CHORD RECOGNITION USING DURATION-EXPLICIT HIDDEN MARKOV MODELS

Ruofeng Chen **Weibin Shen** **Ajay Srinivasamurthy**
Georgia Tech Center for Music Technology
{ruofengchen, weibin_shen, ajays}@gatech.edu

Parag Chordia
Smule Inc.
parag@smule.com

ABSTRACT

We present an audio chord recognition system based on a generalization of the Hidden Markov Model (HMM) in which the duration of chords is explicitly considered - a type of HMM referred to as a hidden semi-Markov model, or duration-explicit HMM (DHMM). We find that such a system recognizes chords at a level consistent with the state-of-the-art systems – 84.23% on Uspop dataset at the major/minor level. The duration distribution is estimated from chord duration histograms on the training data. It is found that the state-of-the-art recognition result can be improved upon by using several duration distributions, which are found automatically by clustering song-level duration histograms. The paper further describes experiments which shed light on the extent to which context information, in the sense of transition matrices, is useful for the audio chord recognition task. We present evidence that the context provides surprisingly little improvement in performance, compared to isolated frame-wise recognition with simple smoothing. We discuss possible reasons for this, such as the inherent entropy of chord sequences in our training database.

1. INTRODUCTION AND BACKGROUND

The problem of audio chord recognition has been explored for over a decade and thus there exists an established basic framework that is widely used. First, a chroma feature, or its variation is computed, followed by classifiers or sequential decoders to recognize the chord sequence. Certain enhancements to the basic feature computation and classification algorithms have been found to be useful.

The chromagram, a sequence of 12-dimensional vectors that attempts to represent the strength of each pitch class, is computed using a log-frequency spectrogram, estimated with a constant-Q transform (CQT). Several methods have been proposed to refine the basic features. Non-Negative Least Square estimation [1] and Harmonic Product Spectrum [2] reduce the power of non-octave overtones in the spectrum. Harmonic-Percussive Source Separation [3] decreases the power of percussive sounds, which typically do not contain chord information [4], [5]. The CRP chroma algorithm [6] attempts to produce a timbre-invariant feature and has been applied in [5] and [7]. Background Subtraction [1] removes the running mean of a given spectrum, which is based on the same principle as the CRP chroma algorithm in that they both conduct long-pass filtering on audio signal, damping timbral information in features. Loudness-based chroma performs A-weighting [5] on log-frequency spectrogram. Besides chroma, a 6-dimensional feature called “tonnetz”, based on Neo-Riemannian theory, is also commonly used and has proven to be helpful [8], [9]. Finally, a machine-learned transformation matrix that converts log-frequency spectrogram to chromagram is shown to outperform an arbitrary transformation matrix in [10].

Structural information has also been utilized to help audio chord recognition. Many systems use beat-synchronous chromagrams that are computed over a beat or half-beat, rather than short frames [1], [9], [11], [12]. In [7], the authors smoothed the chromagram based on a regressive plot. In [11], the authors demonstrate that an attempt to find explicit repetition in a piece can improve performance.

In the domain of classifiers or decoders, many published works use Hidden Markov Models (HMMs). Recent papers have used Dynamic Bayesian Network (DBN) in conjunction with separate bass and treble chromas for recognition [1], [5]. In the past, two methods implementing key detection to assist chord recognition have been proposed. The first method builds a group of key-specific models [4], [9], while the other treats key information as one of the layers in its graphical model [1], [5]. In some cases, transition matrices were based on, or initialized, using principles from music theory rather than learned from the training set [12]. Apart from HMMs, a Pitman-Yor Language Model [13] has also been used to build a vocabulary-free model for chord recognition. Finally, another popular approach is the use of chroma templates of chords [14], [15].

In this paper, we present our approach which proposes a novel method to compute the chroma, and uses duration-explicit HMMs (DHMMs) for chord recognition. DHMMs are discussed in [16], but have rarely been used in MIR research. We also try to answer an important question: how much can transitional context knowledge (i.e. chord progressions) contribute to increasing the accuracy of the model?

This paper is organized as follows: Section 2 describes the chroma feature that we use, emphasizing on a novel way of computing chromagram; Section 3 presents the DHMM and its implementation; Section 4 evaluates our models and analyzes the contribution of duration constraints and transitional context knowledge; Section 5 presents the conclusions and sheds light on future research.

