AUTOMATIC CHORD TRANSCRIPTION WITH CONCURRENT RECOGNITION OF CHORD SYMBOLS AND BOUNDARIES

Takuya Yoshioka, Tetsuro Kitahara, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno
Graduate School of Informatics, Kyoto University
Yoshida-hommachi, Sakyo-ku, Kyoto 606-8501, Japan

ABSTRACT

This paper describes a method that recognizes musical chords from real-world audio signals in compact-disc recordings. The automatic recognition of musical chords is necessary for music information retrieval (MIR) systems, since the chord sequences of musical pieces capture the characteristics of their accompaniments. None of the previous methods can accurately recognize musical chords from complex audio signals that contain vocal and drum sounds. The main problem is that the chord-boundary-detection and chord-symbol-identification processes are inseparable because of their mutual dependency. In order to solve this mutual dependency problem, our method generates hypotheses about tuples of chord symbols and chord boundaries, and outputs the most plausible one as the recognition result. The certainty of a hypothesis is evaluated based on three cues: acoustic features, chord progression patterns, and bass sounds. Experimental results show that our method successfully recognized chords in seven popular music songs; the average accuracy of the results was around 77%.
Keywords: audio signal, musical key, musical chord, hypothesis search

1. INTRODUCTION

The recent rapid spread of online music distribution services demands efficient music information retrieval (MIR) technologies. Annotating musical contents in a universal format is one of the most effective ways to fulfill this demand. Although the new ISO standard MPEG-7 [8] provides a framework for designing such formats, it does not define the methods to obtain musical elements from audio signals. Manual annotation requires a tremendous amount of human work, which makes it difficult to maintain a consistent annotation quality among human annotators. Automatic transcription technologies for musical elements are hence needed to avoid these problems. However, they have not been realized yet.

We focus on musical chord sequences as one of the descriptors of musical elements. A chord sequence is a series of chord symbols with boundaries that are defined as the times when chords change. Descriptors of musical chords will play an important role in realizing effective MIR, since the chord sequences of musical pieces are simple but powerful descriptions that capture the characteristics of their accompaniments. They are also the main factors of determining moods of the pieces, especially in popular music. Therefore, we address the issue of automatic chord transcription.

The main problem in automatic chord transcription is the mutual dependency of chord-boundary detection and chord-symbol identification. It is difficult to detect the chord boundaries correctly prior to chord-symbol identification. If the chord boundaries could be determined before chord-symbol identification, automatic chord transcription could be achieved by identifying the chord symbols in each chord span, which is defined as the time period between the adjacent boundaries. Although chord-boundary-detection methods based on the magnitude of local spectral changes are reported [2, 4], they are not acceptable solutions, because they often mistakenly detect the onset times of non-chord tones or drum sounds when these sounds cause prominent spectral changes.

None of the previous methods [1, 2, 7, 9, 11, 12] has addressed this mutual dependency problem. Aono *et al.* [1] and Nawab *et al.* [9] treated not audio signals from actual musical pieces but chord sounds from a single musical instrument. Kashino *et al.* [7] and Su *et al.* [12] assumed that the chord boundaries were given beforehand. Fujishima [2] developed a method of detecting the chord boundaries based on the magnitude of the spectral changes. However, he treated only musical audio signals that do not contain vocal and drum sounds. Sheh *et al.* [11] developed a method that identifies chord symbols in each 100-ms span without detecting chord boundaries. However, this method cannot correctly identify chord symbols, because the acoustic features in such short spans are liable to be affected by arpeggio sounds and non-chord tones.

To solve this mutual dependency problem, we propose a method that recognizes chord boundaries and chord symbols concurrently. Our method generates hypotheses about tuples of chord boundaries and chord symbols, and evaluates their certainties. It finally selects the most plau-

sible one as the recognition result. As cues for evaluating the certainties of hypotheses, our method uses chord progression patterns (*i.e.* concatenations of chord symbols that are frequently used in actual musical pieces) and bass sounds as well as acoustic features. To use the chord progression patterns appropriately, musical keys are needed. Our method hence also identifies the keys from input audio signals.

