Chord Segmentation and Recognition using EM-Trained Hidden Markov Models

Alexander Sheh and Daniel P.W. Ellis
LabROSA, Dept. of Electrical Engineering,
Columbia University, New York NY 10027 USA
{asheh79,dpwe}@ee.columbia.edu

Abstract

Automatic extraction of content description from commercial audio recordings has a number of important applications, from indexing and retrieval through to novel musicological analyses based on very large corpora of recorded performances. Chord sequences are a description that captures much of the character of a piece in a compact form and using a modest lexicon. Chords also have the attractive property that a piece of music can (mostly) be segmented into time intervals that consist of a single chord, much as recorded speech can (mostly) be segmented into time intervals that correspond to specific words. In this work, we build a system for automatic chord transcription using speech recognition tools. For features we use "pitch class profile" vectors to emphasize the tonal content of the signal, and we show that these features far outperform cepstral coefficients for our task. Sequence recognition is accomplished with hidden Markov models (HMMs) directly analogous to subword models in a speech recognizer, and trained by the same Expectation-Maximization (EM) algorithm. Crucially, this allows us to use as input only the chord sequences for our training examples, without requiring the precise timings of the chord changes — which are determined automatically during training. Our results on a small set of 20 early Beatles songs show frame-level accuracy of around 75% on a forced-alignment task.

Keywords: audio, music, chords, HMM, EM.

1 Introduction

The human auditory system is capable of extracting rich and meaningful data from complex audio signals. Machine listening research attempts to model this process using computers. In the music domain, there has been limited success when the input signal or analysis is relatively simple, i.e. single instrument, beat detection, etc. Unfortunately, for complex signals, such as ensemble performances, or more complex analyses, such as pitch transcription, the task rapidly increases in difficulty. In this paper we investigate a problem with complexity in both dimensions, chord recognition on unstructured, polyphonic, and multi-timbre audio. A system able to transcribe an arbitrary audio recording into an accurate chord sequence would have many applications in finding particular examples or themes in large audio databases, as well as enabling interesting new large-scale statistical analyses of musical content.

Our specific approach uses the hidden Markov models (HMMs) made popular in speech recognition (Gold and Morgan, 1999), including the sophisticated Expectation-Maximization (EM) algorithm used to train them. This is a statistical approach, in which the wide variety of feature frames falling under a single label is modeled as random variation that follows an estimated distribution. By making a direct analogy between the sequence of discrete, non-overlapping chord symbols used to describe a piece of music, and the word sequence used to describe recorded speech, much of the speech recognition framework can be used with minimal modification. In particular, no timing alignment is required between the chord labels and the training audio — using the constraints of the chord sequence alone, the EM approach converges to find optimal segmentations.

We draw on the prior work of Fujishima (1999) who proposed a representation of audio termed "pitch class profiles" (PCPs), in which the Fourier transform intensities are mapped to the twelve semitone pitch classes (chroma). This is very similar to the "chroma spectrum" proposed by Bartsch and Wakefield (2001). The assumption is that this representation captures harmonic information in a more meaningful way, thereby facilitating chord recognition. Fujishima's system uses nearest-neighbor classification to chord templates, and performed well on samples containing a single instrument.

Our system has parallels with the work by Raphael (2002), who also uses HMMs trained by EM to transcribe music in terms of chord labels. However, since his ultimate goal is note-level transcription, his "chord" vocabulary distinguishes between each different combination of simultaneous notes, in contrast to our approach of having a single model for "A minor" etc. This huge state space precludes direct training of models for each chord, and instead structural information about the harmonics expected for any given note combination are used to select among a relatively small set of model 'factors', from which the desired

chord models may be assembled. His system is applied to clean recordings of solo piano music.

In section 2, we describe the structure of our chord analysis system in detail. Section 3 describes the experiments we conducted to evaluate the system, by training and testing on a small collection of 20 early Beatles songs. Finally, section 4 discusses our future work followed by our conclusions.

2 System

The chord recognition system is presented below.

