Music summarization using visual information of music and clustering method

Sangho Kim[1], Mi-kyong Ji[2], Hoi-Rin Kim[3]
School of Engineering, Information and Communications University[1 2 3]
{ksh[1], lindaji[2],hrkim[3]}@icu.ac.kr

Abstract

In this paper, we present effective methods for music summarization which summarize music automatically. It could be used for sample music of on-line digital music provider or some music retrieval technology. When summarizing music, we use different two methods according to music length. First method is for finding sabi or chorus part of music which can be regarded as the most important part of music and the second method is for extracting several parts which are in different structure or have different mood in the music. Our proposed music summarization system is better than conventional system when structure of target music is explicit. The proposed method could generate just one important segment of music or several segments which have different mood in the music. Thus, this scheme will be effective for summarizing music in several applications such as online music streaming service and sample music for T-commerce.

Keyword: Music summarization, Music structure, Digital audio

.

1. Introduction

Nowadays, digital music services are highly important and very attractive commercially. In a Ipsos-Reid survey, more than half of consumers aged 25-34 have downloaded MP3 files onto home computers, storing on average more than 700 files [1]. So, locating and browsing thousands of tracks is a considerable data management problem [2]. While, online music streaming companies give sample music for prospective buyer of music. But the sample music is just a 1 minute segment from the exact beginning of music. So, sometimes, the sample music includes just intro part of music. It may be difficult for consumers to determine whether he or she pay for the music. To solve these problems, there are so many methods to summarize music automatically. Sometimes, those methods use 2D similarity matrix [3] or clustering method [4]. But those conventional methods did not considered duration of music. If the duration of the music is very long and the music has very different structures, it will be desirable to summarize the music by combining various parts of the music. If it is not, we can extract just one important part of the music. So, here, we introduce a new approach to automatically summarize a music track by using two different methods which are included in one system for music summarization. Two methods use clustering algorithm and 2D similarity matrix, respectively. The methods are very popular and frequently used for music summarization. But we use the methods differently and efficiently compared to other existing methods by using variable threshold and music knowledge. Thus, we can summarize music properly based on its length and structure and it will be helpful to browse music or provide sample music for online music streaming companies and its customers.

2. Proposed system overview

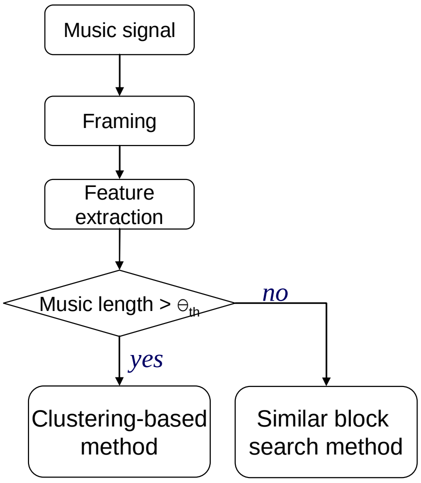


Figure 1. Overall system

As depicted in Figure 1, music signal is split into several frames with uniform length. After the hamming window is used for block processing, we extract Mel filter bank energy, spectral centroid, spectral flux and spectral roll-off as a feature vector per frame. And then, if the duration of music is more than predefined threshold, Θ_{th}. The system use clustering-based method for music summarization. If it is not, the Similar Block Search (SBS) method is used.

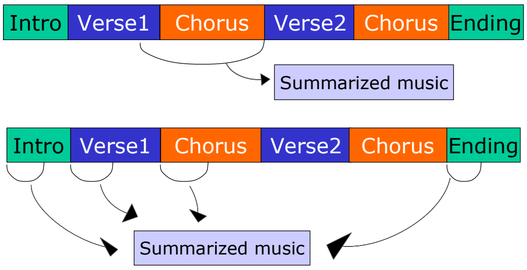


Figure 2. Visual representation for goal of similar block search method (top) and clustering-based method (bottom)

As can be seen in Figure 2, the SBS method is for finding sabi or chorus part of music which can be regarded as the most important part of music and the clustering-based method is for extracting several parts which are in different structure or have different mood in the music. These two methods will be explained in detail at later sections.

