

# Music structure analysis using self-similarity matrix and two-stage categorization

Sanghoon Jun · Seungmin Rho · Eenjun Hwang

Published online: 23 November 2013  
© Springer Science+Business Media New York 2013

**Abstract** Music tends to have a distinct structure consisting of repetition and variation of components such as verse and chorus. Understanding such a music structure and its pattern has become increasingly important for music information retrieval (MIR). Thus far, many different methods for music segmentation and structure analysis have been proposed; however, each method has its advantages and disadvantages. By considering the significant variations in timbre, articulation and tempo of music, this is still a challenging task. In this paper, we propose a novel method for music segmentation and its structure analysis. For this, we first extract the timbre feature from the acoustic music signal and construct a self-similarity matrix that shows the similarities among the features within the music clip. Further, we determine the candidate boundaries for music segmentation by tracking the standard deviation in the matrix. Furthermore, we perform two-stage categorization: (i) categorization of the segments in a music clip on the basis of the timbre feature and (ii) categorization of segments in the same category on the basis of the successive chromagram features. In this way, each music clip is represented by a sequence of states where each state represents a certain category defined by two-stage categorization. We show the performance of our proposed method through experiments.

**Keywords** Music structure · Music segmentation · Signal processing · Self-similarity matrix

## 1 Introduction

With the immense popularity of music streaming services and easier accessibility to large repositories of digital music clips, efficient methods for music segmentation and structure analysis have been of demand in many music-related applications such as music understanding, retrieval, and recommendation.

---

S. Jun · E. Hwang (✉)  
School of Electrical Engineering, Korea University, Seoul 136-713, Korea  
e-mail: ehwang04@korea.ac.kr

S. Jun  
e-mail: ysbhjun@korea.ac.kr

S. Rho  
Department of Computer Engineering, Mevlana University, Konya, Turkey  
e-mail: korea.smrho@gmail.com

For instance, many works have been proposed for content-based music retrieval and browsing. In content-based music retrieval, music clips are usually retrieved on the basis of the acoustic similarity in the music signals. Feature vectors representing acoustic characteristics of music are constructed for measuring the similarity in the music clips. Typical digital music clips are recorded for 3–5 min at 44,100 sampling rate, and a large amount of storage space is required for extracting multidimensional feature vectors from each sample or sampling window within a music clip. Moreover, searching through the music clips by measuring the similarities in the feature vectors requires unacceptable computational time.

One simple method to solve these problems is to use the average of the feature vectors for reducing the dimensionality and computation time. However, this can lead to poor performance in terms of precision and recall because the music features that are distinct depending on the sampling point could disappear during the averaging step [8]. Another method is to determine the most distinct segment of a music clip and consider its feature vector as a representation of the music clip. Such a segment can be used as a summary of the music clip because it is the most representative part of the music clip. Determination of such segments can be automatically performed by analyzing the music structure.

Music structure analysis (discovery) segments music clips and structuralizes them using signal analysis. In [15], Peeters classified the approaches for music structure analysis into two categories: in the sequence-based approach, the music audio signal is considered as a repetition of sequences of events. Most studies in this category employ the self-similarity matrix on the music clip, which was proposed in [4]. Sequence is defined as a set of successive times. The notion of sequence is closely related to the notion of melody (sequence of notes) or chord succession in popular music. The sequence can be observed by tracking the diagonal lines in the self-similarity matrix. In the state-based approach, the audio signal is considered as a succession of states. This approach relies mainly on the clustering algorithms. By clustering the checkerboard patterns in the self-similarity matrix, the candidates for the structural boundary are detected. The music clips are structuralized by applying a clustering algorithm on the music segments.

In our previous work [7], we constructed a self-similarity matrix of music clips using a constant-size window, on the basis of which music segmentation and categorization are performed. Even though its overall performance is good, we observe that there is still room for improvement depending on the music genre. Therefore, in this work, we revise our method to improve the accuracy of music segmentation and structure analysis. The main contributions of this paper are to use a beat-scale window for constructing a self-similarity matrix and to perform two-stage segment grouping for more accurate categorization.

The rest of this paper is organized as follows. In Section 2, we present a brief overview on the techniques used for music structure analysis. Sections 3 and 4 present the methods to generate a self-similarity matrix and segment music on the basis of its self-similarity matrix, respectively. In Section 5, a segment categorization method is presented. Section 6 describes the experiments we performed and shows some of the results. In the last section, we conclude the paper and provide some future direction.

## 2 Related work

Most works on music structure analysis and segmentation are involved with acoustic signal analysis. In the acoustic signal analysis, the acoustic features are extracted from the signal to represent its temporal characteristic. By analyzing the pattern of the features within a music clip, we can identify its structural property such as repetition of motives or melodies.

The music structure can be analyzed more effectively by constructing its self-similarity matrix. The self-similarity matrix, which was initially presented by Foote [4], is a very useful tool for observing the time structure of music and audio. This matrix can be obtained in a 2D domain by measuring the cosine similarities between the successive music signals. The checkerboard patterns in the matrix indicate the repetitive similarities, and the music structure can be determined by tracking those patterns. In [5], automatic audio segmentation methods were presented using the self-similarity matrix. For speech and music segmentation in a mixed source, the segment boundaries are extracted by applying the proposed checkerboard kernel and audio novelty score.

