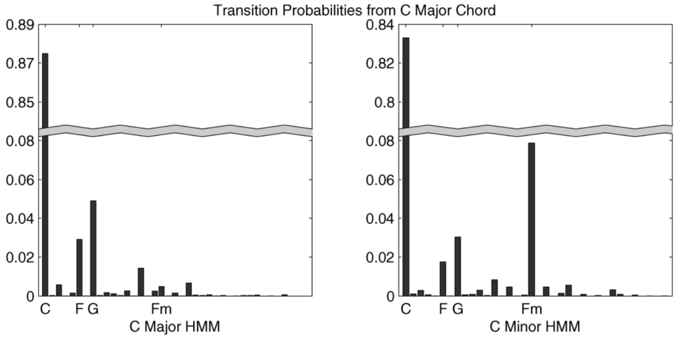

Fig. 9. Transition probabilities from a C major chord in C major key and in C minor key HMM from classical data. The $X$ axis is labeled in the order of major, minor, and diminished chords. Compare this to the generic model shown in Fig. 5.

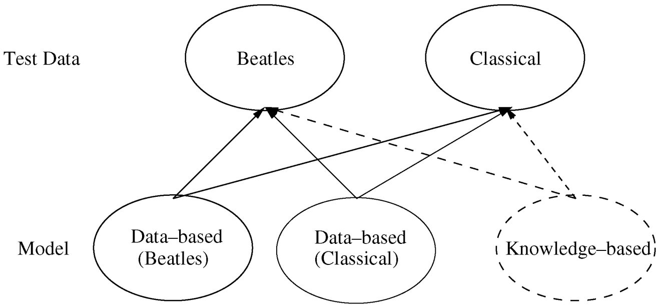

Fig. 10. Cross-evaluation between data-based and knowledge-based model.

Even with key-dependent HMMs, however, we use mean feature vectors and covariance matrices that are obtained from the universal, key-independent HMM because we believe the chord quality remains the same, independent of key context. For instance, the sonic quality of a C major chord in C major key will be the same as that of a C major chord in A minor key. What differs in each key is the distribution of chords, as well as their transition probabilities.

Although changes in key within a piece of music, or modulations, are not rare in Western tonal music, we did not take

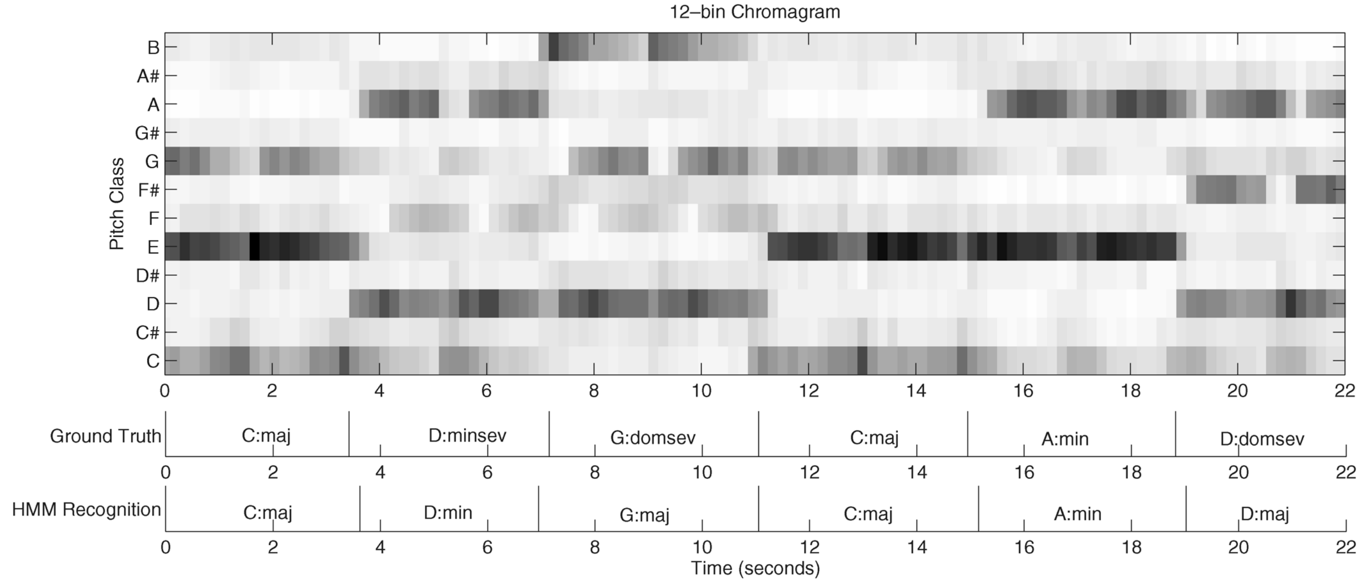

Fig. 11. Frame-rate recognition results for Bach's Prelude in C Major performed by Glenn Gould. Below 12-bin chromagram are the ground truth and the recognition result using a C major key HMM trained on classical symbolic music.