2-1. Feature extraction

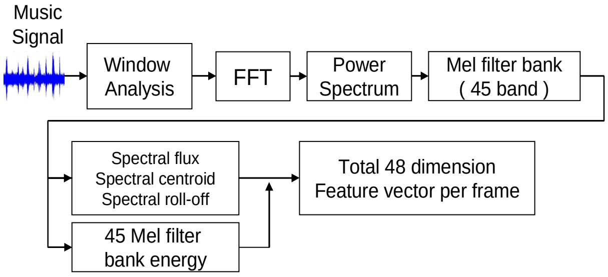

Music Signal
Window Analysis
FFT
Power Spectrum
Mel filter bank (45 band)
Spectral flux
Spectral centroid
Spectral roll-off
45 Mel filter bank energy
Total 48 dimension Feature vector per frame

Figure 3. Feature extraction process

To extract feature, we used 45 Mel filter bank energy and other three features such as spectral flux, spectral centroid and spectral roll-off [5]. So, total dimension of feature vector is 48. Of course, we tried to use Mel Frequency Cepstral Coefficients (MFCC). But Mel filter bank energy was slightly better than MFCC in our method so we used Mel filter bank energy and some timbre feature like spectral centroid instead of using just MFCC. All of the process for feature extraction can be seen in Figure 3.

2-2. 2D similarity matrix

We use 2D similarity matrix for the SBS method. The matrix, visual information, is very well known and popular for music summarization. To visualize music or audio, the similarity measure S(i,j) is calculated for all frame combinations, hence frame indexes i and j. Then an image is constructed so that each pixel at location i, j is given a grayscale value proportional to the similarity measure, by scaling the similarity values such that the maximum value is given the maximum brightness [6]. So, here is a process to construct a 2D similarity matrix. Firstly, segment input music signal with uniform length. Secondly, extract 48-order feature vector at each frame. Thirdly, when V_i, V_j are feature vectors of i-th, j-th frame, find frame to frame similarity S(i,j) using equation (1).

$$S(i,j) = \frac{\upsilon_i \bullet \upsilon_j}{\|\upsilon_i\| \cdot \|\upsilon_j\|} \qquad (1)$$

Finally, we can construct an image using the similarity as depicted in Figure 4 as follows.

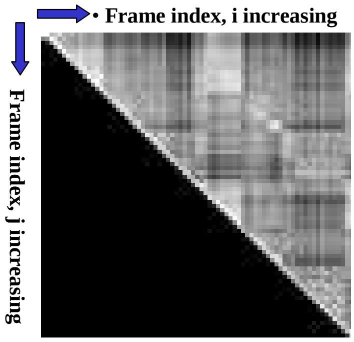


Figure 4. 2D similarity matrix

3. Proposed method

3-1. Similar block search method

In this section, we explain assumptions for the SBS method and the method itself in detail. Generally, there exists some proportion of music structure, especially Korean pop music called Gayo, Pop and some rock music. First assumption is that intro and first verse part of music consist of about 27% of whole duration of the music so we could think the starting point of chorus or sabi part of music will be roughly located on the 27th percentile of the music as described in Figure 5. Of course, the starting point of 1st chorus might not be the exact 27th percentile of music in many cases. To find the specific data, we selected 50 songs randomly among a thousand of Korean pop music and investigate the starting point of chorus part. Thus, we could assume that the starting point of 1st chorus is located between the 24th percentile and the 31st percentile of music.

Second assumption is that the chorus part exists, at least, 2 times in the music. The assumption can be reinforced by the definition of chorus or refrain, that is, the chorus or refrain is a the line or lines that are repeated in music and the refrain or chorus often sharply contrasts the verse melodically, rhythmically, and harmonically, and assumes a higher level of dynamics and activity, often with added instrumentation [7]. Thus, these two assumptions are fundamentals for the SBS method.