2. CHROMA FEATURE

Our chroma feature is based on the 60 dimensional log-frequency spectrogram computation proposed in [5], which uses perceptual loudness scaling of the spectrogram. We set our frame length to 512 ms with a hop size 64 ms. We only consider the energy within 5 octaves between 55 Hz and 1760 Hz. We propose a new method to compute the chromagram from the spectrogram.

2.1 Chroma Based On Linear Regression

Chroma is typically calculated by "folding" the spectrum into one octave and summing over frequencies corresponding to a quarter-tone around a given pitch-class. In [10], the authors show that a machine-learned transformation matrix outperforms this basic method. We developed a method with similar motivation. The ground truth chord label is converted into a 24 dimensional chroma template logical vector, where the first 12 dimensions represent whether one of the 12 notes exists in the bass label, and the last 12 dimensions represent whether one of the 12 notes exists in the chord label. For an example, a "C:maj/5" is converted to
$[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0]$.

The target vectors are the chroma templates and we fit a transformation matrix which converts the log-frequency spectrum to a chroma vector that is as close to the chroma template as possible. Similar to [10], we explored the use of neural networks, experimenting with various activation functions including sigmoid, logistic, linear and quadratic. We found that sigmoid and logistic functions compress the outputs, leading to additional confusion between 0 and 1. Linear and quadratic regressions return nearly the same results without compressing the output. Consequently, we chose linear regression to fit the transformation matrix. The regressed matrix shown in Figure 1 transforms a 60 dimensional log-frequency spectrum into a 24 dimensional chroma vector, which is a concatenation of the bass and treble chroma. It is worth noticing that the transformation matrix has weights for both base frequency and harmonics, leading to a certain degree of inhibition of harmonics in the final chroma. We have additionally tried some other proposed methods to inhibit harmonics (e.g. NNLS [1], HPS [2]), but none of them returned an improvement on the final results. Since the matrix relies on ground truth chord labels in the training set, the linear regression is performed every time we train the model.

2.2 Tonnetz Feature for Bass

The chromagram we have obtained through the aforementioned method is not full rank. This is because the information about the root note occurs in both bass and treble chroma templates. Given the common knowledge that bass chroma contains less chord related information than treble chroma, it might be more suitable for a lower-dimensional representation. So, we convert a bass chroma (i.e. the first 12 dimensions) into a 6-dimensional tonnetz feature, described in detail in [8]. We explored converting the treble chroma into tonnetz, but we did not observe any improvement on accuracy, which is consistent with [5]. Our feature vector is thus 18 dimensional, consisting of the treble chroma and the tonnetz features from bass chroma.

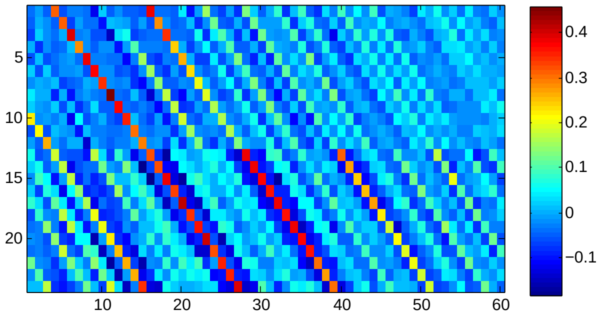

Figure 1. The transformation matrix for calculating chroma from CQTed-spectrum.

2.3 Beat Level Feature

In popular music, chord changes usually occur at beat and bar boundaries and thus, beat information can be useful in determining likely boundaries. We beat-tracked the audio using the dynamic programming approach of Ellis [17]. However, we extended the algorithms to allow for tempo variation within a song. A dynamic programming process was used to estimate the tempo before finding beat positions, similar to a method described by [18]. This resulted in a slight improvement in chord recognition accuracy.

We explored three approaches to calculate beat level features: (1) calculate chroma on the entire beat (large window for CQT); (2) calculate chroma on the frame level, then average over the frames in the beat; (3) calculate chroma on frame level, then take the median of each dimension within a beat. We found that approach (2) worked the best. In our experiments, we explore the use of both the frame level and beat level features in the HMMs.

