UNSUPERVISED AUDIO KEY AND CHORD RECOGNITION

Yun-Sheng Wang
Department of Computer Science,
George Mason University
Fairfax, USA
ywange@gmu.edu

Harry Wechsler
Department of Computer Science,
George Mason University
Fairfax, USA
wechsler@gmu.edu

ABSTRACT

This paper presents a new methodology for determining chords of a music piece without using training data. Specifically, we introduce: 1) a wavelet-based audio denoising component to enhance a chroma-based feature extraction framework, 2) an unsupervised key recognition component to extract a bag of local keys, 3) a chord recognizer using estimated local keys to adjust the chromagram based on a set of well-known tonal profiles to recognize chords on a frame-by-frame basis. We aim to recognize 5 classes of chords (major, minor, diminished, augmented, suspended) and 1 N (no chord or silence). We demonstrate the performance of the proposed approach using 175 Beatles' songs which we achieved 75% in F-measure for estimating a bag of local keys and at least 68.2% accuracy on chords without discarding any audio segments or the use of other musical elements. The experimental results also show that the wavelet-based denoiser improves the chord recognition rate by approximately 4% over that of other chroma features.

1. INTRODUCTION

The ability to extract local keys and chords from audio signals is an important step toward music transcription and segmentation using machines. Transcription of music typically requires the understanding of scale degrees used in a music piece as well as the analysis of harmony which correspond nicely to local keys and chords, respectively. On the other hand, music segmentation is the process of partitioning the target music signals into multiple sections so that each section is homogeneous within its boundary but distinct from its neighboring sections; it usually serves as an intermediate step to solve a larger problem such as content-based information retrieval. In [1], six types of segmentation cues - cadence patterns, key schemes, text, instrumentation, rhythm, and harmony - were discussed; using extracted local keys and chords, a multi-dimensional harmonic rhythm can be constructed for segmenting rock or popular music. However, the interpretation of keys and chords are often subjective [2]. For keys, the presence and exact locations of key modulations are often interpreted differently by musicians. For chords, when power chords are played, are they major or minor triads? Should a chord be extended to the 7^{th}? Due to such uncertainties, in this paper, we present a probabilistic approach to estimate, from audio signals, a "bag of local" (BOL) keys and use the extracted keys to recognize chords. Our previous work [3] adopted similar unsupervised approach in estimating keys and chords of symbolic music (MIDI); in this paper, we extend our previous approach to wave audio signals.

2. RELATED WORK

In this section, we review recent work that extracts keys and chords simultaneously from wave audio signals with concentration on those that utilized the unsupervised approach. To analyze keys or chords from audio signals, the most common front end is to transform sound waves into the frequency domain which is subsequently mapped into a chromagram to represent the energy level of the 12 pitch classes, pioneered by [4].

Most recent unsupervised estimation of local keys and chords uses a probabilistic framework [5, 6, 7] by modeling the acoustic likelihood p(X|K,C) to find the best K and C using dynamic programming search technique in 24 keys × 48 chords space. Specifically in [7], a key-chord model and state transition probabilities comprising three sub models (duration, key, and chord) was proposed using the same search space in [5]. Cosine similarity was computed between key template, proposed by [8] enhanced from the pioneering profiles [9], and observed data. The chord model determines the likelihood of observation given a chord being played. The best key-chord sequence is determined by search using the Viterbi algorithm as proposed in [6]. In [10], a probabilistic framework was also used, where the overall chord probabilities were estimated directly from the music piece using the EM algorithm. The likelihood of each chroma frame given chord templates was modeled as a mixture where the estimated overall chord probabilities are the mixing proportion. The likelihood function is treated as the similarity measure between the chroma vector and chord templates. They achieved 71% overlap accuracy for 3 types of chords (maj, min, and 7).