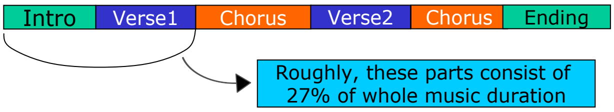


Figure 5. Proportion of music, especially, Korean pop (Gayo)

Firstly, we make a 2D similarity matrix, ***S(i,j)***. And we select a N by N base block. The starting point of the block is ranged from Θ_{low} and Θ_{upper} which are the lower threshold and upper threshold values calculated from our first assumption. After that, we search similar block which minimize d(i,j), the block distance between the base block and the similar block where i is the frame index of the base block and j is the frame index of the similar(or searching) block. The d(i,j) is calculated like as follows.

$$d(i,j) = \frac{1}{N^2}\sum_{l=0}^{N}\sum_{k=0}^{N}|S(i+l,i+k) - S(j+l,j+k)| \quad (2)$$

$$i^* = ArgMin\ d(i,j) \quad (3)$$

, where $\Theta_{low} < i < \Theta_{upper}$, $i+N < j < N_{frame}-N$

N_{frame} is the number of total frames and N is predefined value. After finding the i-th frame, i^* which minimize d(i,j), we adjust the predefined N value to minimize the d(N,j) using equation (2) similarly.

$$d(N,j) = \frac{1}{N^2}\sum_{l=0}^{N}\sum_{k=0}^{N}|S(i^*+l,i^*+k) - S(j+l,j+k)| \quad (4)$$

$$N^* = ArgMin\ d(N,j) \quad (5)$$

,where $\Theta_{lb} < N < \Theta_{ub.}$, $i^*+N < j < N_{frame}-N$

Θ_{lb} and Θ_{ub} are predefined lower bound and upper bound, respectively. Therefore, the best summary is then the excerpt of length N^* starting at i^* and ending at N^*+i^* in case of the SBS method. Of course, if we do not adjust summary length, we can get better results, sometimes. Simple graphical explanation is as follows in Figure 6. The figure shows briefly and clearly how the SBS method works.

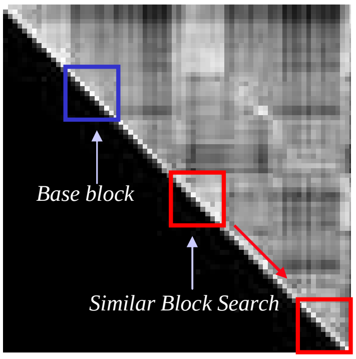
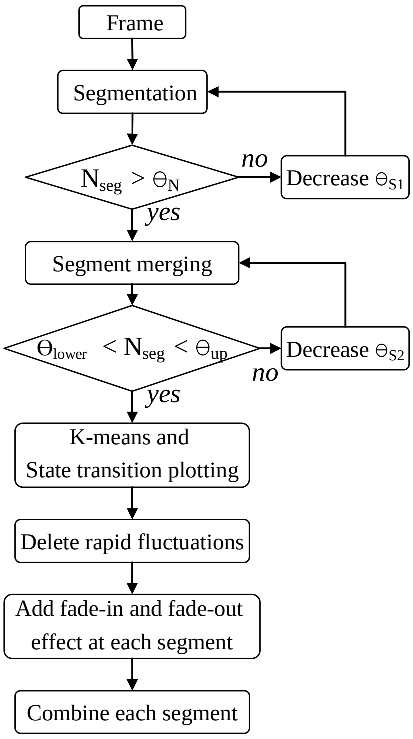


Figure 6. Graphical explanation for the method

3-2. Clustering-based method

In this section, we explain clustering-based method in detail. As explained, the clustering-based method is for extracting several parts which are in different structure or have different mood in the music.