First the input signal is transformed to the frequency domain. Then it is mapped to the PCP domain by summing and normalizing the pitch chroma intensities, for every time slice. These features are then used to build chord models via EM. Finally, chord alignment/recognition is performed with the Viterbi algorithm.

2.1 Pitch Class Profile Features

Monophonic music recordings $x[n]$ sampled at 11025 Hz are divided into overlapping frames of $N = 4096$ points and converted to a short-time Fourier transform (STFT) representation,

$$X_{STFT}[k,n] = \sum_{m=0}^{N-1} x[n-m] \cdot w[m] \cdot e^{-j2\pi km/N} \qquad (1)$$

where k indexes the frequency axis with $0 \leq k \leq N-1$, n is the short-time window center, and $w[m]$ is an N-point Hanning window. The STFT is then mapped to the Pitch Class Profile (PCP) features, which traditionally consist of 12-dimensional vectors, with each dimension corresponding to the intensity of a semitone class (chroma). The procedure collapses pure tones of the same pitch class, independent of octave, to the same PCP bin; for complex tones, the harmonics also fall into particular, related bins. Frequency to pitch mapping is achieved using the logarithmic characteristic of the equal temperament scale. Our experiments use a finer grained PCP vector of 24 dimensions to give some flexibility in accounting for slight variations in tuning. A step size of 100ms, or 10 PCP frames per second, is employed. STFT bins k are mapped to PCP bins p according to:

$$p(k) = \lfloor 24 \cdot log_2(k/N \cdot f_{sr}/f_{ref}) \rfloor \bmod 24 \qquad (2)$$

where f_{ref} is the reference frequency corresponding to $PCP[0]$ and f_{sr} is the sampling rate. For each time slice, we calculate the value of each PCP element by summing the magnitude of all frequency bins that correspond to a particular pitch class i.e. for $p = 0, 1, \cdots, 23$,

$$PCP[p] = \sum_{k:p(k)=p} |X[k]|^2 \qquad (3)$$

2.2 Hidden Markov Models

PCP vectors are used as features to train a hidden Markov model (HMM) with one state for each chord distinguished by the system. An HMM is a stochastic finite automaton in which each state generates an observation. The state transitions obey the Markovian property, that given the present state, the future is independent of the past. (For an introduction to HMMs, see Gold and Morgan (1999)).

To model the PCP vector distribution for each state, we assume a single Gaussian in 24 dimensions, described by its mean vector μ_i and covariance metric Σ_i We additionally assume that the features are uncorrelated with each other, so that Σ_i consists only of variances, i.e. all off-diagonal elements are zero. To specify the model we need to determine the 24 dimension mean vector μ_i and the 24 dimension variance vector $diag(\Sigma_i)$ associated with the emitting state, and the transition probabilities.

If we knew which state (i.e. chord) generated each observation in our training data, the model parameters could be directly estimated. Hand-marked chord boundaries could provide the necessary information, but it is extremely time-consuming to create these files. In our case, we assume only that the chord sequence of an entire piece is known, but treat the chord labels of each frame as hidden values within the EM framework. This frees the researcher from the laborious and problematic process of manual annotation.

2.3 Expectation Maximization

The expectation maximization (EM) algorithm (Gold and Morgan, 1999) is an approach that structures the statistical classifier parameter estimation problem to incorporate hidden variables. We assume a joint density between the observed and missing (hidden) variables, defining the complete-data likelihood $P(X, Q|\Theta)$ where X represents the observed feature vectors, Q stands for the unknown chord labels, and Θ holds the current model parameters. EM estimates the densities by taking an expectation of the logarithm of the complete-data likelihood,

$$E[logP(X,Q|\Theta)] = \sum_{Q} P(Q|x,\Theta_{old}) log(P(X|Q,\Theta)P(Q|\Theta) \qquad (4)$$

This equation expresses the complete-data log likelihood as a function of old and new parameters, Θ_{old} and Θ. At each step the old parameters are fixed, and Θ is adjusted to maximize $logP(X, Q|\Theta)$ in expectation. This process is iterated until the expected improvement is no larger than some ϵ. EM guarantees that the estimates will improve at each step, resulting in a locally optimal set of parameters, though not necessarily the globally optimal solution. Thus, the EM solution reasonably estimates a set of parameters that maximizes the complete-data likelihood, which implements the original MAP decision rule.