The majority of recent supervised approach involves Hidden Markov Models (HMM) which requires labeled training data and is capable of incorporating other facet of musical elements such as beats or bass line information. In [11], constant tempo, 4/4 time beat pattern, and one global key were assumed; bass pitches from melody lines were incorporated into a probabilistic-based key/chord recognition system. Chroma features were modeled using Gaussian mixture model (GMM) whose parameters were estimated using the EM algorithm and number of Gaussian components were preset to 1, 2 4, 8, and 16. For 150 Beatles' songs, they achieved 73.7% recognition rate for two classes of chords (maj & min). In [12], a six-layer dynamic Bayesian network was used to simultaneously estimate chord sequence, bass notes, metric positions of chords and keys in four layers while the other two observed layers model low-level bass and treble audio features. They achieved 71% accuracy on 176 audio tracks from the MIREX 2008 Chord Detection Task.

3. SYSTEM DESCRIPTION

Figure 1 depicts the high-level components and flow of our system. Our main contributions to the system are in stages 1, 3, and 4. Though the concept of undecimated wavelet transform (UWT) and infinite Gaussian mixture models (IGMM) are not new, to the best of the authors' knowledge, this is the first time that they are applied to denoise audio signals in the context of key/chord estimation as well as using well-known tonal profiles to improve chord recognition directly from a chromagram.

1	Apply UWT on WAV audio
2	Extract chroma features from wavelet approximation
3	Extract a BOL keys from chromagram using IGMM
4	Adjust chromagram based on Stage 3 using KK tonal profiles

Figure 1: System components and flow

3.1. Audio Wavelet Transformation

The ability to effectively reduce transient and percussion noise is an important step for key and chord recognition. Audio denoising is typically applied at the pitch representation stage using median filtering [12] or at the chromagram stage using lowpass filters [13]. In this paper, we adopt UWT on the raw audio signals to reduce noise at the very beginning stage to obtain a smoother representation of raw signals before other audio processing tasks. Unlike a discrete wavelet transform (DWT), the UWT is shift-invariant which is a critical property for denoising since the extraction and conversion of signals from audio CDs to WAV format can easily cause slight misalignment (see details in Section 4.2) in signal locations. Furthermore, the output at each level of UWT has the same sample length as that of the input which allows us to use existing tools for chroma extraction without further translation of the denoised signal.

Music in WAV is read at a sampling rate of 22050Hz. To perform UWT, we first choose an appropriate base wavelet which matches the shape of the target audio signals. This is usually done by visual comparison and therefore subjective in nature. In our case, Daubechies (db) and Symlets (sym) are chosen as candidate base wavelets for UWT. Secondly, we need to determine the order of the base wavelet and level of wavelet decomposition. A higher order base wavelet is generally smoother than lower order ones while wavelet decomposition at a higher level also gives smoother representation of the raw audio signals. We employee different orders, from 8 to 12, for each candidate base wavelet (db8 ~ db12 and sym8 ~ sym12) and different levels (N=3~4) to obtain UWT approximation coefficients; detail coefficients are discarded. Therefore, there are a total of 2 (base wavelet) × 5 (order) × 2 (level) wavelet configurations for a target music piece. Two criteria are experimented in selecting the best configurations to represent denoised audio signals. The first criterion is entropy based where we choose the configuration that produces the wavelet approximation with the lowest Shannon entropy as described in Eq. (1). For the second criterion, a correlation coefficient as described in Eq. (2) is used to measure the similarity between the original audio signals and UWT approximation. The chosen denoised (smoothed) approximation of the original signal is used for chroma extraction in Stage 2.

$$E_{entropy}(S) = - \sum_{i=1}^{n} p_i \cdot \log_2 p_i \quad (1)$$

where S is the signal and p is the energy probability distribution of n wavelet approximation coefficients.

$$C(S,A) = \frac{C_{SA}}{\sigma_S \sigma_A} \quad (2)$$

where S is the signal and A is wavelet approximation. C_{SA} denotes their covariance. σ_S and σ_A are the standard deviation of S and A, respectively.

Since the length of the raw audio signals must be a multiple of 2^N for UWT, we satisfy this requirement by removing the last 2^N sampled raw data points, i.e., we remove at most (2^N -1) samples for the N-level UWT from the raw signals. Removal of up to 7 or 15 trailing samples has virtually no impact on chroma representation since the wavelet transformation maintains the original sampling rate of 22050 Hz. Therefore, the removed trailing samples represent a duration of at most 7×10^{-4} second. In other words, the dimensions of the denoised signals will remain the same for each song regardless of the values of N (=3~4) under UWT. Figure 2 shows an example of UWT of 500 sampled audio signals and an example of its UWT approximation.