In [2], Cooper et al. proposed a scheme for automatically summarizing the music clips. The scheme determines the most representative segment of music clip by summing the self-similarity matrix over the segments. Furthermore, in [3], they presented a framework for summarizing music by structural analysis. For music summarization, a self-similarity matrix of the music clip is calculated, and the segment boundaries are detected by correlating a checkerboard kernel. In addition, those segments are clustered by matrix decomposition on the basis of the spectral statistics of each segment.

Lu et al. [12] proposed an effective approach for analyzing the structure of acoustic music data and discovering repeating patterns. They focused on the chroma-based features rather than on the timbre feature because the note information is used for the structural analysis. For the representation of accurate melody similarity, they used the constant Q transform in feature extraction and proposed a new similarity measure between the musical features.

In [13], Maddage et al. presented a method for music structure analysis using beat space segmentation. This method can detect the music chord and vocal/instrumental boundary as well. For this, the regions with similar melody are determined by matching the subchord patterns. In addition, the regions with similar content are determined on the basis of their vocal content. The regions with similar melody and content are used for identifying the music structures.

Peeters [16] presented a method for automatically estimating the structure of music tracks. The author constructed a higher order similarity matrix on the basis of the timbre and pitch-related features. The music segments are detected from the matrix, and a maximum-likelihood approach is used to simultaneously derive its sequence representation and the most-representative segment of each sequence.

Wang et al. [20] presented a method for recognizing the repeating patterns in acoustic music signals. Their approach is based on the constant Q transform, which is used to extract the musical note information. By measuring the melody/note similarity and applying adaptive the threshold setting method, noticeable repeating patterns can be recognized.

In [14], Paulus et al. proposed a method for describing the music structure by segmenting a music clip, grouping the repetition segments, and assigning musically meaningful labels. Three acoustic features were used for describing the different aspects of the acoustic signal, and a probabilistic fitness function was used to select the feature with the highest matching score to the input pieces. The musicological model consisting of N-grams was employed for labeling the segment groups.

Kaiser et al. [9] presented a method for automatically extracting musical structures of popular music. In order to segment a music clip into regions of acoustically similar frames, non-negative factorization (NMF) was applied to its self-similarity matrix. Based on the observation that structural parts can be easily modeled over the dimensions of the NMF decomposition, they presented a clustering algorithm that can explain the structure of the music clip.

In [18], Serrà et al. proposed an unsupervised method for detecting music boundaries by using time series structure features. Structure features were obtained by considering temporal lag information and estimating a bivariate probability density with Gaussian kernels.

Calculation of differences between consecutive structure features yields a novelty curve whose peaks indicate boundary estimates.

### 3 Self-similarity matrix

#### 3.1 Musical features

For music segmentation and its structure analysis, the musical features should be identified and analyzed. For that purpose, we use two musical features in this work.

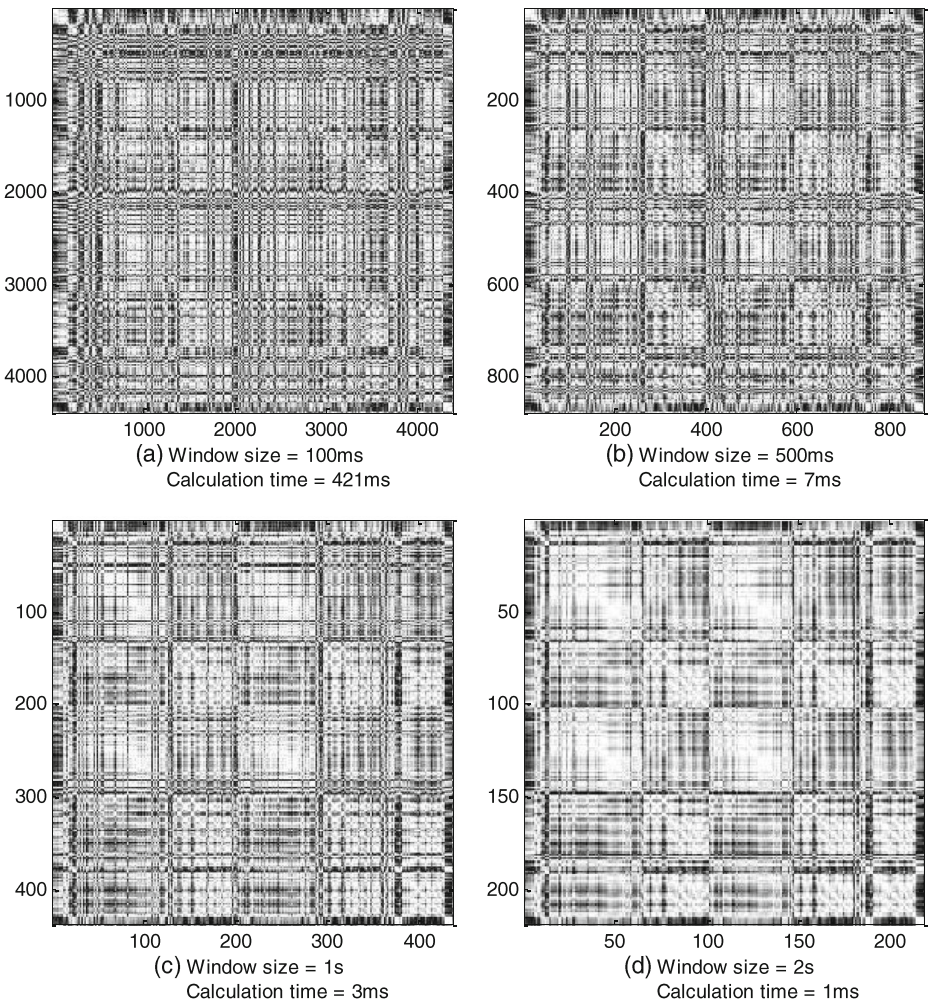
- **Timbre:** Timbre, also known as tone color, is a unique quality of sound that enables people to distinguish the musical instruments in a music clip. In this work, we use the mel-frequency cepstral coefficients (MFCC) to represent the timbre feature of the music clip. MFCC is one of the most important features used in speech recognition and music information retrieval (MIR) [11, 17, 19]. It is generally used with other acoustic features in diverse applications such as genre classification and audio similarity measurement. Particularly, we use the first five MFCC coefficients over the texture window (excluding the coefficient corresponding to the DC component) as was performed in [19].
- **Chromagram:** Chromagram, also known as pitch class profile, is a 12-dimensional vector, which represents the intensities of the 12-semitone pitch classes [6]. The energies of all the semitones over all the octaves are integrated into a single band called pitch class. This chromagram is effective in generating high-level music description such as melody and chord progression; therefore, it can be used in music structural analysis, particularly for detecting the music repetition. In this work, we use this feature to distinguish the segments with similar timbres.