The specific application of EM to find maximum-likelihood parameter estimates for a hidden Markov model is known as the Baum-Welch, or forward-backward algorithm. The update equations derived from maximizing equation 4 amount to setting model parameters to the sample averages of the training features, weighted by the posterior probability of each feature being associated with each particular hidden label, $p(q_n^i|X, \Theta_{old}, M)$, where M is the model comprising the constraints on observations X, constructed by concatenating the states specified in the known chord sequence into a single composite HMM for each song.

2.4 Viterbi Alignment

The EM algorithm calculates the mean and variance vector values, and the transition probabilities for each chord HMM. With these parameters defined, the model can now be used to determine a chord labeling for each song. The Viterbi algorithm (Gold and Morgan, 1999) is used to either forcibly align or

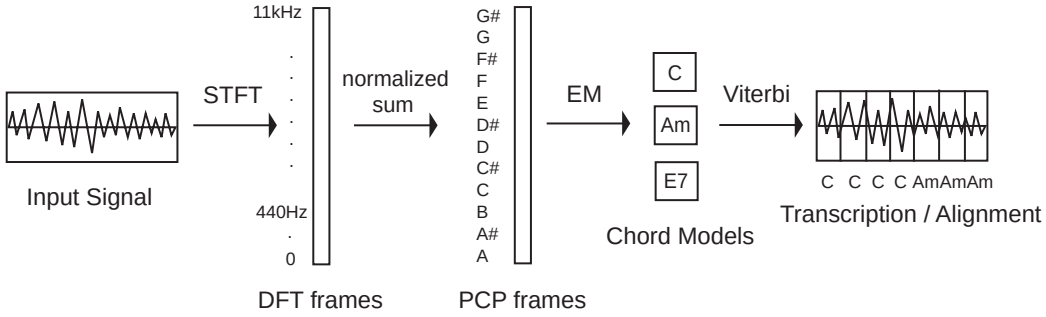

Figure 1: System Overview

recognize these labels; in forced alignment, observations are aligned to a composed HMM whose transitions are limited to those dictated by a specific chord sequence, as in training i.e. only the chord-change times are being recovered, since the chord sequence is known. In recognition, the HMM is unconstrained, in that any chord may follow any other, subject only to the Markov constraints in the trained transition matrix. We perform both sets of experiments to demonstrate that even when pure recognition performance is quite poor, a reasonable accuracy under forced alignments indicates that the models have succeeded in learning the desired chord characteristics to some extent. The output of the Viterbi algorithm is the single state-path labeling with the highest likelihood given the model parameters. This best-path assigns a chord to every 100ms time slice, resulting in a time-aligned song transcription.

2.5 Weighted Averaging of Rotated PCP Vectors

The outcome of the EM training is a set of model parameters including means and variances in PCP feature space for each of the defined chord states. These values define our initial chord models, however, an improvement can be made by calculating a weighted average of the models for every chord family (major, minor, maj7 etc.) across all root chromas (A, A#, B, C, etc.). This involves rotating the PCP vectors from each chroma until PCP[0] is the root pitch class, computing a weighted average across all the chromas (weighted by frequency of chord occurrence), then un-rotating the weighted average PCP vectors back to their original positions to construct new, regularized models for each chord. Thus, if f indexes across chord families and c is the numerical offset of each chroma relative to A in quarter tones (e.g. A $\mapsto$ 0, A# $\mapsto$ 2, B $\mapsto$ 4 etc.), then the mean vector for the parent model of chord family f in PCP space is

$$\bar{\mu_f}[p] = \frac{\sum_c \mu_{f,c}[(p-c) \bmod 24] \cdot N_{f,c}}{\sum_c N_{f,c}} \quad (5)$$

where $\mu_{f,c}$ is the original mean vector for one specific chord family/chroma combination, $N_{f,c}$ is the number of frames assigned to that state in forced alignment of the training data, and p indexes the 24 PCP bins. The rotated models then replace the individual family/chroma state models with

$$\bar{\mu_{f,c}}[p] = \bar{\mu_f}[(p+c) \bmod 24] \quad (6)$$

(Variances are similarly pooled). The motivation is that by using values characteristic to the entire family, a derived state model avoids overfitting its particular chord data. There is also the advantage of increasing each individual chord's training set to the union of all chord family members. The results below show that this simple approach gave very significant improvements.