The rest of this paper is organized as follows: Section 2 describes the problems in realizing automatic chord transcription and our approach to solve them. Section 3 explains our method in detail. Section 4 reports the experimental results that show the effectiveness of our method. Section 5 concludes this paper.

2. AUTOMATIC CHORD TRANSCRIPTION

2.1. Specification of Automatic Chord Transcription

In this paper, we define automatic chord transcription as the process of obtaining chord sequence $c_1 c_2 \cdots c_n$ and key k from musical audio signals. We treat musical pieces that satisfy the following assumptions:

(A1) The key does not modulate.

(A2) The key is a major key.

Chord c_i is defined as follows:

$$c_i = (cs, b, e), \tag{1}$$

where cs denotes the chord symbol, and b and e denotes the beginning and end times of chord c_i respectively. We call duration $[b, e]$ as the chord span of c_i. Chord symbol cs is defined as follows:

$$cs = (root, style) \tag{2}$$

$$root \in \{\mathrm{C}, \mathrm{C\#}, \cdots, \mathrm{B}\} \tag{3}$$

$$style \in \{\text{major, minor, augmented, diminished}\}, \tag{4}$$

where $root$ denotes the root tone and $style$ denotes the chord style. This definition of chord styles, for example, categorizes both the major triad and major 7th chords as major. We think chord styles in such level of detail will be useful in many MIR methods because they capture the moods of musical pieces adequately. Key k is defined as the tuple of its tonic tone ($tonic$) and mode ($mode$):

$$k = (tonic, mode) \tag{5}$$

$$tonic \in \{\mathrm{C}, \mathrm{C\#}, \cdots, \mathrm{B}\} \tag{6}$$

$$mode = \text{major} \tag{7}$$

2.2. Problems: Mutual Dependency in Automatic Chord Transcription

The main difficulty in automatic chord transcription lies in the following mutual dependency of three processes that constitute automatic chord transcription: chord-boundary detection, chord-symbol identification, and key identification. Because of the mutual dependency, these processes are inseparable.

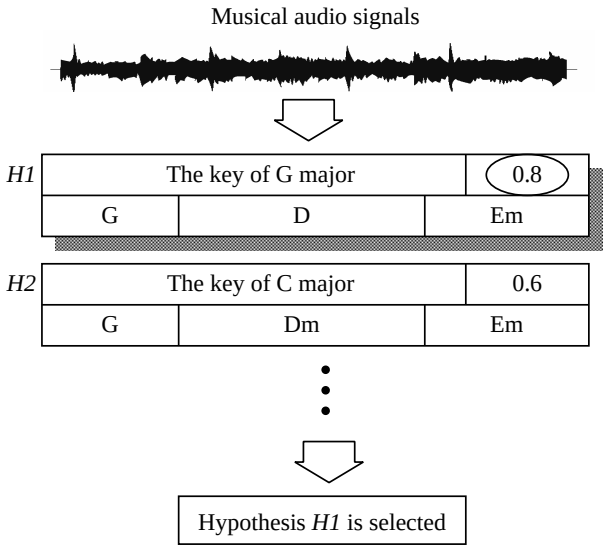

Figure 1. Concurrent recognition of chord boundaries, chord symbols, and keys

1. *The mutual dependency of chord-symbol identification and chord-boundary detection*
   Chord-symbol identification requires a target span for the identification in advance. However, it is difficult to determine the chord spans correctly prior to chord-symbol identification. In order to realize highly accurate chord-boundary detection, the certainties of chord boundaries should be evaluated, based on the results of chord-symbol identification. Chord-symbol identification is therefore indispensable for chord-boundary detection.

2. *The mutual dependency of chord-symbol identification and key identification*
   Chord progression patterns are important cues for identifying chord symbols. Applying the chord progression patterns requires musical keys, because which patterns to apply is dependent on keys. On the other hand, key identification usually requires chord symbols.

2.3. Our Solution: Concurrent Recognition of Chord Boundaries, Chord Symbols, and Keys

In order to cope with the mutual dependency, we developed a method that concurrently recognizes chord boundaries, chord symbols, and keys. Our method generates hypotheses about tuples of a chord sequence and a key with their evaluation values that represent the certainties of the hypotheses, and selects the hypothesis with the largest evaluation value as the recognition result (Figure 1).