#### 3.2 Preprocessing

When generating a self-similarity matrix, an appropriate window size is very important. For instance, a small window size is effective in capturing the delicate feature transitions as required in the sequence-based structure analysis. On the other hand, state-based structure analysis prefers a large window size because it requires feature transition on a larger scale. In addition, the window size has a very close relationship with the time for generating and calculating a self-similarity matrix. For example, a smaller window size results in a larger self-similarity matrix, as shown in Fig. 1. Music tempo is another crucial factor in selecting the appropriate window size. Generally, songs with faster tempo undergo timbre transition in a shorter time period. Therefore, a smaller size window should be used for songs with faster tempo to capture the feature transition more accurately.

Generally, the window size is defined in the time domain. However, in the case of music, the same window size can cover different music ranges according to its tempo. Therefore, for performing consistent feature transition analysis, we define the window size on the basis of the number of beats using Eq. (1). In this equation, *bpm* represents the number of beats per minute of a music clip, and it is calculated using the tempo extraction algorithm described in [10]. In this paper, the window size is 2 beats.

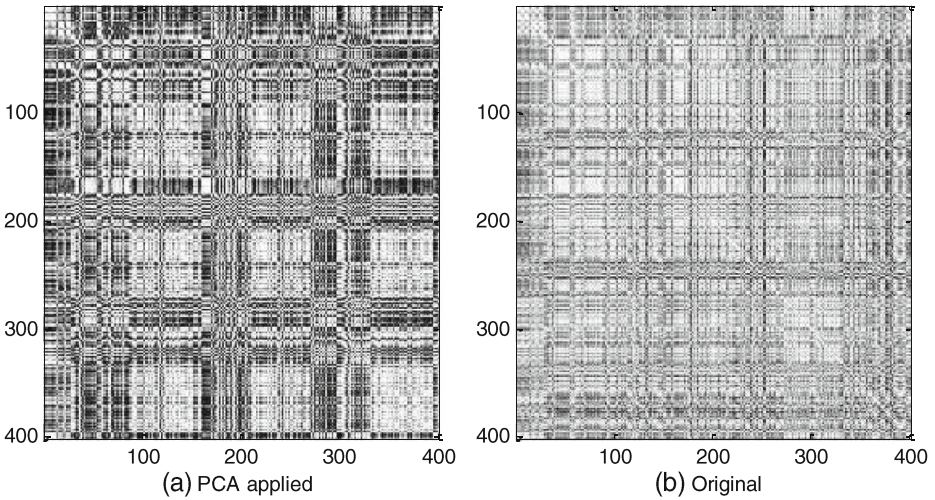
$$\text{window size} = \frac{60}{bpm} \times \# \text{ of beats} \times \text{sampling rate} \quad (1)$$



**Fig. 1** Self-similarity matrices of different window sizes

The self-similarity matrix is generated on the basis of the feature similarities within a song. Particularly, for generating a more distinct self-similarity matrix, we apply the principal component analysis (PCA) to the music feature, which leads to maximized feature difference in songs. More specifically, we calculate the PCA coefficients and the mean vector using the MFCC feature of 210 songs. Those songs were collected from Allmusic.com [1] and cover seven popular genres (blues, electronic, jazz, new age, rap, R&B, and rock). Simultaneously, PCA reduces the dimensionality of the feature vectors from 5 to 3. Figure 2 shows the comparison of the original self-similarity matrix and the PCA-applied self-similarity matrix. As shown in this figure, the PCA-applied self-similarity matrix is clearer than the original self-similarity matrix.





