MIREX 2010: JOINT RECOGNITION OF KEY AND CHORD FROM MUSIC AUDIO SIGNALS USING KEY-MODULATION HMM

Yushi Ueda, Yuki Uchiyama, Nobutaka Ono, Shigeki Sagayama
Graduate School of Information Science and Technology
The University of Tokyo
{ueda,uchiyama,onono,sagayama}@hil.t.u-tokyo.ac.jp

ABSTRACT

This extended abstract describes a submission to the Music Information Retrieval Evaluation eXchange 2010 (MIREX 2010) in the Audio Chord Estimation and Audio Key Detection tasks. We propose a new model to recognize musical keys and chords simultaneously from musical acoustic signals including key modulations. Chords and keys are closely related notions of music involving harmony. Since occurrences of the chords and transitions between chords largely depend on its keys, it is obvious that using key information supports chord recognition task. But the problem here is that keys and chords are both unknown and the possibility of key modulations within a song makes this problem even harder. Considering the mutual dependency of keys and chords, we believe that recognizing both of them at the same time helps one another than recognizing them separately. To realize this, we propose key-modulation hidden Markov model (HMM) whose hidden states representing each pair of key and chord. The model has transition probabilities to different keys which can handle key modulations. By applying the Viterbi decoding, the key and chord for each frame is estimated. The experimental result shows that this model performs better for chord recognition than the conventional chord HMM and can also recognize keys in high accuracy.

1. INTRODUCTION

As the increase of music databases is accelerating, the need for music information retrieval (MIR) has been growing. Among representations of music, chord progression or harmony is one of the most important elements of Western tonal music which plays a dominant role in determining the music structure and mood. When we listen to a music, even without knowing the individual notes in the music, we can hear the harmony. Thus, chord progression of music can be a useful information in MIR tasks such as genre classification or mood classification. More effective retrieval can be done by recognizing not only chords but also keys because we can obtain the functionality of chords (e.g. *Tonic*, *Dominant*) from the key.

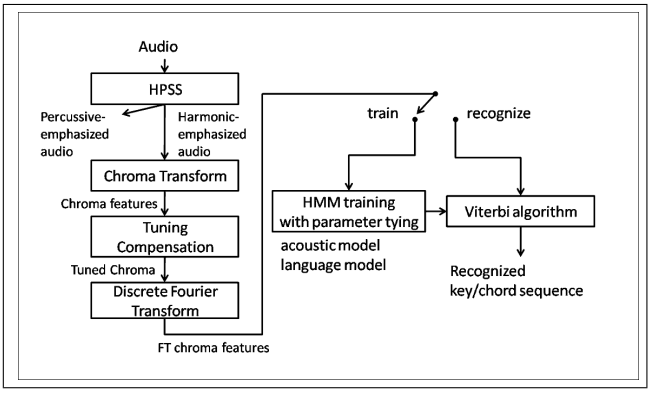


Figure 1. The flow diagram of the system

Recent approaches to identifying chords and keys are based on a hidden Markov model (HMM). The first approach using an HMM-based system was done by Kawakami *et al.* for the purpose of harmonization to the given melody in a symbolic form [1]. Using chroma features [2] and HMM, Sheh and Ellis recognize chords from acoustic signals [3]. They deal with an inadequate amount of training data by assuming that chroma vectors from the same mode (e.g., Major) and different pitch classes can be considered as rotated versions of one another. Saito *et al.* recognize keys in acoustic signals by using key-dependent HMMs and choosing the maximum likelihood key hypothesis within a song [4]. Lee and Slaney increase the amount of training data by synthesizing audio from MIDI to provide accurate chord and boundary information. Improvement is made by using key-dependent HMMs to recognize both the key and chords [5]. Although the key-dependent HMM is effective when the key does not change within a song, the problem of the model is that it does not have flexibility to cope with key modulations which commonly exist in music.