The following three kinds of musical elements are used as cues for calculating the evaluation values of hypotheses:

1. *Acoustic features*
   For acoustic features, we use 12-dimensional

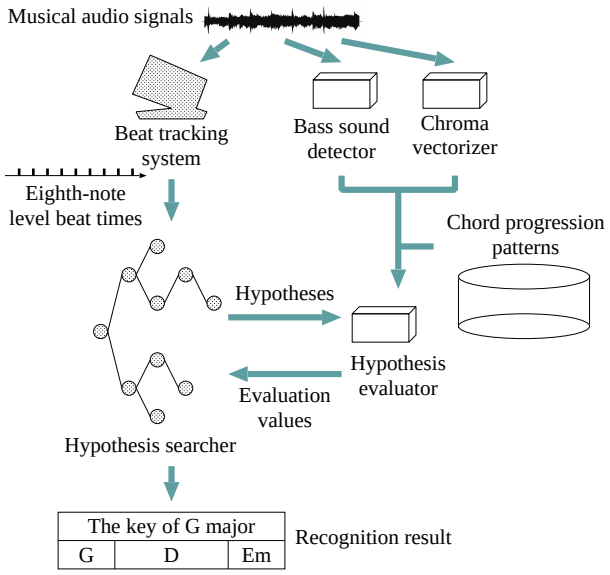

Figure 2. Overview of the automatic chord transcription system

Diatonic chord progression			
G → C	Dm → G	G → Am	C → F
Non-diatonic chord progression			
Am → D → G		G → A♭dim → Am	

Table 1. Examples of the chord progression patterns in the key of C major

chroma vectors [3], which roughly represent the intensities of the 12 semitone pitch classes. Each element of a chroma vector corresponds to one of the 12 pitch classes, and it is the sum of power at frequencies of its pitches over six octaves. The acoustic features are essential cues because chord symbols are defined as collections of the 12 semitone pitch classes.

2. *Chord progression patterns*
Chord progression patterns are concatenations of chord symbols that are frequently used in musical pieces (Table 1) . Using chord progression patterns facilitates reducing the ambiguities of chord-symbol-identification results, which are caused by the absence of chord tones and the presence of non-chord tones.

3. *Bass sounds*
Bass sounds are the most predominant tones in a low frequency region. Using bass sounds improves the performance of automatic chord transcription, because bass sounds are closely related to musical chords, especially in popular music.

Initialization:
for each $s \in S$ **do**
  calculate $f(s)$
  $T \leftarrow T \cup \{s\}$
end
the front time $\leftarrow 0$

Hypothesis search:
while the next time exists **do**
  the front time $\leftarrow$ the next time
  for each $h \in T$ **do**
    Expansion block:
    for each $h' \in V(h,$ the front time$)$ **do**
      calculate $f(h')$
      $T' \leftarrow T' \cup \{h'\}$
    end
    if h is not completely expanded **do**
      $U' \leftarrow U' \cup \{h\}$
    end
  end
  for each $h \in U$ **do**
    do *Expansion block*
  end
  $T \leftarrow$ the best BS hypotheses in T'
  $U \leftarrow U'$
end

return $\arg\max_{h \in T} f(h)$

Figure 3. Hypothesis-search algorithm. S is a set of initial hypotheses. T is a set of hypotheses whose chord sequences reach the front time. U is a set of hypotheses whose chord sequences do not reach the front time. $V(h,t)$ is a set of child hypotheses of hypotheses h at time t. $f(h)$ is an evaluation function that gives the evaluation value of hypothesis h.

3. HYPOTHESIS-SEARCH-BASED AUTOMATIC CHORD TRANSCRIPTION

Our method is based on hypothesis search, which obtains the most plausible hypothesis of all the possible hypotheses that satisfy a given goal statement. In automatic chord transcription, the goal statement is that the chord sequence of a hypothesis ranges from the beginning to the end of an input.