Album	Song	Set
Beatles for Sale	Eight days a week	test
	Every little thing	test
	I don't want to spoil the party	train
	I'll follow the sun	train
	I'm a loser	train
Help	Help	train
	I've just seen a face	train
	It's only love	train
	Ticket to ride	train
	Yesterday	train
	You're going to lose that girl	train
	You've got to hide your love away	train
A Hard Day's Night	A hard day's night	train
	And I love her	train
	Can't buy me love	train
	I should've known better	train
	I'm happy just to dance with you	train
	If I fell	train
	Tell me why	train
	Things we said today	train

Table 1: Corpus of 20 early Beatles songs used in the experiments.

3 Implementation and Experiments

The Hidden Markov Model Toolkit (Young et al., 1997) was used to implement our chord recognition system. Twenty songs from three early Beatles albums were selected for our experiments (see table 1). The songs were read from CD then downsampled and mixed into mono files at 11025 Hz sampling rate. The chord sequences for each song were produced by mapping the progressions from a standard book of Beatles transcriptions (*Paperback Song Series: The Beatles*, 1995) to a simpler set of chords as shown in table 2. The twenty soundfiles and their associated chord sequences comprise the input to our system. Two songs, "Eight Days a Week" and "Every Little Thing", were

Chord families	maj, min, maj7, min7, dom7, aug, dim
Roots	A♭, B♭, C♭, D♭, E♭, F♭, G♭, A, B, C, D, E, F, G, A♯, B♯, C♯, D♯, E♯, F♯, G♯,
Examples	Amaj, C♯min7, G♭dom7

Table 2: Definition of the 147 possible chords that can appear as HMM states. The label "X" is given to chords not covered by this set. In practice, only 32 labels occurred in our data.

Feature	Align		Recog	
	train18	train20	train18	train20
MFCC	27.0	20.9	5.9	16.7
	14.5	23.0	7.7	19.6
MFCC_D	24.1	13.1	15.8	7.6
	19.9	19.7	1.5	6.9
MFCC_0_D_A	13.9	11.0	2.2	3.8
	9.2	12.3	1.3	2.5
PCP	26.3	41.0	10.0	23.6
	46.2	53.7	18.2	26.4
PCP_ROT	68.8	68.3	23.3	23.1
	83.3	83.8	20.1	13.1

Table 3: Percent Frame Accuracy results. Within each row, the first subrow refers to "Eight Days a Week" and second subrow to "Every Little Thing". Columns show the frame accuracy for forced alignment and recognition, using the train18 (excluding test cases) and train20 (including test cases) sets.

designated as the test set, and for these songs the actual chord boundaries were hand-labeled using WaveSurfer; this provided the ground-truth used to determine frame error rates.

We made separate HMM trainings using five distinct feature configurations. The first three use Mel-frequency cepstral coefficients (MFCCs), the ubiquitous features of speech recognition, calculated using HTK We included MFCCs as a comparison: We did not expect them to be well suited to this task, since these features suppress pitch information. However, MFCCs have performed surprisingly well in some other music content analysis tasks (Logan, 2000), so they make a good baseline. In each case, the total model dimensions were kept at 24 to match the number of parameters in the PCP systems; in the first case (MFCC) we used 24 MFCCs to get a relatively fine spectral description. Case 2 (MFCC_D) used just 12 MFCCs but included their deltas (velocities) since these are a popular addition in speech recognition. The third case (MFCC_0_D_A) used 7th order MFCCs including the c_0 (average log energy) term, along with deltas and accelerations for each dimension, again mimicking successful speech feature configurations.