3. DURATION-EXPLICIT HIDDEN MARKOV MODEL (DHMM)

In this section, we present a detailed discussion of the DHMM and its implementation, at the beat level. DHMMs estimate the chord sequence by simultaneously estimating chord labels and positions, which can be thought of as estimating on the chord level (See Figure 2). Initially, we applied DHMM hoping to reveal transitional context knowledge since at the frame and the beat level, self-transition is dominant. However, as we show in section 4.2, transitional context knowledge is not as important as we had hypothesized. Yet, modeling the duration of chords contributes to the majority of our improvement.

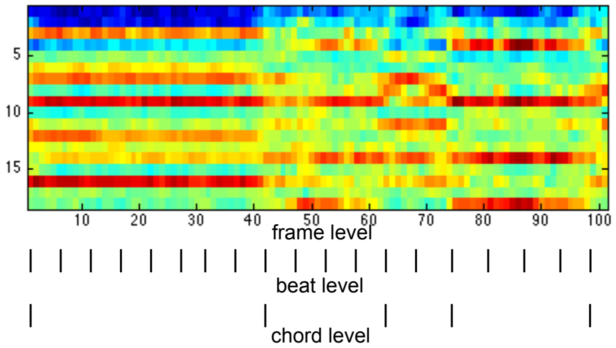

Figure 2. Frame level, beat level and chord level. Horizontal axis is frame level feature vector index; vertical axis is feature dimension index.

We adopt the notation used in [16]. To better understand the following expressions, readers are encouraged to briefly review III.A and III.B of [16]. In chord recognition, T is the number of beats in a song; observation sequence $O = \{O_1 O_2 \ldots O_T\}$ is the time sequence of feature vectors; state sequence $Q = \{q_1 q_2 \ldots q_T\}$ is the hidden chord label sequence; N is the number of chords being considered (i.e. the total number of states of the HMM); $S_1 S_2 \ldots S_N$ are possible states; $q_t \in \{S_1 S_2 \ldots S_N\}$; π is the vector of initial probabilities; $A = \{a_{ij}\}$ is the chord transition matrix, which denotes the probabilities of transitions from S_i to S_j; $B = \{b_i(O_t)\}$ is the emission matrix, which denotes the probabilities of emitting O_t from S_i; $p = \{p_i(d)\}$ is the duration distribution, which denotes the probabilities of S_i spanning d beats.

The model λ comprises of π, A, B and p. In our experiments, we found that π is unimportant so we set it to uniform distribution. A is trained by counting all the chord changes. A small value (0.05) is added to the diagonal of A before normalization, in order to bring the Viterbi algorithm back to sync when the actual duration has zero probability in p. A multivariate normal distribution is trained for each chord in order to calculate B. p is computed by counting the durations (i.e. number of beats) of each chord. The same duration distribution is used for all chords. However, the notation $p_i(d)$ is retained for better generalization. We limit the maximum duration D to 20 beats.

3.1 Viterbi Algorithm

Viterbi algorithm is a dynamic programming algorithm that finds the globally optimal state sequence $Q^* = \{q_1^* q_2^* \cdots q_T^*\}$ explaining an observation sequence, given the model λ. We denote

$$\delta_t(i) = \max_i P(S_1 S_2 \ldots S_i \text{ ends at } t|\lambda)$$

A. Initialization ($t \leq D$):

$$\delta^* = \max_{d=1}^{t-1} \max_{j=1, j\neq i}^{N} \delta_{t-d}(j) a_{ji} p_i(d) \prod_{s=t-d+1}^{t} b_i(O_s)$$

$$\delta_t(i) = \max\{\pi_i p_i(t) \prod_{s=1}^{t} b_i(O_s), \delta^*\}$$

B. Recursion ($D < t \leq T$):

$$\delta_t(i) = \max_{d=1}^{D} \max_{j=1, j\neq i}^{N} \delta_{t-d}(j) a_{ji} p_i(d) \prod_{s=t-d+1}^{t} b_i(O_s)$$

In addition to $\delta_t(i)$, we need two other variables: $\psi_t(i)$ to track the last optimal state, and $\phi_t(i)$ to track optimal duration. If S_i ends at t, the optimal duration of S_i would be $\phi_t(i)$, and the optimal last state would be $\psi_t(i)$.