**Fig. 2** PCA effect on the self-similarity matrices

### 3.3 Self-similarity measurement

As mentioned earlier, the self-similarity matrix can be used to determine the boundaries where the timbre features significantly vary within a music clip. In this work, we construct the matrix by calculating the cosine similarity in all the feature vector pairs within the music clip. Each entry in the matrix, *ssm*, is calculated using the following cosine similarity equation

$$ssm(i, j) = similarity(v_i, v_j) = \frac{v_i \cdot v_j}{\|v_i\| \times \|v_j\|} \tag{2}$$

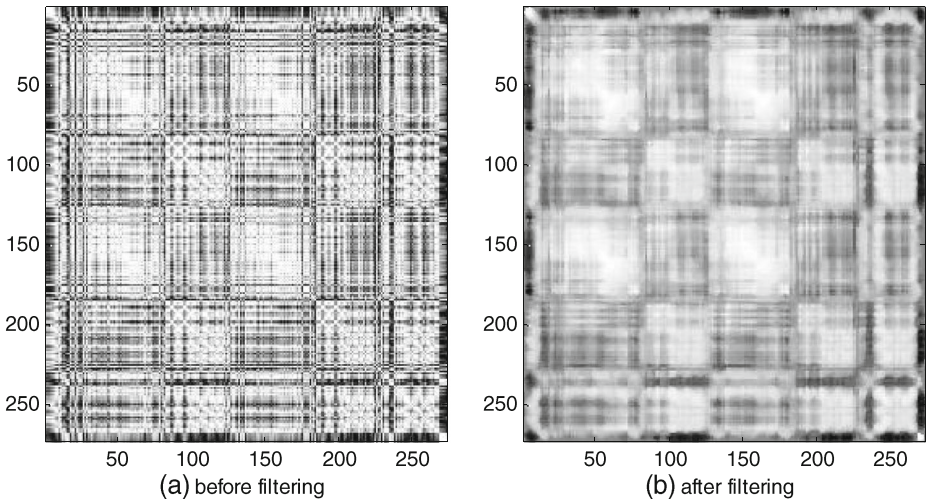
where  $v_i$  and  $v_j$  are the feature vectors within the music clip. In some cases, it is difficult for the matrix to recognize owing to quick and wide transition within the matrix. For instance, the matrix for a song of vocal or acoustic instruments tends to show rough transitions. In order to interenate the matrix, we have applied a two-dimensional median filter to the self-similarity matrix. Figure 3(a) and (b) show the self-similarity matrix before and after filtering, respectively.

## 4 Music segmentation

### 4.1 Sum of row-wise standard deviations

For capturing the significant changes in successive checkerboard pattern in the self-similarity matrix, we use a sliding window whose size is up to 20 samples and compute the summation of its standard deviation. Figure 4 shows an example of the sliding window and its standard deviation. The standard deviation for a vector  $x$  is defined as follows:

$$SD(x) = \left( \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \right)^{\frac{1}{2}} \tag{3}$$



**Fig. 3** Filtering effect on the self-similarity matrices

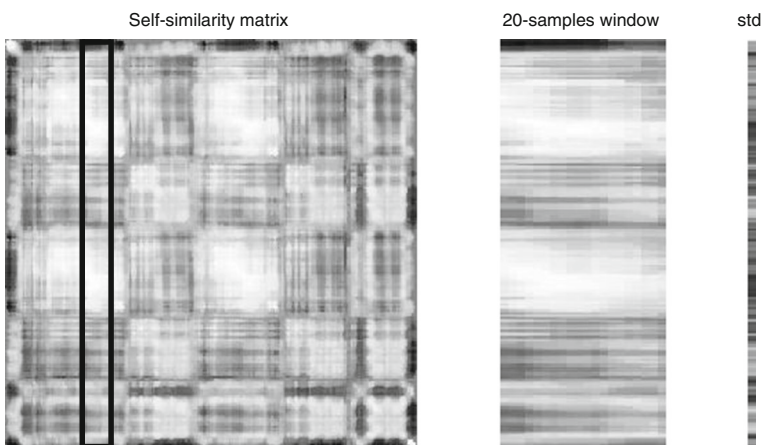
In addition, the sum of the row-wise standard deviations (SRSD) in the  $m$ -th window is calculated as follows:

$$SRSD(m) = \sum_{i=1}^N SD(rv(m, i)) \tag{4}$$

Here,  $N$  is the length of the matrix, and  $rv(m, i)$  is defined as

$$rv(m, i) = \{ssm(i, m), ssm(i, m + 1), \dots, ssm(i, m + L - 1)\} \tag{5}$$

where  $L$  denotes the window length, and we set it to 20 to cover sufficient transitions between the segments.