In this paper, we propose a new HMM, the key-modulation HMM, which is an extension of the key-dependent HMM. This model handles key modulations within a song by allowing transitions among chords in different keys and recognizing the key and the chord jointly every frame. Because this extension increases the number of parameters, we apply acoustic and language model parameter tying techniques exploiting characteristics of key and chord. From audio inputs, refined chroma features using harmonic com-

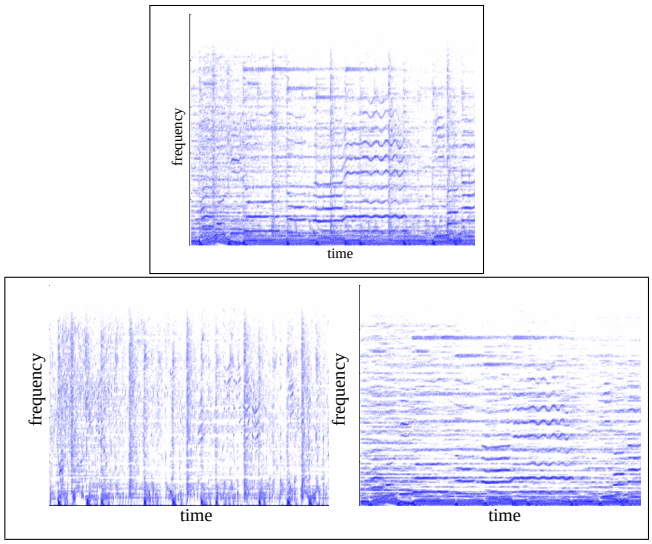


Figure 2. The original spectrogram (upper), the harmonic-emphasized spectrogram (lower right) and the percussive-emphasized spectrogram (lower left) of a popular music piece .

ponent emphasis, tuning compensation and DFT are extracted as in [6].

Figure 1 shows the flow diagram of the system. In the following sections, we describe the algorithms utilized in the system. The feature extraction algorithms are described in Section 2. Then the conventional models are reviewed and the key-modulation model is proposed in Section 3. This section also describe parameter tying techniques and key and chord recognition procedure using the proposed HMM. In Section 4, experimental results are shown. Section 5 gives conclusions of the paper.

2. FEATURE EXTRACTION

2.1 Harmonic Component Emphasis

Generally, harmonic and percussive sounds are mixed in the observed spectrograms of audio pieces. Percussive sounds are not needed for key and chord recognition because they have no particular pitches and overlaps with harmonic sounds. Therefore in order to perform key and chord recognition in higher accuracy, it is useful to separate these components as a preprocessing step and use only the harmonic component. We utilize the harmonic/percussive sound separation (HPSS) technique [7, 8] that is based on the difference of general timbrel features. By looking at the upper row in Figure 2, a typical instance of spectrogram, one can observe that harmonic components tend to be continuous along the temporal axis in particular frequencies. On the other hand, percussive components tend to be continuous along the frequency axis and temporally short. Mask functions for separating the two components (harmonic and percussive) are calculated in the framework of a maximum a priori (MAP) estimation approach using the expectation maximization (EM) algorithm. Applying this approach to the shown spectrogram, harmonic and percussive components are separated and harmonic ones are emphasized (harmonic and percussive components are shown in the lower right and the lower left of Figure 2 respectively).

2.2 Chroma Features

PCPs or chroma vectors [2] are the most commonly used features in key or chord detection. They are 12 dimensional time series vectors corresponding to the energy distributions of 12 pitch classes. There are various methods to calculate a chroma vector, among which we use the constant Q transform [9], which provides spectral analysis using a logarithmic spacing of the frequency domain. The spectrum S of the audio signal $s(t)$ is given by

$$S(k) = \sum_{t=0}^{T(k)-1} w(t,k)s(t)\exp(-j2\pi Qt), \quad (1)$$

$$Q = \frac{f_k}{\delta f_k}, \quad (2)$$

$$f_k = f_{min}2^{\frac{k}{12}}, \quad (3)$$

where $w(k)$ is the windowing function, f_k is the kth frequency bin ,δf_k is the window length for the kth bin and f_{min} is the lowest frequency. The chroma vector $c_n(b)$ is calculated by summing $S(k)$ over octaves as