In initialization and recursion, we can get the index $\hat{j}$ and $\hat{d}$ that produce $\delta_t(i)$, then

$$\psi_t(i) = \hat{j}$$
$$\phi_t(i) = \hat{d}$$

If $\delta_t(i)$ equals $\pi_i p_i(t) \prod_{s=1}^{t} b_i(O_s)$ in initialization, then

$$\psi_t(i) = i$$
$$\phi_t(i) = t$$

C. Termination:

$$q_T^* = \arg \max_{1 \leq i \leq N} \delta_T(i)$$

D. Backtracking:

```
$d = \phi_T(q_T^*)$
$t = T$
while $t > d$ do
    $q_{t-d+1} \ldots q_{t-1} = q_t$
    $q_{t-d} = \psi_t(q_t)$
    $t = t - d$
    $d = \phi_t(q_t)$
end while
$q_1 \ldots q_{t-1} = q_t$
```

3.2 Probability of Observation Sequence

In some cases, it is necessary to know $P(O|\lambda)$, the probability that a model λ generates an observation sequence O. The computation of this probability is detailed in [16]. We applied the scaling method to prevent the probability from going below the machine precision.

The forward variable is defined as

$$\alpha_t(i) = P(O_1 O_2 \ldots O_t, S_i \text{ ends at } t|\lambda)$$

A. Initialization ($t \leq D$):

$$\alpha^* = \sum_{d=1}^{t-1} \sum_{j=1, j\neq i}^{N} \alpha_{t-d}(j) a_{ji} p_i(d) \prod_{s=t-d+1}^{t} b_i(O_s)$$

$$\alpha_t(i) = \pi_i p_i(t) \prod_{s=1}^{t} b_i(O_s) + \alpha^*$$

$$c_t = \frac{1}{\sum_{s=1}^{t} \sum_{i=1}^{N} \alpha_s(i)}$$

$$\alpha_s(i) = \alpha_s(i) c_t, s = 1 \ldots t$$

B. Recursion ($D < t \leq T$):

$$\alpha_t(i) = \sum_{d=1}^{D} \sum_{j=1, j\neq i}^{N} \alpha_{t-d}(j) a_{ji} p_i(d) \prod_{s=t-d+1}^{t} b_i(O_s)$$

$$c_t = \frac{1}{\sum_{s=t-D+1}^{t} \sum_{i=1}^{N} \alpha_s(i)}$$

$$\alpha_s(i) = \alpha_s(i) c_t, s = t - D + 1 \ldots t$$

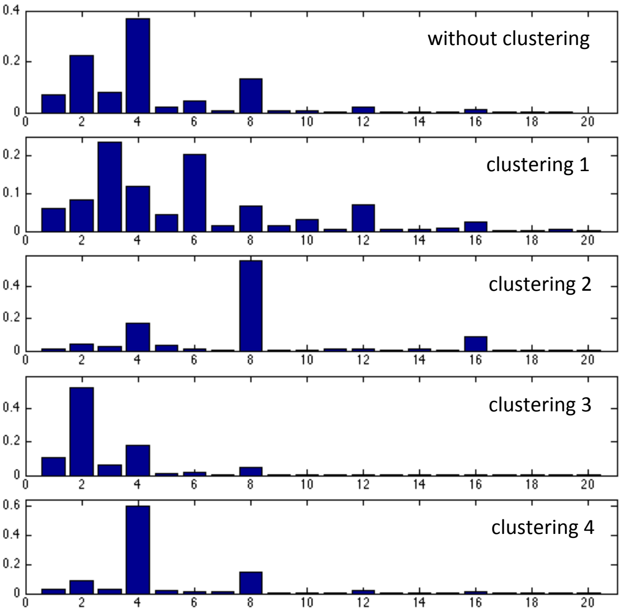

Figure 3. Top panel: Global duration distribution trained using the whole training set. Panel 2-5: Clustered duration distributions.

C. Termination:

$$\log P(O|\lambda) = -\sum_{t=1}^{T} \log c_t$$

Another precision problem (which does not exist in an ordinary HMM) is caused by $\prod b_i(O_s)$. Our solution is to divide all $b_i(O_s)$'s by the maximum value of B.

3.3 Time Signature Clustering

As will be shown in section 4.2, a global duration model has a limited contribution towards the accuracy improvement, because we train one single duration distribution using the whole training set. In fact, popular music is composed using a limited number of time signatures (e.g. 4/4, 6/8), and usually keeps its time signature unchanged for the whole length.

In other words, we train multiple duration distributions so that we have multiple models $\lambda_1, \lambda_2 \ldots \lambda_m$ (where only their p's are different), and we calculate $P(O|\lambda_1)$, $P(O|\lambda_2)$ $\ldots P(O|\lambda_m)$ and choose the model which maximizes likelihood, before running the Viterbi algorithm.