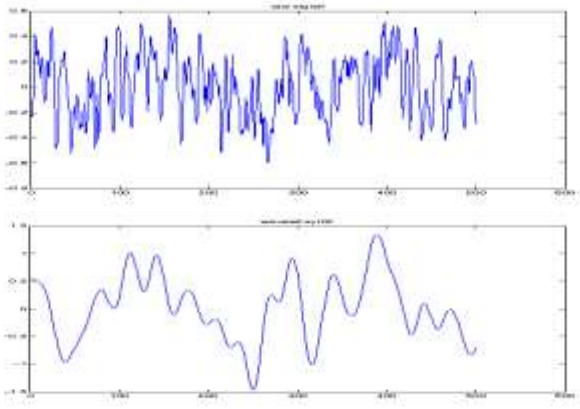

Figure 2: Top: audio signal with 500 samples. Bottom: UWT (sym8, level-4 approximation) transformed signal

3.2. Chroma Extraction

Since the wavelet transform is undecimated, the UWT approximation coefficients represent the signal with the same sampling rate as the original WAV signal. The wavelet-transformed signals are used for chroma feature extraction. We feed these wavelet coefficients as denoised signals into the Chroma Toolbox [14] where a constant Q multirate filter bank is used with sampling rate of 22050 Hz for high pitches, 4410 Hz for medium pitches, and 882 Hz for low pitches. The hop size is half of the sampling rate which results in a feature rate of 10 Hz. Table 1 describes the variants of chroma features (first 3 rows) proposed in [14] and our wavelet-based chroma feature (CUWT-N where N=3 ~ 4). In the following discussion, we use these specific names to address different variants of chroma features for performance comparison. However, for a general discussion of chroma features without the need to address a specific variant, we use CF_i to denote the chroma feature of the ith frame.

Table 1: Variants of Chroma features used in this paper

Name	Feature Description
CLP	Chroma Log Pitch
CENS	Chroma Energy Normalized Statistics (no log)
CRP	Chroma DCT-Reduced log Pitch
CUWT-N	UWT on raw signals to produce CLP

3.3. Local Key Estimation

To achieve higher performance of chord recognition, we first extract local keys of a music piece for two reasons. First, since a key typically covers wider segments of the music piece than a chord, we assume that extracting local keys from a chromagram is less impacted by noises (such as percussion) due to their wider coverage than that of chords in a music piece. Second, given local keys of a music piece, we can predict prominent pitches that reside within the key; therefore we have a higher chance of extracting correct chords from a noisy chromagram.

Each frame of the chromagram represents the energy level of 12 pitch classes and we want to use prominent pitches to quickly estimate keys within the whole music piece. Since triads (major and minor) are the most prevalent chords in pop music, we apply a simple peak-picking algorithm on each frame to pick out the major or minor triad with the highest energy to represent the frame for key recognition. We denote y_i as the triad representing frame i and denote $\mathbf{Y} = \{y_1, y_2, \ldots, y_n\}$ for n frames of a music piece. Note that Y is a series of preliminary triads that we use to estimate local keys and therefore not the results of chord recognition.

We use a generative process to determine what local keys (latent variable $\boldsymbol{\theta}$) generated Y without any training data. Our emphasis is on finding the most likely local keys that are present in the target music piece but ignore their sequence and precise modulation points. Each θ_i in $\boldsymbol{\theta}$ is modeled as a Gaussian component, specified by its mean and covariance. To bypass the requirement of specifying the number of local keys in a Gaussian mixture, we use an infinite Gaussian mixture model (IGMM) depicted in Figure 3. For details of IGMM which is a specific variant of Dirichlet mixture model, we refer readers to [15].