The remaining two feature configurations are plain PCP vectors (PCP) and the averaged PCP vector rotations (PCP_ROT), both in 24 dimensions. A matlab script was written to perform a STFT of Hanning length 4096, and a subsequent PCP mapping using reference frequency 440 Hz (A4).

We trained on the 18 songs from our dataset not designated as test examples. We also repeated the experiments training on all 20 songs — i.e. including the test examples — to establish a performance ceiling in the optimistic condition when the test

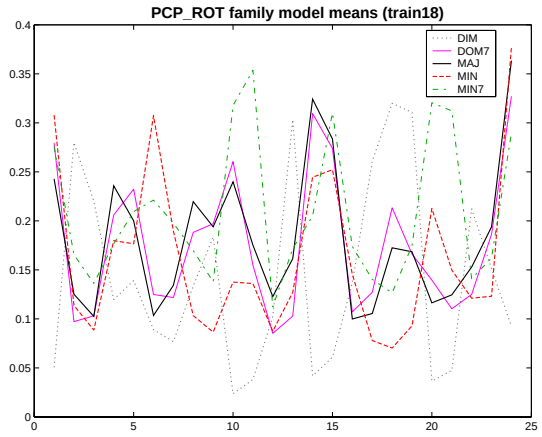

Figure 3: Mean vectors for the PCP_ROT average chord family templates.

cases exactly match part of the training set.

Training begins with the uniform segmentation and chord labeling of every training song, using chord sequence information. HMM state chord models were initialized with global mean and variance values from the entire dataset (so called flat-start EM initialization). Ideally, enough of the chord models align with the actual realizations of the chord to allow successively more accurate models to evolve during training iterations. Prior to training, a single composite HMM for each song is constructed according to the chord sequence information (see section 2.2), which constrains the training process. EM proceeds for 13 to 15 iterations. After all the training songs have been processed, the total set of statistics is used to re-estimate the parameters of the individual chord models. At this point, averaged-rotated PCP models are combined as described in section 2.5.

Lastly, the Viterbi algorithm is applied to generate either a boundary alignment of an existing chord sequence or recognize a new chord sequence. In the case of alignment, the chord sequence file is used to generate a simple composite HMM with allowable transitions determined by the song's progression. Recognition must be able to accommodate any sequence drawn from the set of training song chords. An appropriate chord loop is derived from the chord sequence files of all training songs. The Viterbi algorithm will determine the best path in this chord network. Each PCP frame is assigned a chord class such that the likelihood of the entire path is maximized. The chord-labeled frame sequence along this path is the aligned or recognized output of the system. By comparing the automatic chord labels with the hand-marked ground-truth labels of the test examples, we can calculate the frame error rate.

3.1 Frame Accuracy Results

A summary of the frame accuracy results is presented in table 3, with the alignment and recognition accuracy percentages on each of the two test examples for the five feature configurations and two training sets. We see that models trained using the base PCP features perform better than models trained using MFCCs in all cases except one (recognition of the first test example using MFCC_D and train18). Using averaged-rotated PCPs (PCP_ROT) results in models that outperform all MFCC-trained ones, as well as the base PCP models in all cases ex-