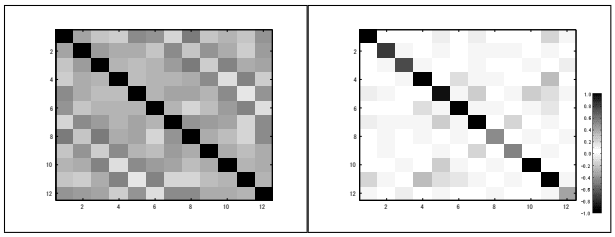
$$c_n(b) = \log\left(\sum_{r=0}^{R}|S(b+r\cdot 12)|\right), \quad (4)$$

where b is the bth pitch class of a chroma vector and R is the number of octaves used to calculate chroma vectors. We take the logarithm here, since the power distribution of a chromagram usually lean toward small values. By taking the logarithm, the distribution approaches a Gaussian, and the approximation of the output probability of HMM to a Gaussian fits well.

2.3 Tuning compensation

In audio signals, tuning pitch may differ from recording to recording and ignoring this difference blurs the chromagram. This is because chromagrams assume that center frequencies of the filterbank match with the performed pitches and if it does not match, energy leaks to the neighboring bins. We can assume that chroma vectors tuned closer to the correct tuning of the recording have larger energy than those tuned farther because energy distributions of the performed pitches fit the filterbank. So, one way to deal with the problem is to choose the chroma vector with largest energy from n tuning frequency candidates which are placed equally every $100/n$ cents, *i.e.*, $f_{min} = f_0$, $f_0 2^{\pm 1/12n}$, $f_0 2^{\pm 2/12n}$, $\cdots$, $f_0 2^{\pm(n-1)/24n}$. We assume that the tuning of the recording does not change over time. Then the maximum-energy chroma vectors $c_{\hat{j}}$ can be obtained by summing c_j over chroma and time bins and choosing the largest energy index $\hat{j}$ as

$$\hat{j} = \underset{j}{\text{argmax}} \sum_{\tau}\sum_{l=0}^{11} c_j(l,\tau) \quad , \quad j = 0, \cdots, n-1, \quad (5)$$



(a) chroma (b) FT of chroma

Figure 3. Covariance matrices – the covariance matrices of chroma vectors are not diagonal but almost circulant, and are therefore diagonalized by the Fourier Transform.

where c_j represents the chroma vectors tuned in the jth frequency candidate.

2.4 DFT Chroma Features

For a robust recognition, it is necessary to reduce the number of parameters to avoid overfitting to the training data. Assuming a Gaussian distribution, it is effective to diagonalize the covariance matrix. In general, the bins of a chroma vector are not independent of each other. The covariance matrix Σ of 180 songs of the Beatles are shown in Fig. 3 (a). Non-diagonal elements are non-zero obviously. Musical sounds usually contain harmonic overtones, for example when a single C note is played energy will also be present in the chroma bins of its overtones G, E and B♭. Also there are co-occurrences of pitches, as notes are often played together in polyphonic music.

Now we consider the assumptions that each note of the input signals has the same harmonic structure and the amount of occurrence of the same intervals (*e.g.* C-G and D-A) is the same. Though there are various harmonic structures and the amount of occurrence of the same intervals differ among recordings, we can consider these assumptions approximately hold as a whole. Therefore the covariance matrix Σ becomes circulant matrix as Fig. 3 (a). A circulant matrix is diagonalized by the DFT matrix F independent of its values [10]. The (i, j) element of F can be written as

$$F_{ij} = \begin{cases} \dfrac{\cos(2\pi ij/12)}{\sqrt{\sum_j \cos(2\pi ij/12)^2}} & \text{if} \quad i = 0, 1, 2, 3, 4, 5, 6 \\ \dfrac{\sin(2\pi(i-6)j/12)}{\sqrt{\sum_j \sin(2\pi(i-6)j/12)^2}} & \text{if} \quad i = 7, 8, 9, 10, 11. \end{cases} \tag{6}$$