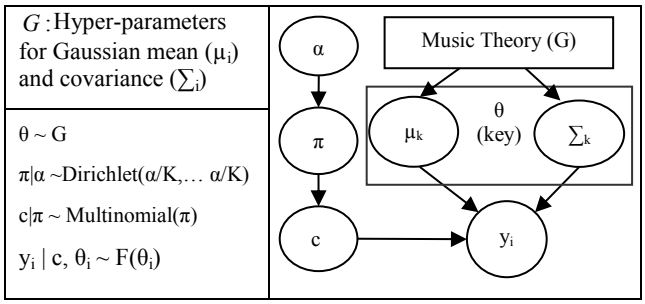

Figure 3: IGMM for keys generation

In Figure 3, θ_i is a Gaussian component with mean (μ_i) and covariance ($\sum_i$). $\mathbf{c} = \{c_1, c_2, \ldots, c_n\}$ is an indicator variable establishing a mapping between each chroma vector in $\mathbf{Y}$ and $\boldsymbol{\theta}$. Hyper-parameter α is the prior for a discrete distribution for mixture proportions π_i where i = 1 …k. A GMM would have a set value of k, but in the case of an IGMM, k is completely determined by the generative process which allows it to go into infinity. The mixing proportions ($\boldsymbol{\pi}$) are modeled as a Dirichlet distribution which serves as a conjugate prior for multinomial component indicators ($\mathbf{c}$).

Given a chromagram Y, the joint posterior distribution of model parameters is described in Eq. (3). Since the indicator variable $\mathbf{c}$ associates each chroma vector to key $\boldsymbol{\theta}$, together they completely determine what local key generated each chroma vector. Therefore, our goal is to use an iterative sampling process to obtain $\mathbf{c}$ and $\boldsymbol{\theta}$.

$$p(\mathbf{c}, \boldsymbol{\theta}, \boldsymbol{\pi}, \alpha|Y) \propto p(Y|\mathbf{c}, \boldsymbol{\theta})p(\boldsymbol{\theta}|G) \prod_{i=1}^{n} p(c_i|\boldsymbol{\pi})p(\boldsymbol{\pi}|\alpha)p(\alpha) \quad (3)$$

Following [18], an efficient way to sample θ and c is based on Eqs. (4 & 5):

$$p(\text{a new } \theta) \propto \alpha / (n - 1 + \alpha) \quad (4)$$

where a new θ sample can be generated based on G as described in Figure 3 and Table 2.

$$p(\text{an existing } \theta_j) \propto n_j / (n - 1 + \alpha) \quad (5)$$

where n and n_i represent the total number of data points and the number of data points generated by θ_j, respectively. θ_j can repeat its value due to discreteness.

Eqs. (4 & 5) govern how to sample a new (or existing) configuration c_i for data point y_i. The idea is that for each y_i in $\mathbf{Y}$ that we process iteratively, we first use Eqs. (4 & 5) to probabilistically determine whether it was generated by a local key that was not seen before or by one of the existing local keys; based on the determination, we generate a new θ as the new unseen local key for y_i or associate y_i to an existing local key. Therefore, if c_i is obtained by Eq. (5), we simply associate y_i with an existing θ_j. If c_i is obtained through Eq. (4), we sample a new θ from G as described in Figure 3. Table 2 describes how we encode mean (μ_i) and covariance ($\sum_i$) of Gaussian key in C major and C minor. Minor keys are encoded based on a mix of harmonic and natural minor scales. Other keys can be obtained by circular shifting the matrices described in Table 2. We implement $\sum_i$ as a diagonal matrix and assign a value of 1 for notes present in the key.

From Figure 3 and Eq. (3), we see that hyperparameter α serves as a prior to the mixture proportions as well as a probabilistic event to introduce a new θ into the mixture of local keys. To sample α from the generative process described in Figure 3, we follow the sampling process proposed by [16] in Eq. (6). The idea is to draw a new value for α at the end of each iteration (after processing all n data points) based on the most recent values of α and k (number of Gaussian components) using Gamma(1,1) as the prior for α.

$$p(\alpha|k, \boldsymbol{\pi}, \mathbf{Y}) = p(\alpha|k) \propto p(\alpha)p(k|\alpha) \quad (6)$$

We feed $\mathbf{Y}$ into the IGMM to iteratively generate local key samples that most likely produced $\mathbf{Y}$. We arbitrarily generate the first key sample and after 4 burn-in iterations, these samples start to converge to the estimated local keys very quickly, usually in less than 12 iterations. Note that a sample generated from an iteration contains all possible local keys used in the entire music piece. We iterate s times to obtain s samples of local keys and

discard those that cover less than 10% of the chromagram due to their short existence. Table 3 summarizes the algorithm.