**Fig. 4** Sliding window and its standard deviation

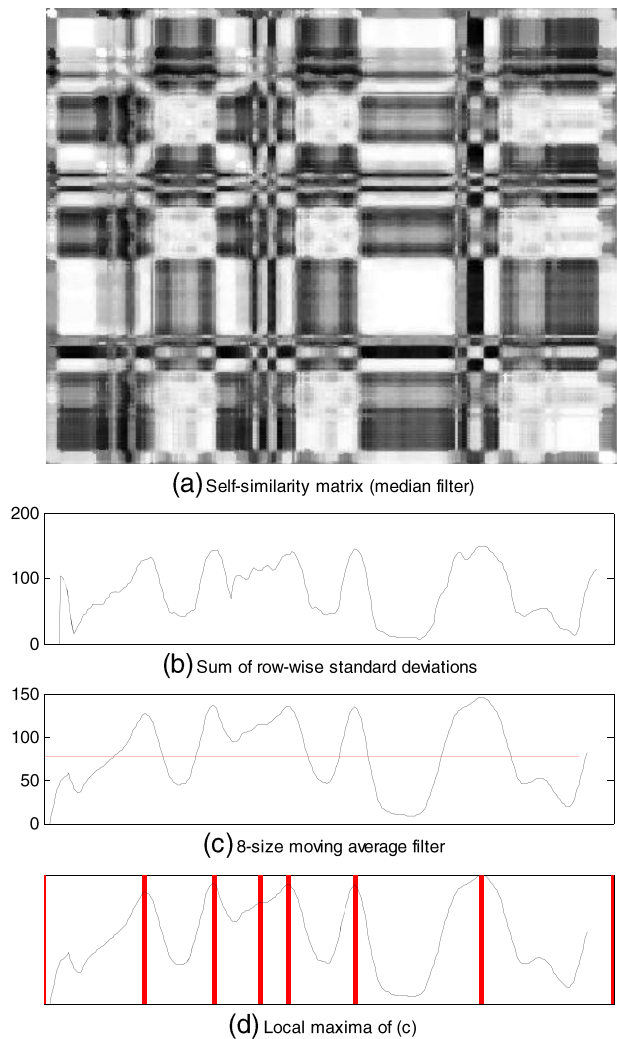
## 4.2 Segmentation

For music segmentation, we need to detect the boundaries of different music features. To detect such boundaries, we first locate the local peak in the SRSD. Since we are interested in conspicuous checkerboard pattern in the self-similarity matrix, we consider local peaks whose size is above some threshold. The threshold can be defined as the average of SRSD within a music clip. On the other hand, small and short peaks are removed by using an eight-size moving average filter. The detailed description of this is illustrated in Fig. 5.

## 5 Segment categorization

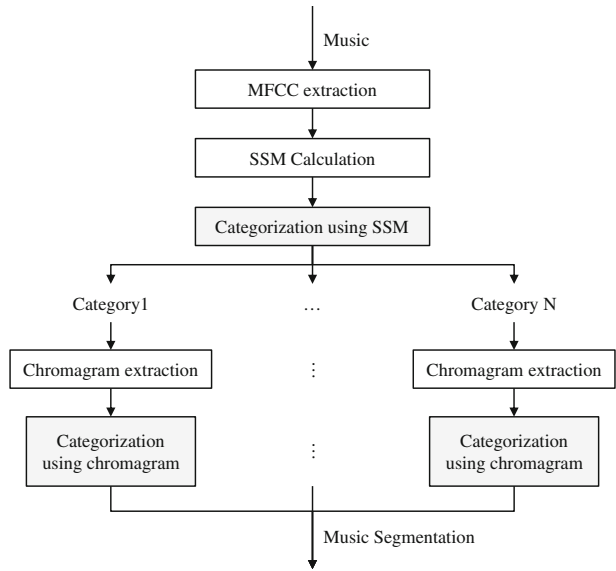
In this section, we describe the two-stage categorization method for feature-based music segment classification. In the first stage, the segments with similar timbre are grouped together using the

**Fig. 5** Music segmentation





**Fig. 6** Two-stage categorization method

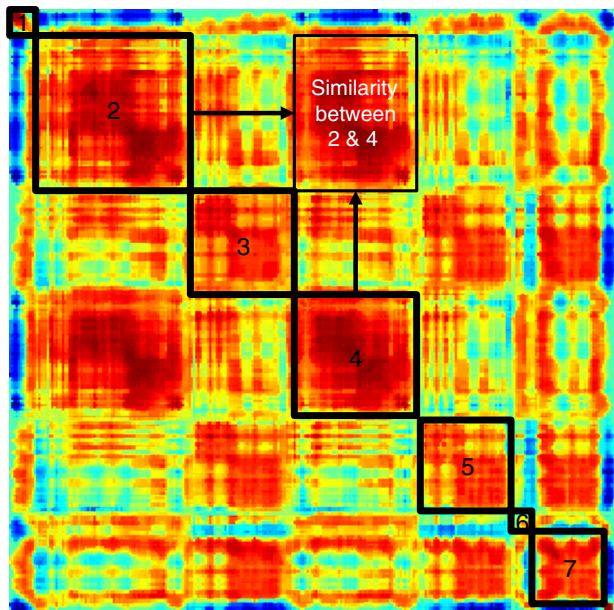


self-similarity matrix. In the second stage, the segments in each group are subgrouped according to their chromagram sequence. The overall steps for categorization are illustrated in Fig. 6.

### 5.1 Categorization using the self-similarity matrix

After segmentation, the music clip is divided into a few segments. However, such segmentation does not have any structural information. Music structure such as repeating segments or

**Fig. 7** Similarity measurement among the segments



neighboring similar segments can be defined by comparing the segments in the self-similarity matrix. Figure 7 shows an example of the comparison of the segments. For comparing the 3th and 5th segments, the average of the similarities in the intersection area in the vertical and horizontal directions is calculated. When the average similarity is high, we assume that the two segments are very similar and have the same state. On the basis of this intuition, we perform categorization from the segments with the highest similarity to the segments with the lowest similarity. The segments are categorized with their highest similarity pair until all the segments are categorized. As a result, the music clips typically have three to five categories Fig. 8.

## 5.2 Categorization using chromagram

Categorization on the basis of the self-similarity matrix considers timbre feature only for segment grouping. As a result, the segments with similar timbre are categorized together. In this work, for more delicate categorization, we perform additional comparison for chromagram sequences of segments using the cross-correlation function. Figure 9 shows the detailed steps for categorization, and Fig. 10 shows an example of the chromagram sequence comparison.