3. KEY AND CHORD MODELING AND RECOGNITION

3.1 Proposed Model: Key-Modulation HMM

Figure 4 shows our proposed model. The backward arrow is added to the key-dependent HMM, which makes it possible to transit to the chords belonging to different keys. The contribution of this model is that it recognizes keys and chords frame by frame and has a flexibility to cope with key modulations. Also in the case we do not have to consider key modulations, i.e., no key modulations in the training data, this model is equivalent to the key-dependent HMM. The problem is this model requires large amount of parameters to train because of its degree of freedom. In the following section, we deal with this problem.

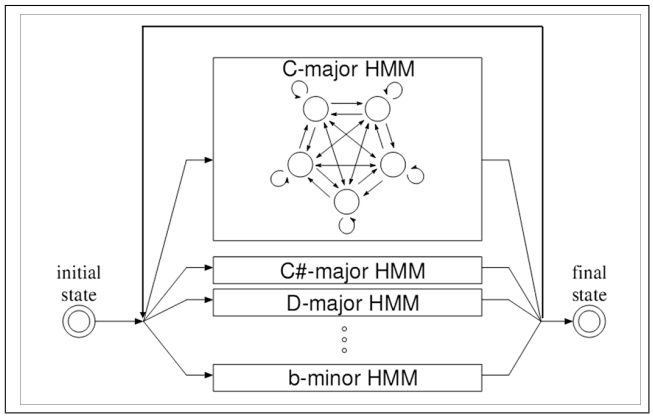


Figure 4. Key-modulation HMM: ergodic transitions between chords in any key

3.2 Model Parameter Tying

The number of states in the proposed models is (# of keys)$\times$ (# of chords), which is quite a number to train parameters of each model independently from the data. There may also be a bias in training data depending on key or chord occurrence, so we cannot assume that all keys and chords appear equally. This can cause problems, for example, when a particular key never appears in training data, but exists in test data. To combat this problem, we apply a model parameter tying technique considering characteristics of keys and chords.

3.2.1 Acoustic Model Tying

In order to avoid a data sparseness problem, the acoustic model of each HMM is tied together considering only the chord, e.g., the model of chord C in key C is identical to chord C in key G, and so on. Assuming a Gaussian distribution to each acoustic model, a probability of an observation vector x emitted from chord c can be written as

$$P(x|c) = \frac{1}{\sqrt{(2\pi)^{12}|\Sigma_c|}} \exp\{-\frac{1}{2}(x-\mu_c)^T\Sigma^{-1}(x-\mu_c)\}, \tag{7}$$

where μ_c and Σ_c are a 1×12 mean vector and a 12×12 covariance matrix of the chord c respectively.

Moreover, to train parameters from a data efficiently, a circular shift can be applied as in [3] so that all the chords in the same mode (e.g. Major or minor) share parameters. For each $N = 0, \cdots, 11$ which represents a chord such as N={A, A#, $\cdots$, G#} or N={Amin, A#min, $\cdots$, G#min},

the parameters of chord N can be written as

$$\mu_N = S^N \mu_0, \quad (8)$$
$$\Sigma_N = S^N \Sigma_0 (S^N)^T, \quad (9)$$

where S is a 12×12 circular shift matrix defined as

$$S = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \end{pmatrix}. \quad (10)$$

3.2.2 Language Model Tying

To compensate a key bias depending on the data, it is natural to consider that chord transition probabilities are independent of key shifts, e.g., the transition probability of C Major to G Major in C Major key is the same as the probability of D Major to A Major in D Major key, etc. Assuming bigram language model, transition probabilities for each key shift $M = \{0, 1, \cdots, 11\}$ can be tied as

$$p(K_2, N_2|K_1, N_1) = p(K_2 + M, N_2 + M|K_1 + M, N_1 + M), \quad (11)$$

where K_1 and K_2 respectively represent each key {A, A#, $\cdots$, G#} or {Amin, A#min, $\cdots$, G#min} and N_1 and N_2 represent each chord {A, A#, $\cdots$, G#} or {Amin, A#min, $\cdots$, G#min}.