Table 2: Gaussian coding examples for keys

	Example	[C, C#/Db, …, A#/Bb, B]
μ_i	C major key	[1 0 1 0 1 1 0 1 0 1 0 1]
	C minor key	[1 0 1 1 0 1 0 1 1 1 1 1]
$\sum_i$	C major / minor key	A 12×12 diagonal matrix based on μ_i

Table 3: Key sampling algorithm using IGMM

Obtain peak pitches Y (triad peak-picking)
Initialize G; Initialize c_1 and θ_1 to random values.
For i = 1:s samples
For j=1:n (n = size of Y)
Sample a new c_j based on Eqs. (4 & 5)
If a new θ is required, sample a new θ
Update α based from iteration (i-1) using Eq. (6)
Regroup Y based on all sampled θ;
Discard θ_{-s} that cover less than 10% of the chromagram; output **θ** as a BOL keys

Based on Eqs. (4 & 5) and the sampling process described in Table 3, we see that data points in Y are assumed to be exchangeable which is a prerequisite of a Dirichlet mixture model. In our case, it means that every finite subset of Y, the joint distribution of them is invariant under any permutation of the **c** indicator variable. Obviously, exchangeability does not exist in music since musical notes contained therein are products of careful orchestration by composers and performers. However, for tonal music, its tonal centers (keys) dominate the use of specific pitch hierarchy of the tonic, so the random exchange, in terms of their placement in the music piece, of pitches would have minimal effect in our estimation of a BOL keys; therefore, we can uphold the presumption of exchangeability in applying the IGMM for key analysis.

3.4. Chord Recognition

Given a BOL keys for the chromagram, we recognize 5 chord classes (maj, min, aug, dim, sus) and 1 "N" label representing "no chord" or silent period on a frame-by-frame basis. We catalog the mapping of 17 chord types to the 5 chord classes in Table 4. The idea is that once we have local keys extracted, we only consider pitch energy of diatonic tones within the detected local keys and further adjust chroma energy using the Krumhansl & Kessler (KK) profiles [9] described in Figure 4.

We use binary templates, denoted as TKey, to represent the local keys that we have determined as described in Table 2 (μ_i). Similarly, binary templates are used for chord classes. Therefore, a C major chord has a template TChord_{maj} = [1 0 0 0 1 0 0 1 0 0 0 0]. Given the key information TKey, we use the corresponding KK profile to adjust CF_i (chroma feature for the ith frame, defined in Section 3.2) accordingly by promoting prominent while suppressing less prominent ones in CF_i. We denote $KK_{determined}$ as the key profile for local keys determined from IGMM by circular shifting either $KK_{maj\ or}$ KK_{min}. Every time we circular shift TChord for one of the 5 chord classes, we compute the dot product as described in Eq. (7) to obtain the adjusted chroma energy for frame i. The TChord_c template corresponds to the highest energy sum, $CF_{i\_adjusted}$, of the dot product is the recognized chord for frame i.

$$CF_{i\_adjusted} = CF_i \cdot TKey \cdot KK_{determined} \cdot TChord \quad (7)$$

Table 4: Chord classes

Chord Class	Chord Type
Major	maj, maj7, 7, maj6, 9, maj9
Minor	min, min7, minmaj7, min6, min9
Diminished	dim, dim7, hdim7
Augmented	Aug
Suspended	sus2, sus4