---

**Algorithm** Categorization using self-similarity matrix

**Input:** self-similarity matrix  $SSM$ ,  
 music segments  $\{sg_1, \dots, sg_n\}$   
 segmentation boundaries  $\{b_1, \dots, b_{n+1}\}$   
 where  $b_1=0$  and  $b_{n+1}=\text{length of song}$

**Output:** category sequence  $C = \{c_1, \dots, c_n\}$

---

**initialize** segment similarity matrix  $SGSIM$  with  $n \times n$  size

**for**  $i = 1$  to  $n$  **do**

**for**  $j = 1$  to  $n$  **do**

$SGSIM(i,j) = \text{average of } SSM(m,n)$   
 where  $m=\{b_i, \dots, b_{i+1}\}$  and  $n= \{b_j, \dots, b_{j+1}\}$

**end**

**end**

**initialize** category sequence  $C = \{c_1, \dots, c_n\}$  with *null* value

// set threshold  $\theta$  which distinguishes similar segments

**while** all  $c \neq \text{null}$  **do**

**find** segments  $sg_i, sg_j$  having maximum similarity in  $SGSIM(i,j)$

**if**  $SGSIM(i,j) > \theta$  **then**

**if** both  $c_i$  and  $c_j$  are null **then**  
       assign new category to  $c_i$  and  $c_j$

**if** either  $c_i$  or  $c_j$  is null **then**  
       copy non-null category to null category

**else**  
     assign new category to  $c$

**end**

**remove**  $sg_i, sg_j$  from  $SGSIM(i,j)$

**end**

**return**  $C$

---

**Fig. 8** Categorization using the self-similarity matrix

---

**Algorithm** Categorization using chromagram

**Input:** category sequence  $C = \{c_1, \dots, c_n\}$

**Output:** category sequence  $C' = \{c'_1, \dots, c'_n\}$

---

**initialize** category sequence  $C' = \{c'_1, \dots, c'_n\}$  with null value

**for** each category  $g$  **do**

**initialize** segment similarity matrix **SGSIM2** with  $m \times m$  size

where  $m$  is the number of segments in the category

**for** each segment  $i$  in category  $g$  **do**

**for** each segment  $j$  in category  $g$  **do**

**extract** successive chromagram  $CH_i(t), CH_j(t)$  from segment  $i, j$

$CHSIM = \max(\text{cross-correlation between } CH_i(t) \text{ and } CH_j(t))$

// set threshold  $\theta$  which distinguishes similar segment

**if**  $CHSIM > \theta$  **then**  $SGSIM2(i, j) = 1$

**else**  $SGSIM2(i, j) = 0$

**end**

**for** each column in **SGSIM** **do**

$k = \text{find}$  row indices with value 1

**allocate** new category to  $c'_k$

**if** all segments in category  $g$  are allocated **then** break

**end**

**return**  $C$

---

**Fig. 9** Categorization using chromagram

## 6 Experimental results

### 6.1 Segment boundary hit accuracy

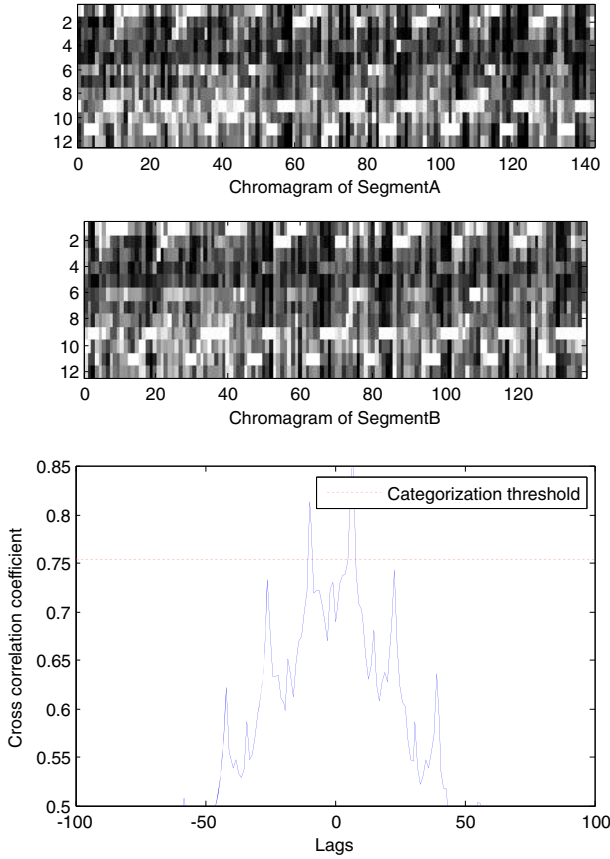
For music segmentation, we collected 50 popular songs over various genres. After listening to the songs several times, the subjects were requested to define the boundary set  $b_H$ , which denotes the music structural changes, such as the transition from verse to chorus or bridge, and the music mood changes based on personal perception. Another boundary set  $b_A$  is defined by our proposed method. By comparing the two boundary sets  $b_H$  and  $b_A$ , we estimate the effectiveness of our method. Here, we consider that the two boundaries are same when their interval difference is less than 3 s. The average precision and recall accuracies are summarized in Table 1.

### 6.2 Comparison of the segmentation methods

In this experiment, we compare the proposed segmentation method (SRSD) with the CBK method. The proposed segmentation method uses the sum of row-wise standard deviations of the self-similarity matrix. The CBK method uses the checkerboard kernel for segmenting the music clips on the basis of the self-similarity matrix. Figure 11 shows the comparison of the two methods for segmenting 50 popular songs. As shown in this figure, the proposed method outperforms in accuracy and computation time.