In order to train multiple duration distributions, we calculate a duration distribution for each song in the training set, and then cluster all the duration distributions into c categories using k-means algorithm (in our experiments, $c = 4$). We don't manually annotate time signatures because beat tracking algorithm is very likely to double or triple the tempo. Through clustering, we don't need to actually know the true time signature, and can account for potential errors caused by beat tracking. Figure 3 gives an example of clustered duration distributions compared to the global duration distribution.

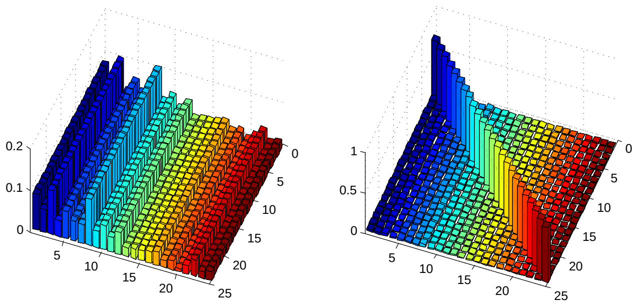

Figure 4. Left: Putting prior distribution to all rows of transition matrix. Right: Adding 3 to the diagonal of the matrix on the left and normalize each row.

4. EVALUATION

4.1 Experiments

We evaluate our models on two datasets: Beatles set by Harte [8] and Uspop set by Cho [7]. 44 songs in Uspop were excluded because we couldn't find audio of a length that matched the corresponding annotation. 12 songs were excluded from Beatles for reasons such as audio being off tune, or inconsistent time offsets of annotations. (See http://www.weibinshen.com/ISMIR2012Table.html for a full list of songs that were not used in this study).

We perform a 4-fold cross-validation experiment and report the average chord recognition accuracy on 24 major/minor chords. During training, all chord labels are remapped to 24 major/minor chords as in [5]. Each chord is trained with a single Gaussian distribution and corresponds to one state in HMM. During testing, each frame or beat is recognized as one of the major/minor chords. During evaluation, recognized labels on beat level are transformed to the frame level, and only frames with major/minor labels in the ground-truth annotations are counted for evaluation. The recognition accuracy metric is the frame-based recall rate - the number of frames that are recognized correctly, divided by the total number of frames, same as the evaluation metric used in MIREX evaluations.

In order to determine the contribution of chord progression information to the improved performance, we also baseline with Bayesian-type classifiers, where a chord prediction is determined by a MAP classifier independently at each frame or beat. We implement Bayesian-type classifiers by simply replacing every row of a transition matrix with the prior chord distribution, obtained by counting the unconditional occurrence of each chord (Figure 4).

4.2 Results

We compare the accuracy of different models in Table 1. In "Bayes", we train the transition matrix by applying the prior distribution to all the rows. In "Bayes+smooth", we apply a "majority" filter on the Bayesian classifier's output, in order to remove short-term deviations. In "Mod. Bayes", we add a relatively large number (arbitrarily, 3) to the diagonal elements of the "Bayes" transition matrix, and then normalize each row (see Figure 4-Right). In order to compare it with the state of the art, we also run the

Harmony Progression Analyser (HPA) proposed by Ni et. al [5] on the same datasets. It is a state of the art model using a three-layer HMM with key, bass note and chord estimation.

Level	Model	Uspop	Beatles
Frame level	Bayes	0.7518	0.7206
	Bayes+smooth	0.8285	0.8204
	HMM	0.8096	0.7966
Beat level	Bayes	0.7867	0.7733
	Mod. Bayes	0.8340	0.8331
	HMM	0.8365	0.8361
Duration model	Bayes	0.8371	0.8398
	DHMM	0.8377	0.8352
Time signature clustered duration model	Bayes	**0.8410**	**0.8413**
	DHMM	**0.8423**	**0.8419**
	HPA [5]	0.8401	0.8278

Table 1. A comparison of Accuracy

4.3 Analysis

We see that we achieve a performance comparable to the state of the art [5]. At the frame level, Bayesian classifier has a fairly low accuracy (75.18% on Uspop and 72.06% on Beatles). With smoothing (82.85% and 82.04%), it outperforms the basic frame-level HMM (80.96% and 79.66%). At the beat level, the Bayesian classifier attains a recognition rate of 78.67% for Uspop. However, when the self-transitions are emphasized in the “Modified Bayes” method, accuracy is on par with the beat-level HMM. Smoothing, as well as emphasizing self-transitions, essentially incorporate the knowledge that most chords last for more than a few frames or beats.