Figure 2 shows an overview of our automatic chord transcription system. First, the beat tracking system detects the eighth-note level beat times of an input musical piece using the method developed by Goto [4]. Then, the hypothesis searcher searches the most plausible hypothesis about a chord sequence and a key. The search progresses every eighth-note level beat time from the beginning of the input. Finally, the searcher outputs the obtained most plausible hypothesis.

The overall process of the hypothesis search is briefly described as follows. At the beginning, initial hypotheses

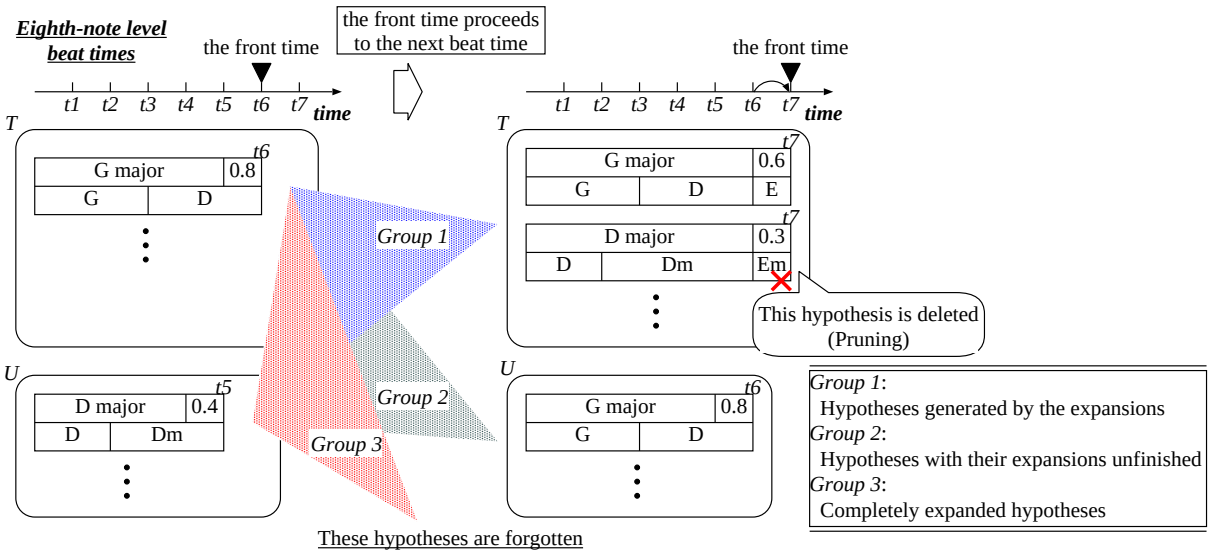

Figure 4. Two sets of hypotheses for reasonable pruning

are given to the hypothesis searcher. Whenever the front time (*i.e.* the time to which the search has progressed) proceeds to the next eighth-note level beat time, the hypothesis searcher expands all hypotheses at that time into ones whose chord sequences range to the front time, and the hypothesis evaluator then calculates the evaluation value of them. When the front time finally reaches the end of the input, the hypothesis that has the largest evaluation value is adopted.

3.1. Hypothesis-search Algorithm

In order to avoid a combinatorial explosion of the number of hypotheses, a search algorithm must contain operations for pruning, which prohibits the expansion of hypotheses with small evaluation values. The pruning must be performed from hypotheses whose chord sequences end at the same time, because pruning from hypotheses whose chord sequences end at different times can incorrectly delete hopeful hypotheses.

Our hypothesis-search algorithm is shown in Figure 3. The key idea of our pruning method is to manage two sets of hypotheses: one is a set of hypotheses with end times that are equal to the front time. The other is a set of hypotheses with end times that are not equal to it. The pruning is performed from the hypotheses in the former set (Figure 4). Therefore, this algorithm reduces the risks of wrong pruning.

The progress of this algorithm is straightforward. It always needs audio signals only around the front time. The time complexity of this algorithm for an n-length input is $O(n)$ when the hypothesis-expansion algorithm takes time $O(1)$. Since our hypothesis-expansion algorithm is of order $O(1)$, our method is able to operate in real time with a large amount of computational power.