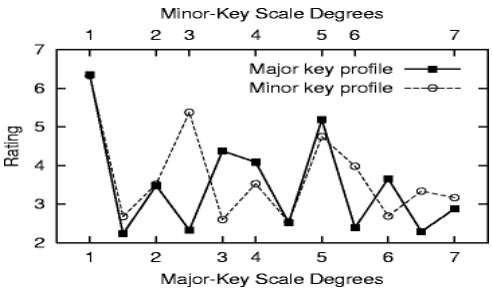

Figure 4: The Krumhansl & Kessler major- and minor-key profiles

For example, if C major has been detected as one of the local keys in a music piece, we set TKey = [1 0 1 0 1 1 0 1 0 1 0 1] and $KK_{determined}$ to values as described in Figure 4. We also circular shift 5 chord templates (maj, min, dim, aug, sus) as TChord to represent C major, C minor, and etc. Finally, for each chroma frame CF_i, we use Eq. (7) to obtain the adjusted chroma feature $CF_{i\_adjusted}$ and we select the TChord among all detected local keys and 5 chord classes that produced the highest energy sum $CF_{i\_adjusted}$ as the recognized chord for the ith frame.

4. EXPERIMENTAL RESULTS

We tested the performance of this new approach using 175 songs[1] from the Beatles' 13 albums. The signals were down-sampled to 22050Hz with mono channel.

The two selection criteria described in Eqs. (1 & 2) represent competing perspectives on how to choose a base wavelet for denoising: one seeks to minimize the entropy while the other maximizes similarity. We randomly chose one song from each album and tested the two criteria and it was evident

[1] We exclude 5 songs out of 180 due to ambiguous tunings. They are: Revolution 9, Love You Too, Wild Honey Pie, Don't Pass Me By, and The Continuing Story of Bungalow Bill.

that a trade-off criterion was in order. Therefore, we selected the best wavelet that maximized $C(S,A)/E_{entropy}(S)$ among the configurations as described in Section 3.1. Furthermore, based on this preliminary test of wavelet configurations, we determined that level-4 (N = 4) approximation was the most suitable for applying UWT on the WAV audio signals to estimate local keys and chords.

4.1. Local key Estimation

For competitions in global key extraction, MIREX[2] provides an evaluation method that gives a full point for correct key estimation and various partial points to related keys such as perfect fifth, relative or parallel major/minor. However, no evaluation method has been proposed for competition in local key extraction. Fixed-length windows have been proposed by [6, 7] to extract keys; with the aid of metrical information, [18] estimated local keys without using such fixed-length window. Since we extract a BOL keys, none of the above evaluation method can be applied for our work. The most suitable evaluation method for our work is to use precision, recall, and F-measure which is a widely adopted metric for information retrieval task. We used musicologist Allan Pollack's complete annotation of all the Beatles' recordings from the internet [19] as the ground truth to calculate the three measures. We extracted all possible local keys (even described as "hint of" modulation) described in his notes for each song to compare with recognized local keys from our algorithm. Also, recall from Section 3.3, we discarded local keys that covered less than 10% of total frames in the chromagram. Moreover, we strictly compared our results with Pollack's notes – i.e., related keys (fifth, relative or parallel major/minor) were not counted as correct recognition and no partial points are given. We categorized songs into single and multiple keys and computed their precision, recall, and F-measure. Table 5 depicts the findings of using the CUWT-4 denoised chromagram. We give a brief definition of the three measures to facilitate the discussion.

- Precision: fraction of retrieved keys are true keys
- Recall: fraction of true keys that are successfully retrieved
- F-measure: (2 × precision × recall) / (precision + recall)

In Table 5, the high recall values for songs without key modulations indicate that 86% of true global keys are extracted and almost 70% of extracted keys are true keys. However, for the 30 songs with key modulations, we extracted 60% of true local keys while 65% of extracted local keys are true keys. Overall, 69% of extracted keys are true keys and 82% of true keys are retrieved. The result is encouraging and indicates that a chromagram using level-4 approximation of UWT in conjunction with an IGMM generative process can be used to recognize global as well as local keys in a music piece. Since the overall precision is 13% lower than that of recall, we conclude that the algorithm generates a high number of false positives.