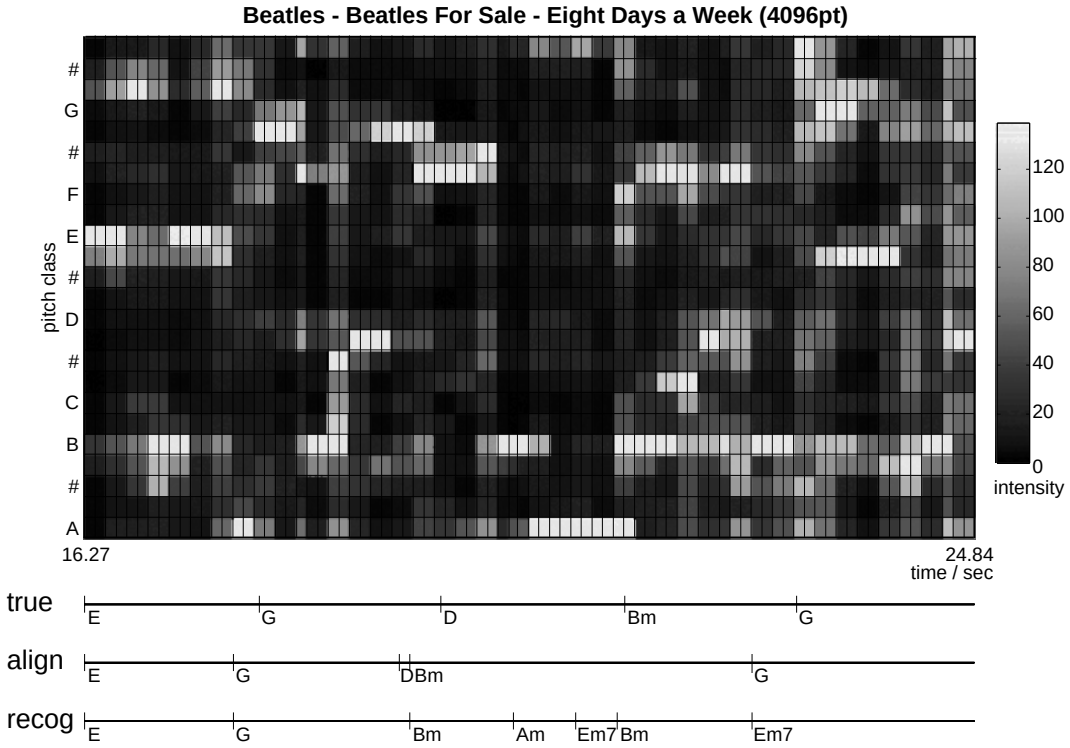

Figure 2: Illustration of PCP features vectors, ground truth labels, forced alignment output, and recognition output, for a brief segment of "Eight Days a Week".

cept, curiously, recognition based on train20. The strength of the PCP representation, and the model averaging approach, is clearly demonstrated, with the PCP_ROT models performing as much a four times better than the best MFCC counterparts.

Forced alignment always outperforms recognition, as expected since the basic chord sequence is already known in forced alignment which then has only to determine the boundaries, whereas recognition has to determine the chord labels too. Comparing the performance of train18 and train20 (i.e. testing on examples that are distinct from, or included in, the training set), we see a mixed effect with MFCC features. For the PCP system, testing on the training set (train20) gives a significant increase in accuracy for both alignment and recognition, indicating that these models are able to exploit the 'cheating' information of getting a preview of the test cases. By contrast, PCP_ROT achieves no benefit from training on the test set (and even does significantly worse on recognizing "Every Little Thing", which may reflect some pathological case in the local maximum found by EM). As a general rule, if including the test data in the training set does not significantly increase performance, we can at least be confident that the models are not overfitting the data; thus, for PCP_ROT, we could try training with more model parameters, such as Gaussian mixtures rather than single Gaussians, since we have not already overfit our models to the data—even though this is already the best-performing system overall.

3.2 Chord Confusion

Greater insight into system performance can be obtained by examining the specific kinds of errors being made in terms of misrecognitions of particular chords into other classes. The case we are most interested in is recognition (rather than alignment) using weight-averaged PCP HMMs (PCP_ROT), trained without using test songs (train18). Table 4 presents the confusion matrices for every frame in "Eight Days a Week", which we label with only 5 chords plus "X". Notice the frequent confusion between major chords and their minor version, which differ only by the semitone between the major and minor third intervals. Better discrimination of these chords might be achieved by increasing the system's frequency resolution.

3.3 Model Means

Figure 3 shows the actual PCP-domain 'signatures' — the pooled chord family mean vectors — learned in the PCP_ROT train18 system. While it is difficult to make any strong interpretation of this plot, it is interesting to see the similarities and differences between the different chords.

3.4 Output Example

Figure 2 shows an eight-second segment of the song "Eight Days a Week" taken about 16 seconds into the song. The display consists of the PCP feature vectors shown in a spectrogram-like format. Underneath are three sets of chord labels: the hand-marked ground truth, the labels obtained by forced alignment, and the labels returned by recognition (using the PCP_ROT/train18 system). While this is only a small fragment, it gives a flavor of the nature of the results obtained.