Implementing this algorithm requires definition of the following six elements:

1. *Input-scanning times*
   Input scanning times are time points at which hypotheses are expanded. The input-scanning times in our system are defined as the eighth-note-level beat times of an input musical piece.

2. *Data structure of a hypothesis*
   We define hypothesis h of our system as a tuple of chord sequence $c_1 c_2 \cdots c_n$ and key k:
   $$h = (c_1 c_2 \cdots c_n, k). \tag{8}$$

3. *Set of initial hypotheses*
   Our system's set (S) of initial hypotheses is defined as follows:
   $$S = \{(\varepsilon, k_i)\}_{i=0}^{\mathrm{NK}}, \tag{9}$$
   where ε denotes the empty chord sequence, and k_i denotes a key. In our system, $\mathrm{NK} = 11$ based on assumption **A2**; k_0 denotes the key of C major, k_1 denotes the key of D♭ major, $\cdots$, and k_{11} denotes the key of B major.

4. *Hypothesis-expansion algorithm*
   Hypothesis-expansion algorithm, which is denoted by $V(h, t)$ in Figure 3, defines the child hypotheses of hypothesis h at front time t. Its definition in our system is given in section 3.2.

5. *Criterion for determining the end of expansion*
   Our system determines that a hypothesis has completely expanded when the interval between the front time and the end time of the chord sequence of the hypothesis exceeds the measure-level-beat interval of an input musical pieces.

6. *Evaluation function*
   Evaluation function $f(h)$ gives the evaluation value

of hypothesis h. Its definition in our system is given in section 3.3.

3.2. Hypothesis-Expansion Algorithm

Our system's hypothesis-expansion algorithm expands hypothesis $h = (c_1 c_2 \cdots c_n, k)$ into NC hypotheses $h^{(i)} = (c_1 c_2 \cdots c_n c_{n+1}^{(i)}, k)(1 \leq i \leq \text{NC})$, and calculates score $sc_{n+1}^{(i)}$, which indicates the certainty of $c_{n+1}^{(i)}$ based on acoustic features. $c_{n+1}^{(i)}$ is a chord that begins at the end time of chord c_n and ends at front time t. This algorithm ignores the possibility of modulation based on assumption **A1**.

The procedure for determining $c_{n+1}^{(i)}$ and their scores is as follows:

1. Extract a chroma vector from the spectrum excerpt from the span that begins at end time (e) of c_n and ends at front time t.

2. Calculate the Mahalanobis distance between the extracted chroma vector and the mean chroma vector from the training audio signals for each chord.

3. Select NC chord symbols $cs_{n+1}^{(i)}(1 \leq i \leq \text{NC})$, whose distances are smaller than the others. Then, $c_{n+1}^{(i)}$ is represented as $(cs_{n+1}^{(i)}, e, t)$, and $sc_{n+1}^{(i)}$ is defined as the normalized value of the reciprocal of the distance of $c_{n+1}^{(i)}$.

3.3. Evaluation Function

Given hypothesis $h = (c_1 c_2 \cdots c_n, k)$, evaluation function $f(h)$ calculates the evaluation value of h. To calculate the evaluation values of hypotheses, our method evaluates the acoustic-feature-based, chord-progression-pattern-based, and bass-sound-based certainties of the hypotheses. The acoustic-feature-based certainty of a hypothesis indicates the degree of similarity between the chroma vectors from its chord spans and training chroma vectors for each chord. The chord-progression-pattern-based certainty indicates the number of chord-symbol concatenations of the hypothesis corresponding to one of the chord progression patterns. The bass-sound-based certainty indicates the degree of predominance of its chord tones in a low frequency region.

Evaluation function $f(h)$ in our system is defined as follows:

$$f(h) = \log ac(h) + \text{WPR} \times \log pr(h) + \text{WBA} \times \log ba(h), \quad (10)$$

where $ac(h)$ denotes the acoustic-feature-based certainty, $pr(h)$ denotes the chord-progression-pattern-based certainty, $ba(h)$ denotes the bass-sound-based certainty, WPR denotes the weight of the chord-progression-pattern-based certainty, and WBA denotes the weight of the bass-sound-based certainty.