Table 5. Performance of local key estimation

	# of songs	Precision	Recall	**F-Measure**
Single key	145	.698	.866	**.773**
2 ~ 4 keys	30	.650	.603	**.625**
Overall	175	**.690**	**.820**	**.750**

4.2. Chord Recognition

We use [17] as ground truth which contains a sequence of chords' start and end times for each song. Recognition rate is defined as the number of frames that correctly identify the chord over the total number of frames (average overlap score, AOS) for the whole duration of the 175 songs. Since all chords specified in the ground truth can be mapped to the 5 chord classes (and "N"), all frames are evaluated against the ground truth. However, since the average time difference in terms of song lengths between Harte's annotation and our chroma features is 0.262 second which is longer than 2 frames, we suspect that there are slight misalignments in our WAV files (comparing with the ground truth) after they are ripped from audio albums. Therefore, we also report recognition rate with 1 frame tolerance on each side of the annotated chord. Table 6 depicts the results. To see the effect of using the extracted local key knowledge for chord recognition, Table 6 also shows chord recognition rates without using the extracted keys (in parenthesis) -- all 24 major and minor keys from the KK profiles were used in Eq. (7) in the process described in Section 3.4.

Table 6: Average overlap score for chord recognition

	Exact Match (no tolerance) (%)	+/- 1 Frame Tolerance (%)
CLP	64.5 (41.5)	68.3
CENS	51.9 (33.8)	54.8
CRP	56.2 (42.9)	59.3
CUWT-3	66.7 (43.5)	70.7
CUWT-4	**68.2** (42.9)	**72.3**

From Table 6, we see that extracted keys improved the chord recognition rate by roughly 15% comparing with the case of bypassing stage 3 and using all 24 KK profiles to adjust the chromagram for chord recognition as described in Section 3.4. Though extracted keys are a subset of all 24 keys and higher energy levels are expected in diatonic pitch classes of a chroma vector, the chord recognition process described in Eq. (7) did not work well without using targeted local keys extracted from Stage 3. We also see that the three chroma features (CLP, CENS, and CRP) produced lower recognition rates which are consistent with the analysis described in [14] with the exception that the CLP outperforms CRP significantly. Furthermore, our UWT level-4 denoised chroma feature gives approximately 4% boost over CLP. Furthermore, if we can properly align our audio files with the ground truth, we estimate that our chord recognizer achieves an AOS of 72% on 5 classes of chords and 1 "N" (no chord) label without discarding any frames from the chromagram.

[2] http://www.music-ir.org/mirex/wiki/2012:Audio_Key_Detection

5. CONCLUSION AND FUTURE WORK

We have proposed a wavelet-based audio denoising and bag-of-keys extraction techniques which lead to an unsupervised chord recognition system without using other musical elements. The UWT can be easily applied to any WAV signals to obtain a smoother approximation as input to a chroma-based audio processing front-end for subsequent tasks. We estimated that the best representation of the raw audio signals for a chord recognizer is the level-4 UWT using either Symlet or Daubachies base wavelet with various orders. We obtained the best approximation by choosing the wavelet configuration that produced the maximum correlation to entropy ratio.

Our second contribution is the use of an IGMM to probabilistically determine a BOL keys that generated the chromagram. For tonal music, we can safely predict that the majority of pitches or musical notes that are present in a music piece are from the diatonic scale of a (local) key and those not in the scale are consider accidentals. Therefore, we treat chroma vectors in the chromagram as exchangeable to obtain a BOL keys using IGMM. Unlike estimating a time series of local keys, our bag-of-local-keys approach bypasses the need to specify the length of a sliding window through the chromagram. The search for the optimal window length is still an open problem [18].

From the experimental results, chroma features using UWT-4 gains approximately 4% accuracy for chord recognition. Using simple mean and covariance profiles based on fundamental music theory, the generating process can produce local key samples that converge quickly in less than 12 iterations. With extracted local keys, we adjust the chromagram by applying the Krumhansl & Kessler profiles to promote diatonic pitches to recognize chords. Other profiles, such as [8] can also be used. From the experiments, the overall chord recognition rate is at least 68.2% and possibly 72.3% using slightly misaligned and perfectly aligned audio files, respectively.