### 6.3 Comparison of the categorization methods

For evaluate the effectiveness of the two-stage categorization method, 50 popular songs are categorized using (i) a self-similarity matrix and (ii) a self-similarity matrix and chomagram, and



**Fig. 10** Example of chromagram comparison

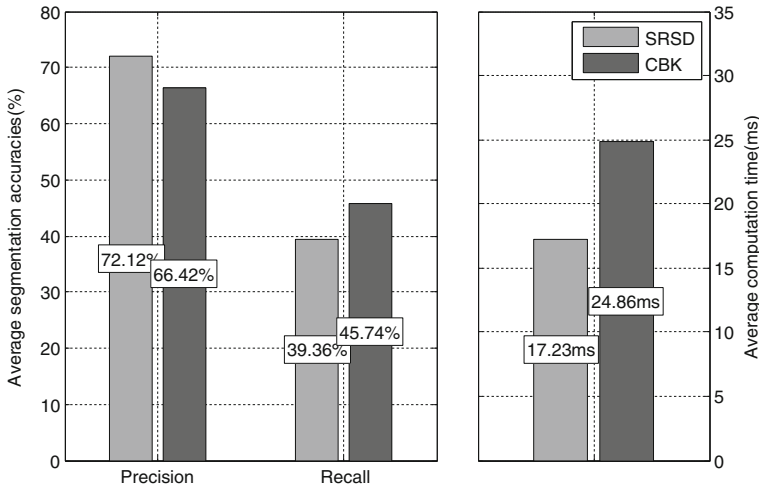
the segmentation results are presented to ten subjects for evaluation. The subjects listened to all the segments in the category of 50 popular songs and evaluated whether those segments are really similar. Fig. 12 shows the results of the user evaluation for 50 songs. As mentioned earlier, the songs cover seven genres. We calculate the average of the categorization accuracy by genre because the music structure differs depending on the genre. Overall, the average categorization accuracy is 74.86 % and 80.4 % for single- and multi-stage categorization, respectively.

### 7 Conclusion

In this paper, we proposed a method for music structure analysis on the basis of segmentation and categorization. Music segmentation was performed using a self-

**Table 1** Average segmentation accuracies for 50 songs

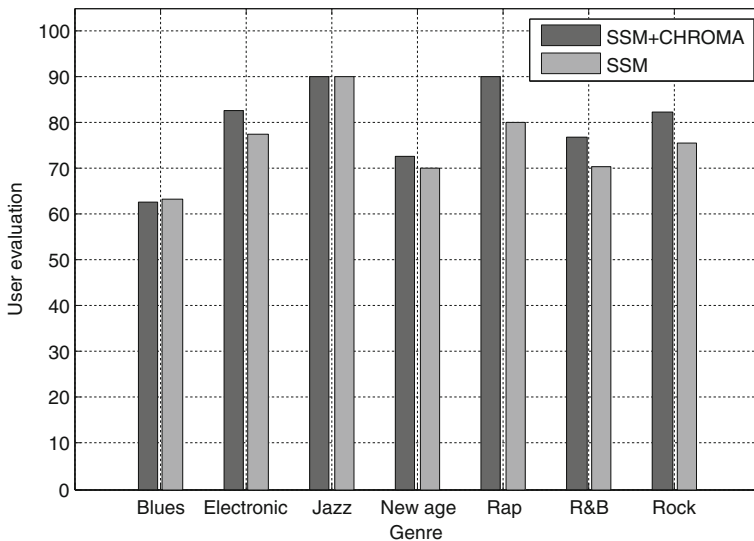
| Precision | Recall  |
|-----------|---------|
| 72.12 %   | 39.36 % |



**Fig. 11** Comparison of the two segmentation methods

similarity matrix, which is constructed by calculating the similarities among the timbre features. For segment categorization, we presented a two-stage categorization method, which uses the timbre and chromagram features of the music clips. For evaluating the performance of our method, we carried out experiments for songs of seven genres. The experimental results show that our method achieves reasonable performance with regard to music segmentation and categorization.

A few important issues still need to be tackled further. Basically, our scheme belongs to the state approach. Even though this approach is computationally less



**Fig. 12** Performance of the two categorization methods



expensive and more robust than the sequence approach, it is poor at performing delicate structure analysis. Future work involves combining the sequence approach and the state approach to facilitate more delicate structure analysis with less expensive computations. We also wish to further investigate how to apply our method in music retrieval and recommendation.

**Acknowledgments** This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF-2013R1A1A2012627) and the MSIP(Ministry of Science, ICT&Future Planning), Korea, under the C-ITRC(Convergence Information Technology Research Center) support program (NIPA-2013-H0301-13-3006) supervised by the NIPA(National IT Industry Promotion Agency).