3.3.1. Acoustic-feature-based certainty

Acoustic-feature-based certainty $ac(h)$ is defined as follows:

$$ac(h) = \prod_{i=1}^{n} (sc_i \times \text{EP}^{l_i - 1}), \quad (11)$$

where sc_i denotes the score of chord c_i, l_i denotes the number of intervals of the eighth-note level beats contained in the span of c_i, and EP denotes the span-extending penalty. Defining acoustic-feature-based certainty as the product of sc_i would cause many deletion errors, because the numbers (n) of chords are not equal among different hypotheses. Multiplying the span-extending penalty is an effective way to avoid deletion errors.

3.3.2. Chord-progression-pattern-based certainty

Chord-progression-pattern-based certainty $pr(h)$ is defined as follows:

$$pr(h) = \text{PPR}^m \quad (12)$$

$$m = n - num(i; \exists p, q \ \ s.t. \ \ p \leq i \leq q, \ \ c_p \cdots c_q \in P) \quad \text{for} \quad 1 \leq i \leq n, \quad (13)$$

where P denotes the set of chord progression patterns for key k, PPR denotes the penalty for mismatched progressions, and $num(i; cond(i))$ denotes the number of values i that satisfy condition $cond(i)$. To obtain the set of chord progression patterns for each key, we stored 71 concatenations of chord functions, according to the theory of harmonics (*e.g.* V $\rightarrow$ I). Given a key, our method yields the set of chord progression patterns for the key from the prestored chord-function concatenations. For example, applying the key of C major to V $\rightarrow$ I yields chord progression pattern G $\rightarrow$ C.

3.3.3. Bass-sound-based certainty

Let p_i denote the most predominant pitch class in a low frequency region of the span of chord c_i, and $pred_i$ denote the degree of its predominance. Then, bass-sound-based certainty $ba(h)$ is defined as follows:

$$ba(h) = \prod_{i=1}^{n} htp_i \quad (14)$$

$$htp_i = \begin{cases} pred_i & (\text{if } p_i \text{ is a chord tone of } c_i) \\ \text{PBA} & (\text{otherwise}), \end{cases} \quad (15)$$

where PBA denotes the penalty for the absence of the chord tones in the low frequency region. To obtain the degrees of predominance of pitch classes in the low frequency region, our method forms the pitch probabilistic density function after applying the band pass filter for the bass line using Sakuraba's [10] automatic music transcription system implementing Goto's method [5]. Then, the degree of predominance of each pitch class is defined as the sum of the values of the function at its pitches.

Piece number	Short span	Acoust		Our method		
		$corr$	acc	$corr$	acc	Key
No.14	42%	86%	74%	89%	84%	◯
No.17	57%	90%	64%	91%	76%	◯
No.40	38%	89%	76%	85%	80%	◯
No.44	34%	90%	46%	88%	67%	◯
No.45	53%	90%	68%	86%	74%	◯
No.46	57%	95%	69%	93%	80%	◯
No.74	45%	90%	71%	92%	80%	◯

◯: Correctly identified

Table 2. Experimental results

4. EXPERIMENTAL RESULTS

Our system was tested on one-minute excerpts from seven songs of RWC-MDB-P-2001 [6]: No.14, 17, 40, 44, 45, 46, and 74. The current implementation uses the following parameters: BS $= 20$, NC $= 7$, WPR $= 1.0$, WBA $= 5.0$, EP $= 0.25$, PPR $= 0.8$, and PBA $= 0.5$. For the training data of chroma vectors, we used 2592 excerpts of audio signals of each chord played on a MIDI tone generator and audio signals of the six songs except an input one. To evaluate the effectiveness of concurrent recognition of chord boundaries and chord symbols, we implemented a system that identifies chord symbols in every short span corresponding to the eighth-note level beat interval (called a short span method). We also implemented a system that calculates the evaluation values of hypotheses based on only acoustic features (called an acoust-method).

For evaluating the outputs, we used two criteria: correctness $corr$ and accuracy acc, which is defined as follows:

$$corr = 1 - \frac{\text{\# (substitution and deletion errors)}}{\text{\# (output chords)}} \quad (16)$$

$$acc = 1 - \frac{\text{\# (substitution, deletion, and insertion errors)}}{\text{\# (output chords)}} \quad (17)$$

The correct chord sequences are hand-labeled.