4 Future Work

4.1 Training Parameters

Our future work on this system will concentrate on the following areas:

"A MAJOR" CONFUSION MATRIX
Eight Days a Week

	Maj	Min	Maj7	Min7	Dom7	Aug	Dim
A	4	17.	.	.	49	.	.
A#/Bb	.	.	.	.	.	.	.
B/Cb	.	.	.	.	.	.	.
B#/C	.	.	.	.	.	.	.
C#/Db	.	.	.	.	.	.	.
D	.	3	.	.	.	.	.
D#/Eb	.	.	.	.	.	.	.
E/Fb	17	5	.	.	1	.	.
E#/F	1	.	.	.	.	.	.
F#/Gb	.	9	.	.	.	.	.
G	.	.	.	.	.	.	.
G#/Ab	.	.	.	.	.	.	.

"D MAJOR" CONFUSION MATRIX
Eight Days a Week

	Maj	Min	Maj7	Min7	Dom7	Aug	Dim
D	30	119	.	5	49	.	.
D#/Eb	.	.	.	.	.	.	.
E/Fb	18	34	.	41	.	.	.
E#/F	7	.	.	.	.	.	.
F#/Gb	.	42	.	15	.	.	.
G	.	.	.	.	.	.	.
G#/Ab	.	.	.	.	.	.	.
A	19	6	.	.	76	.	.
A#/Bb	.	.	.	52	.	.	.
B/Cb	.	20	.	.	.	.	.
B#/C	.	.	.	.	.	.	.
C#/Db	.	1	.	.	.	.	.

"G MAJOR" CONFUSION MATRIX
Eight Days a Week

	Maj	Min	Maj7	Min7	Dom7	Aug	Dim
G	122	35	.	.	.	.	.
G#/Ab	.	.	.	.	.	.	.
A	11	.	.	.	.	.	.
A#/Bb	.	.	.	.	.	.	.
B/Cb	.	24	.	3	.	.	.
B#/C	.	.	.	.	.	.	.
C#/Db	.	.	.	.	.	.	.
D	13	17	.	2	1	.	.
D#/Eb	.	.	.	.	.	.	.
E/Fb	1	104	.	26	.	.	.
E#/F	.	.	.	.	.	.	.
F#/Gb	.	12	.	.	.	.	.

"B MINOR" CONFUSION MATRIX
Eight Days a Week

	Maj	Min	Maj7	Min7	Dom7	Aug	Dim
B/Cb	.	72	.	.	.	.	.
B#/C	.	.	.	.	.	.	.
C#/Db	.	.	.	.	.	.	.
D	.	3	.	.	.	.	.
D#/Eb	.	.	.	.	.	.	.
E/Fb	8	51	.	41	.	.	.
E#/F	3	3	.	.	.	.	.
F#/Gb	.	12	.	6	.	.	.
G	2	.	.	.	.	.	.
G#/Ab	.	.	.	.	.	.	.
A	.	.	.	.	1	.	.
A#/Bb	.	9	.	.	.	.	.

"E MAJOR" CONFUSION MATRIX
Eight Days a Week

	Maj	Min	Maj7	Min7	Dom7	Aug	Dim
E/Fb	158	115	.	9	.	.	.
E#/F	9	.	.	.	.	.	.
F#/Gb	.	.	.	11	.	.	.
G	3	.	.	.	.	.	.
G#/Ab	.	.	.	.	.	.	.
A	9	.	.	.	1	.	.
A#/Bb	.	.	.	.	.	.	.
B/Cb	8	.	.	.	,	.	.
B#/C	.	.	.	.	.	.	.
C#/Db	.	20	.	.	.	.	.
D	.	14	.	.	.	.	.
D#/Eb	.	.	.	.	.	.	.