TABLE I
TEST RESULTS FOR VARIOUS MODEL PARAMETERS (% CORRECT)

| Model | Training Set | Feature | Test Set | | | |
|---|---|---|---|---|---|---|
| | | | Beatles | | Classical | |
| | | | CD1 | CD2 | Bach | Haydn |
| Data-based | Beatles | Chroma | 58.56 (57.89) | 78.21 (78.38) | 70.92 (71.45) | 56.20 (55.74) |
| | | Tonal Centroid | 66.89 (**67.50**) | 83.87 (**85.24**) | 75.03 (71.18) | 69.07 (66.77) |
| | Classical | Chroma | 51.82 (52.46) | 78.77 (79.35) | 88.31 (88.31) | 67.69 (68.61) |
| | | Tonal Centroid | 63.81 (64.79) | 81.73 (83.19) | 94.56 (**94.69**) | **71.98** (70.90) |
| Knowledge-based | N/A | Chroma | 58.96 | 74.78 | 83.40 | 69.37 |

them into account in building the models because modulations occur mostly between harmonically closely related keys such as parallel, relative major/minor keys or those in fifth relation, and therefore do not cause significant alterations in chord distribution or progression characteristics.

## V. EXPERIMENTAL RESULTS AND ANALYSIS

### A. Evaluation

We test our models' performance on two types of musical audio. First, we used Bach's keyboard piece (*Prelude in C Major*) and Haydn's string quartet (*Op.3, No.5: Andante*, mm.1—46) as a test set of classical music. For these test data, the authors perform harmony analysis to obtain the ground-truth annotation. For a more complete test, we then test our models on the two whole albums of the Beatles (CD1: *Please Please Me*, CD2: *Beatles For Sale*) as done by Bello and Pickens [10]. Ground-truth annotations are provided by Harte and Sandler at the Digital Music Center at the University of London in Queen Mary.[5] We reduce the class size from 36 to 24 by discarding the 12 diminished chords for the Beatles' test set since they rarely appear in rock music. In computing frame-level scores, we only count exact matches as correct recognition.

We measure the performance of the models in several configurations. First, because we have two separate parameter sets trained on two different training data sets (classical and the Beatles), we perform tests for each parameter set to measure how each model's performance changes with training data. None of the symbolic files corresponding to the test audio is included in the training data sets. Second, we compare two feature sets—chroma feature and tonal centroid feature. In addition, we compare the performance of a universal, key-independent model with that of a key-dependent model. Finally, we perform cross-evaluation in order to fully investigate how our data-based model performs compared with the knowledge-based model. Fig. 10 illustrates how these models are cross-evaluated.

### B. Results and Discussion

Fig. 11 shows the first 22 s of a 12-bin chromagram of Bach's *Prelude in C Major* performed by Glenn Gould. Below the chromagram are the ground truth and the chord recognition results using a C major key HMM trained on classical symbolic music. As shown, chord boundaries, as well as chord names, are almost identical to those of ground truth except that our system classifies the dominant seventh chords as major chords with the same root, which is consistent with our definition of chord classes.

Table I shows the frame-level accuracy in percentage for all the test data for various model parameters. In order to show that the data-based model performs better than, or comparably to, the knowledge-based model, we include the frame-rate results without beat-synchronous analysis on the same Beatles' data

[5] [Online]. Available: http://www.elec.qmul.ac.uk/digitalmusic

by Bello and Pickens [10]. Because the results of the knowledge-based model on the classical test data are not available, however, we simulate them by combining the knowledge-based output distribution parameters with the transition probabilities learned from our training data without adaptation. We test this method on the Beatles' test data; the difference in results is less than 1% compared with the results of the original model [10], which uses fixed output distribution parameters and adapts the transition probabilities for each input. We therefore believe that the results on the classical data too are not significantly different from what would be obtained with the original knowledge-based model. Results in parenthesis are obtained using a key-dependent model; the best result for each test material is in boldface. Excluding Haydn's string quartet, all the best results use a tonal centroid vector and key-dependent model. This is encouraging in that the results are consistent with our expectations. If we take a closer look at the numbers, however, we find a few items worthy of further discussions.