Comparing with the more complex chord recognizers using supervised learning technique, a simpler and unsupervised counterpart can perform just as well or outperform approaches requiring scarce labeled training data. Our next step is to adjust the framework to replace the one-way interaction (a BOL keys first, then frame-by-frame chords) with two-way estimation so that chord information can be used to transform the BOL keys into a time series of local keys which can in turn improve the chord recognition task iteratively.

6. REFERENCES

[1] Wang, Y.-S. 2013. Toward segmentation of popular music. *Proc. Int. Conf. Multimedia Retrieval*, pp. 345-348.

[2] De Clercq, T. and Temperley, D. 2011. A corpus analysis of rock harmony. *Popular Music*, vol. 30/1, Cambridge University Press, pp. 47-70.

[3] Wang, Y.-S. and Wechsler H. 2012. Musical keys and chords recognition using unsupervised learning with infinite Gaussian mixture. *Proc. Int. Conf. Multimedia Retrieval.*

[4] Fujishima, T. 1999. Realtime chord recognition of musical sound: a system using Common Lisp Music. *Proc. Int. Computer Music Conf*, pp. 464-467.

[5] Catteau, B., Martens, J., and Leman, M. 2007. A probabilistic framework for audio-based tonal key and chord recognition. *Adv. in Data Analysis—Proc. 30th Annu. Conf. Gesellschaft Für Klassifikation*, R. Decker and H.-J. Lenz, Eds., pp. 637–644.

[6] Rocher, T., Robine, M., Hanna, P., and Oudre, L. 2010. Concurrent estimation of chords and keys from audio. *Proc. Int. Conf. Music Inf. Retrieval*.

[7] Pauwels, J., Martens, J.-P., and Leman, M. 2011. Improving the key extracting performance of a simultaneous local key and chord estimation system. *Proc. of the Int. Conf. on Multimedia and Expo*, pp. 1-6.

[8] Temperley, D. 2001. *The Cognition of Basic Musical Structures*, MIT Press.

[9] Krumhansl, C. 1990. *Cognitive Foundation of Musical Pitch*. Oxford University Press.

[10] Oudre, L., Févotte, C., and Grenier, Y. 2011. Probabilistic template-based chord recognition. *IEEE Trans. Audio, Speech, and Language Processing*, 19(8):2249-2259.

[11] Itoyama, K., Ogata, T., and Okuno H.G. 2012. Automac chord recognition based on probabilistic integration of acoustic features, bass sounds, and chord transition. *Lecture Notes in Computer Science*, *Advanced Research in Applied Artificial Intelligence,* vol. 7345, , pp. 58-67.

[12] Mauch, M. and Dixon, S. 2010. Simultaneous estimation of chords and musical context from audio. *IEEE Trans. Audio, Speech, and Language Processing*, pp. 1280-1289.

[13] Harte, C. and Sandler, M. 2005. Automatic chord identification using a quantized chromagram. *Proc. of the Audio Engineering Society Convention*.

[14] Müller, M. and Ewert, S. 2011. Chroma Toolbox: MATLAB Implementations for Extracting Variants of Chroma-Based Audio Features. *Proc. Int. Conf. Music Inf. Retrieval*.

[15] Rasmussen, C.E. 2000. The infinite Gaussian mixture model. *Advances in Neural Information Processing Systems*, pp. 554-560, MIT Press.

[16] West, M., Müller, P., and Escobar, M.D. 1994. Hierarchical priors and mixture models, with application in regression and density estimation. *Aspects of Uncertainty*, pp. 363-386, John Wiley.

[17] Harte, C., Sandler, M., Abdallah, S., and Gómez, E. 2005. Symbolic representation of musical chords: A proposed syntax for text annotations. *Proc. Int. Conf. Music Inf. Retrieval*.

[18] Papadopoulos, H. and Peeters, G. 2012. Local key estimation from an audio signal relying on harmonic and metrical structures. *IEEE Trans. Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1297-1312.

[19] Pollack, A.W. Notes on … Series http://www.icce.rug.nl/~soundscapes/DATABASES/AWP/awp-alphabet.shtml. Retrieved on October 13, 2011.