"X CHORD" CONFUSION MATRIX
Eight Days a Week

	Maj	Min	Maj7	Min7	Dom7	Aug	Dim
A	19	.	.	.	.	.	.
A#/Bb	.	.	.	.	.	.	.
B/Cb	.	21	.	.	.	.	.
B#/C	.	.	.	.	.	.	.
C#/Db	.	.	.	.	.	.	.
D	.	35	.	.	.	.	.
D#/Eb	.	.	.	.	.	.	.
E/Fb	.	.	.	.	.	.	.
E#/F	.	.	.	.	.	.	.
F#/Gb	.	1	.	.	.	.	.
G	.	.	.	.	.	.	.
G#/Ab	.	.	.	.	.	.	.

Table 4: Confusion matrices for recognition of “Eight Days a Week”, PCP_ROT, train18. Enharmonic equivalent chords have been combined.

- **More data and parameters:** In section 3.1, we noted that the PCP_ROT chord family models show no signs of overfitting, so employing more parameters, e.g. by using Gaussian mixture models rather than single Gaussians, should achieve further accuracy improvements. Of course, a larger and more diverse collection of training data should also improve accuracy and applicability of the system. The most significant obstacle to obtaining this data is finding a reliable source for the associated chord sequences.

- **Frequency Resolution**: As observed from the major/minor chord confusion, our recognition system most likely does not have enough frequency resolution. A simple remedy is to use longer FFT windows; increasing the Hanning length to 8192 may allow the system better to distinguish neighboring notes. This issue is particularly serious at low frequencies, when the spacing of adjacent FFT bins becomes greater than one quarter-tone. Currently we assign all energy in these low bins to a single chroma, but better results might be obtained by spreading it across several PCP dimensions in proportion to their overlap with the FFT bin frequency range.

- **Adaptive Tuning**: One argument for using 24-dimensional PCP vectors was to accommodate slight variations in tuning. Another way to help ensure that notes do not adversely interact with the PCP bin edges would be to estimate the precise tuning used in a particular song, and center the PCP feature definition accordingly. This can be accomplished by performing a much finer FFT on long portions of the original file, determining the most intense frequency and its corresponding pitch, and then shifting the FFT to PCP mapping such that this frequency falls precisely in the middle of a note.

- **Different Features**: We chose PCP features because of their prior success in chord classification tasks, but we are also interested in very different kinds of features. One idea we would like to try is looking at the autocorrelation of subband energy envelopes at the very long lags that emerge as the least common multiple of the different fundamental frequencies making up a chord.

5 Conclusion

Our experiments show that HMM models trained by EM on PCP features can successfully recognize chords in unstructured, polyphonic, multi-timbre audio. This is a challenging instance of extracting complex musical information from a complex input signal has many practical applications, since harmonic information covers much of the character of western music. Because our system uses only the raw audio. it should be applicable over a wide range of circumstances.

Although recognition accuracy is not yet sufficient to provide usable chord transcriptions of unknown audio, the ability to find time alignments for known chord progressions may be useful in itself. Moreover, the minimally-supervised EM training approach means that the incorporation of large amounts of additional training data should be straightforward, since no manual annotation is required. A larger system of this kind should result in much more precise and successful models.

Acknowledgments

Our thanks go to the anonymous reviewers for their helpful comments.

References

Bartsch, M. A. and Wakefield, G. H. (2001). To catch a chorus: Using chroma-based representations for audio thumbnailing. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Mohonk, New York.

Fujishima, T. (1999). Realtime chord recognition of musical sound: A system using common lisp music. In *Proc. ICMC*, pages 464–467, Beijing.

Gold, B. and Morgan, N. (1999). *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*. John Wiley & Sons, Inc.

Logan, B. (2000). Mel frequency cepstral coefficients for music mode ling. In *Proc. Int. Symposium on Music Inform. Retriev. (ISMIR)*, Plymouth.

Paperback Song Series: The Beatles (1995). Hal Leonard Corporation.

Raphael, C. (2002). Automatic transcription of piano music. In *Proc. Int. Symposium on Music Inform. Retriev. (ISMIR)*, Paris.

Young, S., Odell, J., Ollason, D., Valtchev, V., and Woodland, P. (1997). *HTK Hidden Markov Model Toolkit*. Entropic Research Laboratories Inc., Cambridge University.