For DHMMs, duration information is decoupled from the transition matrix and results for the Bayesian classifier (83.71% and 83.98%) and the DHMM (83.77% and 83.52%) are similar. Using multiple duration models after clustering raises accuracy to 84.23% on Uspop, which is comparable the current state of the art.

The results suggest that the primary reason why HMMs are more effective than Bayesian classifier is that the strong self-transition probabilities emphasize continuity, rather than the knowledge of the chord progression represented in the transition matrix. In other words, when continuity is enforced by smoothing, or modeling durations separately, HMMs perform no better than a Bayesian classifier. Although there have been past works stating an improvement by using smoothing [22], we did not find any previous work discussing if the reason for HMMs outperforming Bayesian Classifiers is because of the smoothing effect of its transition matrix, or if the context information was really useful.

To further understand the contribution of chord progression knowledge we constructed an “oracle” condition in which the true transition matrix for a song was revealed (i.e. the transition matrix was computed using the ground truth labels for a particular song). This transition matrix was then used by the DHMM. The results are summarized in Table 2 and can be interpreted as an upper bound for chord recognition accuracy using a first-order DHMM. These results suggest that even in the case where the chord transitions are exactly known for a song, accuracy improves no more than 2%.

Level	Model	Uspop	Beatles
Duration model	Bayes	0.8735	0.8726
	DHMM	0.8863	0.8919

Table 2. Upper bound on performance

Why doesn’t knowledge of the chord progression give greater improvements? In most cases, it seems the evidence provided in the local chromagram feature is quite strong, minimizing the need for top-down information. On the other hand, when the feature is noisy or ambiguous, it seems that the the prior imposed by the transition matrix is not very strong. In other words, chords progressions are less predictable than they seem.

We tested this hypothesis by estimating the entropy of chord progressions in the training set using a Variable Length Markov Model (VLMM) [19], [20]. In other words, given the context, we tested how sure we can be of the next symbol, on an average. A VLMM is an ensemble of Markov models which effectively captures variable length patterns in the chord sequence. A VLMM was used, as opposed to a first-order Markov model, because we wanted to ensure that long patterns were captured in addition to local chord transitions. Given the true symbol sequence till the current time frame, we obtain a predictive distribution over the chord labels for the next time frame, which is used to obtain the cross entropy of the test sequence. Using a VLMM, the minimum cross-entropy obtained was 3.74 on the Uspop dataset and 3.67 on the Beatles dataset (at a maximum VLMM order 2), in a leave-one-out cross validation experiment. It was found that the cross-entropy increased beyond order 2 in both datasets. An entropy value of 3.74 corresponds to a perplexity of 13.4, which can be interpreted as the average number of symbols the system was confused between. Thus, knowing the chord history does not, in general, narrow the possibilities greatly, and is unlikely to overcome a noisy or ambiguous feature vector.

5. CONCLUSIONS

In this paper, we presented an implementation of DHMMs and applied them to the chord recognition task. This model decouples the duration constraints from the transition matrix. We then build separate models for duration distributions that indicate different time signatures to improve the duration constraint in each model. Using this method, a comparable performance to the state of the art is demonstrated.

Though duration-explicit HMMs don’t produce groundbreaking results, we believe that the proposed model may benefit other MIR tasks in the future, e.g. melody estimation and structural segmentation. Perhaps most importantly we show that state of the art results can be obtained using simple classifiers that do not use transition information. Further attempts to fully incorporate key and chord-progression knowledge (at least for popular songs of this type) using techniques such as high-order HMMs, are unlikely to yield significant improvements.

This material is based upon work supported by the National Science Foundation under Grant No. 0855758.

6. REFERENCES

[1] M. Mauch, S. Dixon: "Approximate Note Transcription For The Improved Identification Of Difficult Chords," In *Proceedings of the 10th International Conference on Music Information Retrieval*, Utrecht, Netherlands, 2010.

[2] K. Lee, "Automatic Chord Recognition Using Enhanced Pitch Class Profile," In *Proceedings of International Computer Music Conference*, New Orleans, USA, 2006.