First of all, we observe a strong dependence on the training set, especially with classical test data. This is because the model parameters, i.e., observation distribution and transition characteristics are different for the two distinct musical styles. We notice such a genre dependency in our earlier work [3]. We also find that the model trained on classical data is more robust to the change in musical genre of the test input. That is, the classical model performs equally well on both test sets while the performance of the Beatles' model drops sharply when a different style of music is used as an input. We believe this is because the model trained only on the Beatles' music fails to generalize; it fails because it is only trained on music by one specific artist and the training data set is small. On the other hand, the classical model is trained on a larger training data set by more than eight composers, and thus performs equally well on both test data sets.

Second, we can see that the tonal centroid feature performs better than the chroma feature. As we mentioned earlier, a possible explanation for this is because the tonal centroid vector is obtained by projecting the 12-bin chroma vector only on specific interval relations, like fifths and major/minor thirds; thus, it is more suitable and robust for identifying musical chords since these interval relations define most chords in Western tonal music.

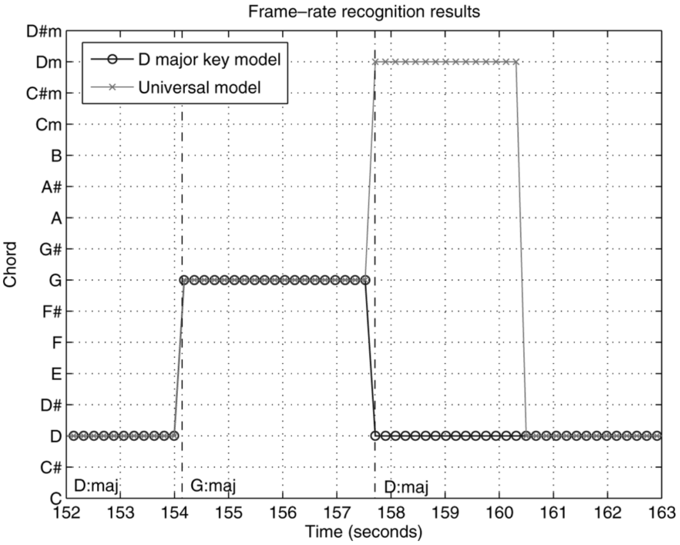

Fig. 12. Frame-rate recognition results from the Beatles' *Eight Days A Week*. In circles are the results of D major key model, and in x's are those of universal, key-independent model. Ground-truth labels and boundaries are also shown.

TABLE II
TRANSITION PROBABILITIES FROM G MAJOR TO D MAJOR AND D MINOR CHORD IN EACH MODEL

| Model | Transition Probability | |
|---|---|---|
| | G:maj→D:maj | G:maj→D:min |
| D major key (A) | 0.0892 | 0.0018 |
| Key-independent (B) | 0.0332 | 0.0053 |
| Ratio (A/B) | 2.67 | 0.34 |

Finally, we see the overall effect that a key-dependent model has on the performance. Except for a few cases, we find that a key-dependent model always increases performance. As mentioned in Section IV-D, chord progression is based on the musical key; therefore, knowing the key helps determine which chord is more likely to follow. Such an example is illustrated in Fig. 12. This figure shows an excerpt of frame-level recognition results of the Beatles' song *Eight Days A Week*, which is in the key of D major. The results of the D major key model are shown with circles, and those of a key-independent model are indicated with x's. We observe that a key-independent model makes a wrong transition from G major to D minor chord near 158 s, while the D major key model correctly switches to D major chord. As mentioned, D major and D minor chord have a parallel major/minor relation. They share two chord tones—tonic and dominant or D and A—which makes the observation vector of those chords look similar, and thus causes similar output probabilities. In the D major key, however, scale degree 1 or D is a tonic center and is defined as a major chord, although it is not impossible to use a D minor chord. Therefore, a D major chord occurs more often in D major key than for example, in the C major or in the D minor key, resulting in higher transition probability to it than to other chords. For the same reason, since a D minor chord is rarely used in the D major key, it is less probable. This is clearly indicated in Table II, which shows transition probabilities learned from the data. As shown, a transition from G major to D major chord is almost three times more likely in the key-specific model than in the key-independent model, while a G major to D minor chord transition is three times less likely.