Frame
Segmentation
$N_{seg} > \Theta_N$
no
Decrease Θ_{S1}
yes
Segment merging
$\Theta_{lower} < N_{seg} < \Theta_{up}$
Decrease Θ_{S2}
no
yes
K-means and State transition plotting
Delete rapid fluctuations
Add fade-in and fade-out effect at each segment
Combine each segment

Figure 7. Flowchart of clustering-based method

Firstly, segment frames using feedback and calculate means of each segment. If the frame to frame similarity between i-th frame and (i+1)-th frame is higher than threshold and the similarity between (i+1)-th frame and (i+2)-th frame is less than the threshold, vice versa, we detect the frame index, (i+1) as a one of the point for segmentation. But if we fix the threshold, the segmentation technique will not generate enough number of segments to merge segments when changes in the signal content of some music is not rapid. So initial threshold is set to almost maximum similarity value and decreased by feedback. Thus, we use variable threshold to get enough segments to merge those segments in the next process. Secondly, calculate segment to segment similarity. Each segment was generated in previous process. Thirdly, merge similar segments using feedback until the number of segments after merging process reach predefined range and calculate each mean of the final segments. After that, use mean values of the feature vectors of each merged segment for initial codeword of k-means algorithm. Then, use k-means algorithm to get final codewords. And then, plot state transition of the music using the final codewords. Finally, several post-processing such as deleting fluctuation of state transition, adding fade-in and fade-out effect to each segment and combining each segment will be conducted. Among the process, deleting or filtering fluctuations of state transition is important in our method because generally, pop music has various short-time sound effects such as cymbal sound and electronic sound which cause feature vectors of adjacent frames to be very dissimilar although the frames are similar or in same structure of music, actually. Figure 7 shows overall procedure of the method in detail. In the figure, N_{seg} is a number of segments generated, Θ_N is a predefined number of segments before merging, Θ_{S1} is a threshold of similarity before merging, $\Theta_{lower,\ up}$ is a predefined number of segments after merging, Θ_{S2} is a threshold of similarity after merging. And figure 8 shows an example of state transition of a song, "Blind love" by *Seo, Tai-Ji* who is very well-known musician in Korea. In the figure, frame index is increasing along the horizontal axis and state or codeword index is explicit along the vertical axis. If the feature vector of the 1st frame has minimum distance with the 1st codeword of codebook generated by k-means algorithm, we can mark the state of the 1st frame on proper position of vertical axis. Likewise, we can plot the state transition along time.

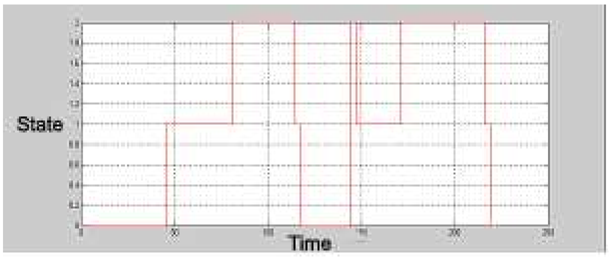


Figure 8. State transition of the feature vectors of "Blind love" (by Seo Tai Ji) along time

4. Performance evaluation

4-1. The SBS method

To evaluate the SBS method, we adopted MOS test which is subjective evaluation dependent on each human perception. Firstly, we choose some students those who have some experiences or enough knowledge of playing musical instruments or writing music. Secondly, three questions are asked to the listeners like as follows and score is ranged from 1 to 5. And 5 is the best score.

Q1) Does the summarized music include chorus or sabi part of the music?

Q2) Does the summarized music include verse part properly before chorus part?

Q3) How about overall quality as a summarized music?

The first question is about how well or how much the summarized music includes the chorus part of music which is the most important part of music. Of course, the importance could be dependent on individuals. Some individuals could think the most important part of some music is solo part or ending part. In case of typical music, however, the chorus part will be the most important. The second question is about how much the amount of verse part precedes the chorus part properly. In general, good music summary include the verse part a little before chorus part begin because it could be more attractive for listeners than music summary exactly starting chorus part without any verse part. For the test, we selected two genres of music such as Gayo and Rock. Each genre contains 20 songs. The length of the testing songs is from 2min 5sec to 3min 59sec. The results are as follows.