[3] N. Ono, K. Miyamoto, H. Kameoka, S. Sagayama: "A Real-time Equalizer of Harmonic and Percussive Components in Music Signals," In *Proceedings of the 9th International Conference on Music Information Retrieval*, Philadelphia, USA, 2008.

[4] Y. Ueda, Y. Uchiyama, T. Nishimoto, N. Ono, S. Sagayama, "HMM-based Approach for Automatic Chord Detection Using Refined Acoustic Features," In *The 35th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Dallas, Texas, USA, 2010.

[5] Y. Ni, M. McVicar, R. Rodriguez and T. Bie: "An end-to-end machine learning system for harmonic analysis of music," In *IEEE Transactions on Audio, Speech and Language Processing*, In Press, 2012.

[6] M. Muller, S. Ewert, "Towards timbre-invariant audio features for harmony-based music," In *IEEE Transactions on Audio, Speech, and Language Processing (TASLP)*, 18(3):649–662, 2010.

[7] T. Cho and J. Bello, "A Feature Smoothing Method For Chord Recognition Using Recurrence Plots," In *Proceedings of the 12th International Conference on Music Information Retrieval*, Miami, Florida, USA, 2011.

[8] C. Harte, M. Sandler, M. Gasser: "Detecting Harmonic Change In Musical Audio," In *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, Volume: C, Issue: 06,Santa Barbara, California, USA, 2006.

[9] K. Lee, M. Slaney, "A Unified System for Chord Transcription and Key Extraction Using Hidden Markov Models," In *Proceedings of the 8th International Conference on Music Information Retrieval*, Vienna, Austria, 2007.

[10] O. Izmirli, R. Dannenberg: "Understanding Features and Distance Functions for Music Sequence Alignment," In *Proceedings of the International Conference on Music Information Retrieval*, Utrecht, Netherlands, 2010.

[11] M. Mauch, K. Noland, S. Dixon: "Using musical structure to enhance automatic chord transcription," In *Proceedings of the 10th International Conference on Music Information Retrieval*, Kobe, Japan, 2009.

[12] J. Bello, J. Pickens, S. Pauws "A Robust Mid-Level Rerpresentation For Harmonic Content In Music Signals," In *Proceedings of the 6th International Conference on Music Information Retrieval*, London, UK, 2005.

[13] K. Yoshi, M. Goto, "A Vocabulary-Free Infinity-gram Model For Nonparametric Bayesian Chord Progression Analysis," In *Proceedings of the 12th International Conference on Music Information Retrieval*, Miami, Florida, USA, 2011.

[14] L. Oudre, C. Fevotte, Y. Grenier, "Probabilistic Template-Based Chord Recognition," In *TELECOM ParisTech*, Paris, France, 2010

[15] T. Fujishima, "Realtime Chord Recognition of Musical Sound: a System Using Common Lisp Music," In *The 1999 International Computer Music Conference*, Beijing, China, 1999.

[16] L. Rabiner: "A Tutorial On Hidden Markov Models And Selected Applications In Speech Recognition," In *Proceeding of the IEEE*, Vol.77, No.2, February 1989.

[17] D. Ellis, "Beat Tracking by Dynamic Programming," In *Journal of New Music Research* Vol.36(1), 51-60, 2007.

[18] F. Wu, T. Lee, J. Jang, K. Chang, C. Lu, W. Wang, "A Two-Fold Dynamic Programming Approach to Beat Tracking for Audio Music With Time-Varying Tempo," In *Proceedings of the 12th International Conference on Music Information Retrieval*, Miami, Florida, USA, 2011.

[19] D. Conklin, I. H. Witten, "Multiple viewpoint systems for music prediction," In *Journal of New Music Research*, 24:51-73, 1995.

[20] M. Pearce, D. Conklin, and G. Wiggins, "Methods for Combining Statistical Models of Music," In *U. K. Wiil (Ed.), Computer Music Modelling and Retrieval (pp. 295-312)*. Heidelberg: Springer.

[21] J. A. Burgoyne, J. Wild, I. Fujinaga, "An Expert Ground-Truth Set For Audio Chord Recognition And Music Analysis," In *Proceedings of the 12th International Conference on Music Information Retrieval*, Miami, Florida, USA, 2011.

[22] T. Cho, R. J. Weiss, J. P. Bello, "Exploring Common Variations in State of the Art Chord Recognition Systems," In *7th Sound and Music Computing Conference*, Barcelona, Spain, 2010.