Our results compare favorably with other state-of-the-art systems by Harte and Sandler [7] and by Bello and Pickens [10]. Using the same Beatles' test data set, Harte and Sandler obtain frame-level accuracy of 53.9% and 70.8% for CD1 and CD2, respectively. They define 48 different triads including augmented triads, and use 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' knowledge-based system yields the performance of 68.55% and 81.54% for CD1 and CD2, respectively, after they go through a preprocessing 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, as is shown in Table I.

In computing the frame-level accuracy shown in Table I, we count only exact matches as correct. However, we believe it is more accurate to measure performance with a tolerance of one frame. In other words, if a detected frame boundary or its neighbor is equal to the ground truth, we classify it as a correct match. This assumption is fair since the segment boundaries are generated by humans listening to audio, and thus they are not razor sharp. Using this error metric, the accuracy of our key-dependent tonal centroid models rises to 69.15% and 86.66% for the Beatles' CD1 and CD2, respectively.

We also performed a quantitative evaluation on the key estimation algorithm. We compared our results to manually labeled ground truth for the Beatles' test set [18]. Of all the 30 pieces (28 Beatles and two classical) in the test set, our system correctly estimated 29 of them, achieving an accuracy of 97%. The only song our system mis-recognized was *A Taste of Honey*, which is in the key of F♯ minor. Our algorithm recognized it as a E major key instead, which is not a related key. One possible explanation is that the extensive use of E, A, and B major chords strongly implies the key of E major because those chords form the most important functions intonal harmony, namely tonic, subdominant, and dominant of E major key. The 97% accuracy for finding key from audio is very encouraging, although the test set was small.

## VI. Conclusion

In this paper, we have demonstrated that symbolic music data, such as MIDI files, can be used to train machine-learning models like HMMs, with a performance that matches the best knowledge-based approach. The key idea behind our data-based approach was the automatic generation of labeled training data to free researchers from the laborious task of manual annotation.

In order to accomplish this goal, we used symbolic data to generate label files, as well as to synthesize audio files. The rationale behind this idea was that it is far easier and more robust to perform harmony analysis on the symbolic data than on the raw audio data since symbolic music files, such as MIDI, contain noise-free pitch information. In addition, by using a sample-based synthesizer, we created audio files that have harmonically rich spectra as in real acoustic recordings. This nearly labor-free procedure to obtain labeled training data enabled us to build richer models like key-dependent HMMs, resulting in improved performance.

As feature vectors, we first used conventional 12-bin chroma vectors, which have been successfully used by others in chord recognition. In addition, we tried another feature set called tonal centroid which yielded better performance.

Each state in our HMMs was modeled by a multivariate, single Gaussian completely represented by its mean vector and a diagonal covariance matrix. We defined 36 classes or chord types in our models, which include for each pitch class three distinct sonorities—major, minor, and diminished. We treated seventh chords as their corresponding root triads and disregarded augmented chords since they very rarely appear in Western tonal music.

Based on the close relationship between key and chord in Western tonal music, we built 24 key-dependent HMMs, one for each key. Given an acoustic input, our system performed a feature analysis and a sequence of observation was taken as in input to the key-dependent HMMs. Using a Viterbi decoder, we estimated the key by selecting the model with the maximum likelihood; at the same time, we recognized frame-level chord sequence because it is the same as the optimal state path in a selected key model. Experimental results showed that a key-dependent model not only gives the key information of an input, but also increases the chord recognition accuracy.

In order to examine the generality of our approach, we obtained two different model parameters trained on two musically distinct data sets. Experiments with various kinds of unseen test input, all real acoustic recordings, showed that there is positive correlation between the test and the training data. In other words, the results were better when the test and training data are of the same kind. This dependency on the training set, however, was less significant when the size of the training set was larger. This in turn suggests that we can generalize our model with even larger amount of training data.

Bello and Pickens showed approximately an 8% performance increase using beat-synchronous analysis. While there is some chance for increased errors if beat-tracking is done incorrectly, we believe that this result is orthogonal to the arguments presented in this paper. Thus, a state-of-the-art system for chord recognition should combine the data-driven approach described here, with tonal centroid features and beat-synchronous analysis.

In the near future, we plan to build higher order HMMs because chord progressions based on Western tonal music theory have 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 (GMMs) in order to better represent the emission probabilities as we increase the size of training data even more.

In addition, we will also consider discriminative HMMs, where we compute output probabilities using a discriminative model, such as SVMs, instead of a generative model like Gaussian models.