The results are listed in Table 2 (for short span method, only accuracies are shown). Our system's average accuracy was 77%. This result shows that our method can correctly recognize chord sequences from complex musical audio signals that contain vocal and drum sounds. The performance of the short span method was poor. This is because the short span method often confused major chords and their minor versions, since there were many spans where the third tones of chords did not appear. The accuracy of the acoust-method was very smaller than that of our method in spite of the high correctness, since the acoust-method made many insertion errors. This is because the acoustic-feature-based certainties in correct chord spans were liable to be smaller than those in shorter spans due to the spectral changes caused by arpeggio sounds. These results show that our concurrent recognition method of chord boundaries and chord symbols achieves high improvement of chord-recognition performance, and that using chord progression patterns and bass sounds also improves the performance.

5. CONCLUSION

We have described a method that recognizes musical chords and keys from audio signals. To cope with the mutual dependency of chord-boundary detection, chord-symbol identification, and key identification, our method runs these processes concurrently, which is achieved by searching the most plausible hypothesis about a tuple of a chord progression and a key. This method operates without any prior information about the input songs. The experimental results show that our method is robust enough to achieve 77% accuracy of chord recognition on seven popular music songs that contain vocal and drum sounds.

Acknowledgments: This research was partially supported by the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Grant-in-Aid for Scientific Research (A), No.15200015, the Sound Technology Promotion Foundation, and Informatics Research Center for Development of Knowledge Society Infrastructure (COE program of MEXT, Japan). We thank Mr. Yohei Sakuraba for his permission to use his program.

6. REFERENCES

[1] Aono, Y., Katayose, H., and Inokuchi, S. "A Real-time Session Composer with Acoustic Polyphonic Instruments", *Proc. ICMC*, pp.236–239, 1998.

[2] Fujishima, T. "Realtime Chord Recognition of Musical Sound: a System Using Common Lisp Music", *Proc. ICMC*, pp.464–467, 1999.

[3] Goto, M. "A Chorus-Section Detecting Method for Musical Audio Signals", *Proc. ICASSP*, **V**, pp.437–440, 2003.

[4] Goto, M. "An Audio-based Real-time Beat Tracking System for Music With or Without Drum-sounds", *Journal of New Music Research*, Vol.30, No.2, pp.159–171, 2001.

[5] Goto, M. "A Robust Predominant-F0 Estimation Method for Real-time Detection of Melody and Bass Lines in CD Recordings", *Proc. ICASSP*, **II**, pp.757–760, 2000.

[6] Goto, M., Hashiguchi, H., Nishimura, T., and Oka, R. "RWC Music Database: Popular, Classical, and Jazz Music Databases", *Proc. ISMIR*, pp.287–288, 2002.

[7] Kashino, K., Nakadai, K., Kinoshita T., and Tanaka, H. "Application of the Bayesian Probability Network to Music Scene Analysis", Rosenthal, D.H. and Okuno, H.G. *Computational Auditory Scene Analysis*, Lawrence Erlbaum Associates, Pulishers, pp.115–137, 1998.

[8] Manjunath, B.S., Salembier, P., and Sikora, T. *Introduction to MPEG-7*, John Wiley & Sons Ltd., 2002.

[9] Nawab, S.H., Ayyash, S.A., and Wotiz, R. "Identification of Musical Chords using Constant-Q Spectra", *Proc. ICASSP*, **V**, pp.3373–3376, 2001.

[10] Sakuraba, Y., Kitahara, T., and Okuno, H.G. "Comparing Features for Forming Music Streams in Automatic Music Transcription", *Proc. ICASSP*, **IV**, pp.273–276, 2004.

[11] Sheh, A. and Ellis, D.P.W. "Chord Segmentation and Recognition Using EM-Trained Hidden Markov Models", *Proc. ISMIR*, 2003.

[12] Su, B. and Jeng, S. "Multi-timber Chord Classification Using Wavelet Transform and Self-organized Map Neural Networks", *Proc. ICASSP*, **V**, pp.3377–3380, 2001.