Table 1. Evaluation for the SBS method

Genre	Q1	Q2	Q3
Gayo	4.65	3.6	4.4
Rock	4.5	3.4	3.95

Table 1 show that this method was good for grasping chorus part of music. In addition, overall quality was also good as a summary. To confirm these good results, we did conventional overlap ratio evaluation which evaluates the ratio between manually selected region and automatically generated summary. Table 2 shows a result of the overlap ratio evaluation. From the results, we could know our method is good for extracting chorus part of music.

Table 2. Overlap Ratio result

Genre	Fixed length (30sec)	Variable length (20sec~30sec)
Gayo	0.82	0.63
Rock	0.78	0.78

4-2. The clustering-based method

To evaluate the clustering-based method, we also use MOS test. But questions are different. The two questions are as follows.

Q1) How many similar segments (or part) are there in the summarized music?

Q2) How many segments (or part) are included in same structure?

The answer for the first question is the Number of Similar Part (NSP) and the answer for the second question is the Number of Same Structure (NSS). We use these two factors for the evaluation. For example, assume that a listener checked that the NSP of song 1, 2 is 0, 2, respectively and the NSS of song 1, 2 is 2, 3, respectively. Additionally, NSP ratio (NSPR) and NSS ratio (NSSR) could be easily calculated like as follows. Table 3 shows the example clearly about how we can evaluate the clustering-based method.

Table 3. An example

Song	Segments	NSP	NSS	NSPR	NSSR
1	5	0	2	0/5	2/5
2	5	2	3	2/5	3/5
Average	5	2/2	5/2	(0+2/5)/2	(2/5+3/5)/2

In this case, Average of NSPR is 0.2 and average of NSSR is 0.5. If these ratios are very small, we could think the automatically generated several segments of music are not similar and that is exactly what we want. For evaluation, we selected three genres of music such as Gayo, very long-length rock and middle-length rock. Gayo, long-length rock and middle-length rock genre contains 50, 20, 30 songs, respectively and results of the evaluation are as follows.

Table 4. Evaluation for the clustering-based method

Genre	NSPR	NSSR
Gayo	0.11	0.19
Long-length Rock	0.04	0.14
Rock	0.04	0.12

From the results, we could find the NSPR is less than the NSSR, that is, our method is more suitable for generating different part of music although they are included in same music structure. The one of the reason is why we used relatively low-level feature vector such as spectral energy. However, this method is fast and robust. It means this method could be used for summarizing almost music even though the variations of music in the signal contents are very steady.

5. Conclusion and future works

Our proposed system for music summarization is fast and could generate two kinds of summary according to music length. To summarize one song, about 1 or 2 seconds is needed so it can be applied to real-time applications such as providing sample music on the web or for the T-commerce. In addition, this system will be helpful for individuals who want to find a song efficiently or listen to just some part of the music. But this system is suitable for music with explicit structure and repetitive patterns. So we need to study more about music information and human perception of understanding music structure in several cases. In addition, we need to gather more evaluation results and find more reasonable method for evaluation.

Acknowledgement

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the Digital Media Lab. support program supervised by the IITA (Institute of Information Technology Assessment)

References

[1] Ipsos-Reid, “Digital Music Behavior Continues to Evolve,” press release, January 31, 2002. http://www.ipsos-reid.com

[2] Matthew Cooper and Jonathan Foote, “Summarizing popular music via structural similarity analysis,” *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, October, 2003

[3] Matthew Cooper and Jonathan Foote, “Automatic Music Summarization via Similarity Analysis,” *Proc. IRCAM*, pp. 81-85, Oct. 2002

[4] Geoffroy Peeters, Amaury La Burthe, Xavier Rodet, “Toward Automatic Music Audio Summary Generation from Signal Analysis,” *Proc. ISMIR*, Paris, 2002

[5] Eric Scheirer, Malcolm Slaney, “Construction and evaluation of a robust multifeature speech/music discriminator,” *Proc. ICASSP-97*, pp. 1331~1334, Apr 21-24, Munich, Germany

[6] Jonathan Foote, “Visualizing Music and Audio using Self-Similarity,” *Proc. ACM Multimedia Conference*, pp. 77~80, Orlando, Florida, November 1999.

[7] http://en.wikipedia.org/wiki/Refrain