## References

1. AllMusic. <http://www.allmusic.com/>. Accessed 24 October 2013.
2. Cooper M, Foote J (2002) Automatic music summarization via similarity analysis. Proceedings of the international conference on musical information retrieval (ISMIR), pp 81–85
3. Cooper M, Foote J (2003) Summarizing popular music via structural similarity analysis. IEEE workshop on applications of signal processing to audio and acoustics, pp 127–130
4. Foote J (1999) Visualizing music and audio using self-similarity. Proceedings of ACM international conference on multimedia (ACM MM). pp 77–80
5. Foote J (2000) Automatic audio segmentation using a measure of audio novelty. Proceedings of IEEE international conference on multimedia and expo (ICME), vol. 1. pp 452–455
6. Fujishima T (1999) Realtime chord recognition of musical sound: a system using common lisp music. Proceedings of international computer music conference (ICMC), pp 464–467
7. Jun S, Hwang E (2013) Music segmentation and summarization based on self-similarity matrix. Proceedings of the 7th international conference on ubiquitous information management and communication. 82:1–4
8. Jun S, Rho S, Hwang E (2010) Music retrieval and recommendation scheme based on varying mood sequences. *Int J Semant Web Inf Syst* 6(2):1–16. doi:10.4018/jswis.2010040101
9. Kaiser F, Sikora T (2010) Music structure discovery in popular music using non-negative matrix factorization. Proceedings of international conference on music information retrieval (ISMIR), pp 429–434
10. Klapuri A (1999) Sound onset detection by applying psychoacoustic knowledge. Proceedings of IEEE international conference on acoustics, speech, and signal, vol.6. pp 3089–3092
11. Logan B (2000) Mel frequency cepstral coefficients for music modeling. Proceedings of international conference on music information retrieval (ISMIR)
12. Lu L, Wang M, Zhang H-J (2004) Repeating pattern discovery and structure analysis from acoustic music data. ACM SIGMM international workshop on multimedia information retrieval, pp 275–282
13. Maddage NC, Xu C, Kankanhalli MS, Shao X (2004) Content-based music structure analysis with applications to music semantics understanding. Proceedings of ACM international conference on multimedia (ACM MM), pp 112–119
14. Paulus J, Klapuri A (2009) Music structure analysis using a probabilistic fitness measure and a greedy search algorithm. *IEEE Trans Audio Speech Lang Process* 17:1159–1170. doi:10.1109/TASL.2009.2020533
15. Peeters G (2004) Deriving musical structures from signal analysis for music audio summary generation: “Sequence” and “State” approach. *Computer music modeling and retrieval*. Springer Berlin, Heidelberg, pp 169–185
16. Peeters G (2007) Sequence representation of music structure using higher-order similarity matrix and maximum-likelihood approach. Proceedings of the international conference on musical information retrieval (ISMIR), pp 35–40
17. Rabiner L, Juang B-H (1993) *Fundamentals of speech recognition*. Prentice Hall
18. Serrà J, Müller M, Grosche P, Arcos JL (2012) Unsupervised detection of music boundaries by time series structure features. Proceedings of twenty-Sixth AAAI Conference on Artificial Intelligence, pp 1613–1619
19. Tzanetakis G, Cook P (2002) Musical genre classification of audio signals. *IEEE Trans Speech Audio Process* 10:293–302. doi:10.1109/TSA.2002.800560
20. Wang M, Lu L, Zhang H-J (2004) Repeating pattern discovery from acoustic musical signals. Proceedings of IEEE international conference on multimedia and expo (ICME), vol. 3, pp 2019–2022



**Sanghoon Jun** received his B.S. and M.S. Degree in Electrical engineering from Korea University, Korea, in 2008, respectively. Currently, he is a Ph.D. candidate at the Electrical Engineering faculty in the School of Electrical Engineering. His current research interests include music retrieval and recommendation, multimedia systems, machine learning, emotional computing and Web applications.



**Seungmin Rho** received his M.S. and Ph. D Degrees in Computer Science from Ajou University, Korea, in Computer Science from Ajou University, Korea, in 2003 and 2008, respectively. He visited Multimedia Systems and Networking Lab. in Univ. of Texas at Dallas from Dec. 2003 to March 2004. Before he joined the Computer Sciences Department of Ajou University, he spent two years in industry. In 2008–2009, he was a Postdoctoral Research Fellow at the Computer Music Lab of the School of Computer Science in Carnegie Mellon University. He had been working as a Research Professor at School of Electrical Engineering in Korea University during 2009–2011. In 2012, he was an assistant professor at Division of Information and Communication in Baekseok University. Also, he was a faculty of Department of Multimedia at Sungkyul University in Korea during 2013. Now he is currently an associate professor at Department of Computer Engineering, Mevlana University, Konya, Turkey. His current research interests include database, big data analysis, music retrieval, multimedia systems, machine learning, knowledge management as well as computational intelligence. He has published 100 papers in refereed journals and conference proceedings in these areas. He has been involved in more than 20 conferences and workshops as various chairs and more than 30 conferences/workshops as a program committee member. He has edited a number of international journal special issues as a guest editor, such as Multimedia Systems, Information Fusion, Engineering Applications of Artificial Intelligence, New Review of Hypermedia and Multimedia, Multimedia Tools and Applications, Personal and Ubiquitous Computing, Telecommunication Systems, Ad Hoc & Sensor Wireless Networks and etc. He has received a few awards including Who's Who in America, Who's Who in Science and Engineering, and Who's Who in the World in 2007 and 2008, respectively.



**Eunjung Hwang** received his B.S. and M.S. degrees in Computer Engineering from Seoul National University, Seoul, Korea, in 1988 and 1990, respectively; and his Ph.D. degree in Computer Science from the University of Maryland, College Park, in 1998. From September 1999 to August 2004, he was with the Graduate School of Information and Communication, Ajou University, Suwon, Korea. Currently, he is a member of the faculty at the School of Electrical Engineering, Korea University, Seoul, Korea. His current research interests include database, multimedia systems, information retrieval, XML, and Web applications.