## Acknowledgment

The authors would like to thank J. Smith, J. Berger, and J. Bello for fruitful discussions and suggestions regarding this research. The authors would also like to thank the anonymous reviewers for their insightful comments that helped improve the quality of this paper.

## References

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

[2] C. A. Harte, M. B. Sandler, and M. Gasser, "Detecting harmonic change in musical audio," in *Proc. Audio Music Comput. Multimedia Workshop*, Santa Barbara, CA, 2006, pp. 21–26.

[3] K. Lee, "A system for automatic chord recognition from audio using genre-specific hidden Markov models," in *Proc. Int. Workshop Adaptive Multimedia Retrieval*, Paris, France, 2008, LNCS, Springer-Verlag, accepted for publication.

[4] K. Lee and M. Slaney, "A unified system for chord transcription and key extraction using hidden Markov models," in *Proc. Int. Conf. Music Inf. Retrieval*, Vienna, Austria, 2007, pp. 245–250.

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

[6] T. Fujishima, "Realtime chord recognition of musical sound: A system using common lisp music," in *Proc. Int. Comput. Music Conf.*, Beijing, China, 1999, pp. 464–467.

[7] C. A. Harte and M. B. Sandler, "Automatic chord identification using a quantised chromagram," in *Proc. Audio Eng. Soc.*, Spain, 2005, Audio Engineering Society, paper 6412.

[8] K. Lee, "Automatic chord recognition using enhanced pitch class profile," in *Proc. Int. Comput. Music Conf.*, New Orleans, LA, 2006, pp. 306–313.

[9] A. Sheh and D. P. W. Ellis, "Chord segmentation and recognition using EM-trained hidden Markov models," in *Proc. Int. Conf. Music Inf. Retrieval*, Baltimore, MD, 2003, pp. 185–191.

[10] J. P. Bello and J. Pickens, "A robust mid-level representation for harmonic content in music signals," in *Proc. Int. Conf. Music Inf. Retrieval*, London, U.K., 2005, pp. 304–311.

[11] J. Morman and L. Rabiner, "A system for the automatic segmentation and classification of chord sequences," in *Proc. Audio Music Comput. Multimedia Workshop*, Santa Barbara, CA, 2006, pp. 1–10.

[12] K. Lee and M. Slaney, "Automatic chord recognition from audio using an HMM with supervised learning," in *Proc. Int. Conf. Music Inf. Retrieval*, Victoria, BC, Canada, 2006, pp. 133–137.

[13] K. Lee and M. Slaney, "Automatic chord recognition from audio using a supervised HMM trained with audio-from-symbolic data," in *Proc. Audio Music Comput. Multimedia Workshop*, Santa Barbara, CA, 2006, pp. 11–20.

[14] J. C. Brown, "Calculation of a constant-Q spectral transform," *J. Acoust. Society Amer.*, vol. 89, no. 1, pp. 425–434, 1990.

[15] D. Huron, "The Humdrum Toolkit: Software for music research." [Online]. Available: http://www.musiccog.ohio-state.edu/Humdrum/

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

[17] D. Temperley*, The Cognition of Basic Musical Structures*. Cambridge, MA: MIT Press, 2001.

[18] A. W. Pollack, "Notes on . . . series," in *soundscapes.info*, 2000 [Online]. Available: http://www.icce.rug.nl/~soundscapes/DATABASES/AWP/awp-notes_on.shtml

**Kyogu Lee** (M'07) received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 1996, the M.M. degree in music technology from New York University, New York, in 2002, and the M.S. degree in electrical engineering from Stanford University, Stanford, CA, in 2007. He is currently pursuing the Ph.D. degree in computer-based music theory and acoustics at the Center for Computer Research in Music and Acoustics (CCRMA), Stanford University.

His research interests include the application of signal processing and machine learning techniques towards music and multimedia information retrieval.

**Malcolm Slaney** (SM'01) received the Ph.D. degree from Purdue University, West Lafayette, IN, for his work on diffraction tomography.

Since the start of his career, he has been a Researcher at Bell Labs, Schlumberger Palo Alto Research, Apple's Advanced Technology Lab, Interval Research, IBM Almaden Research Center, and most recently at Yahoo! Research, Santa Clara, CA. Since 1990, he has organized the Stanford CCRMA Hearing Seminar, where he now holds the title (Consulting) Professor. He is a coauthor (with A. C. Kak) of the book *Principles of Computerized Tomographic Imaging*, which has been republished as a Classics in Applied Mathematics by SIAM Press. He is a coeditor of the book *Computational Models of Hearing* (IOS Press, 2001).