3.3 Key and Chord Recognition Procedure

In the framework of Maximum A Posteriori (MAP) estimation, recognizing key sequence $K=\{k_0, k_1, \cdots, k_T\}$ and chord sequence $C=\{c_0, c_1, \cdots, c_T\}$ from observation sequence $X=\{x_0, x_1, \cdots, x_T\}$ can be formulated as

$$\{\hat{K}, \hat{C}\} = \underset{K,C}{\operatorname{argmax}}\, p(K, C|X) = \underset{K,C}{\operatorname{argmax}}\, p(X|K, C)p(K, C), \quad (12)$$

using the Bayes' theorem. Assuming bigram language models and key independent acoustic models described in 3.2, equation (12) can be rewritten as

$$\begin{aligned} \{\hat{K}, \hat{C}\} &\simeq \underset{K,C}{\operatorname{argmax}}\, p(x_0|k_0, c_0)p(k_0, c_0) \times \prod_{t=1}^{T} p(x_t|k_t, c_t)p(k_t, c_t|k_{t-1}, c_{t-1}) \\ &\simeq \underset{K,C}{\operatorname{argmax}}\, p(x_0|c_0)p(k_0, c_0) \times \prod_{t=1}^{T} p(x_t|c_t)p(k_t, c_t|k_{t-1}, c_{t-1}), \end{aligned} \quad (13)$$

and calculated efficiently by the Viterbi algorithm.

4. REFERENCES

[1] T. Kawakami, M. Nakai, H. Shimodaira and S. Sagayama, "Harmonization for melody using HMM," in *Proc. JHES*, F-61, p. 361, 1999. (in Japanese)

[2] T. Fujishima, "Real-time chord recognition of musical sound: A system using common lisp music," in *Proc. ICMC*, pp. 464–467, 1999.

[3] A. Sheh and D. P. W. Ellis, "Chord segmentation and recognition using EM-trained hidden Markov models," in *Proc. ISMIR*, pp. 183–189, 2003.

[4] S. Saito, H. Takeda, T. Nishimoto and S. Sagayama, "Key detection of music audio signals via HMM using chroma vector through specmurt analysis," in *Technical Report of IPSJ*, 2005-MUS-61, pp. 85–90, 2005. (in Japanese)

[5] K. Lee and M. Slaney, "Acoustic chord transcription and key extraction from audio using key-dependent HMMs trained on synthesized audio," *IEEE Trans. on Audio Speech and Language Processing*, vol. 16, no. 2, pp. 291–301, 2008.

[6] J. Reed, Y. Ueda, S. M. Siniscalchi, Y. Uchiyama, S. Sagayama and C.-H. Lee, "Minimum Classification Error Training to Improve Isolated Chord Recognition," in *Proc. ISMIR*, pp. 609–614, 2009.

[7] N. Ono, K. Miyamoto, J. L. Roux, H. Kameoka and S. Sagayama, "Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram," in *Proc. EUSIPCO*, 2008.

[8] Nobutaka Ono, Kenichi Miyamoto, Hirokazu Kameoka, Jonathan Le Roux, Yuuki Uchiyama, Emiru Tsunoo, Takuya Nishimoto, Shigeki Sagayama, "Harmonic and Percussive Sound Separation and its Application to MIR-related Tasks," Advances in Music Information Retrieval, ser. Studies in Computational Intelligence, Z. W. Ras and A. Wieczorkowska, Eds. Springer, 274, pp. 213–236, Feb., 2010.

[9] J. Brown, "Calculation of a Constant Q Spectral Transform," *Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, 1991.

[10] G. Golub and C. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1996.