# AUTOMATIC SUMMARIZATION OF MP3 MUSIC OBJECTS

*Chih-Chin Liu and Pang-Chia Yao*

Department of Computer Science and Information Engineering
Chung Hua University
Hsinchu, Taiwan 300, R.O.C.
ccliu@chu.edu.tw

**ABSTRACT**

In this paper, we propose an approach to automatically summarize MP3 music objects. In our approach, the MP3 music summary is constructed in three steps. First, the coefficients extracting from the output of the polyphase filters are used to compute the MP3 features. Based on these features, an MP3 music object can be automatically segmented into a sequence of MP3 phrases. Then the phrase clustering technique is applied to group similar phrases into a cluster. The phrases in a cluster can be considered as repeating patterns in the MP3 song. Finally, the RP-tree algorithm is applied to find the nontrivial repeating patterns such as chorus, refrains, themes, and motives of the song. Experiments are also performed and analyzed to show the effectiveness of the proposed method.

## 1. INTRODUCTION

As the explosive growth of the Internet and the progress of multimedia compression technologies, digital multimedia data such as images, video, and music can be found everywhere. There are various data formats designed for the storage of music information, which enable many new kinds of music applications such as on-line music store, music-on-demand, digital music library, and Internet radio. In these applications, to provide users a comprehensive overview about large amounts of music data, music summarization techniques are highly demanded. Previous works on music summarization can be found in [2][3][7][8][9]. Logan and Chu proposed an approach to build music summaries using key phrases [7]. In their approach, the Mel-cepstral coefficients extracted from music frames are used as the features to cluster fixed-length music segments. The longest section that contains the most frequently repeating frames is considered as the most interesting part of the song. Finally, 10 seconds music segment is selected from the longest section as the key phrase to summarize this song. Xu et al. pointed out that "Mel-cepstral coefficients cannot uniquely reflect the characteristics of music content" [9]. Therefore, they used spectral power, amplitude envelop, as well as the Mel-cepstral coefficients as the features for clustering music frames. Cooper and Foote presented a method for music summarization based on self-similarity analysis [2]. In their approach, an audio signal is divided into a sequence of fixed-length (92.87ms) frames. Then, the Mel-cepstral coefficients of each frame are extracted as the feature vector. The cosine distance between each pair of the feature vectors is computed to form a 2D similarity matrix . The most repeated segment can be found by summing the similarity matrix over the support of a segment and choosing the segment with the maximum summary score.

Previous techniques for music summarization mentioned above were proposed for music data in waveform or symbolic format (SMF). MP3 (MPEG-1 Audio Layer 3) is an ISO international standard for the compression of digital audio data [1]. Since MP3 can provide high compression ratio with CD-quality sound, there is a growing amounts of MP3 music data available on the Internet today. However, only keyword-based searching mechanisms are provided by the MP3 content providers (e.g. http://www.mp3.com/) and MP3 searching engines. In spite of its great success in many music applications, few research works were done on the analysis of MP3 music data. In our previous works, issues about content-based retrieval [5], and classification [6] of MP3 music were addressed. In this paper, we proposed an approach for automatically constructing MP3 music summary. In our approach, the MP3 music summary is generated in three steps. First, the coefficients extracting from the output of the polyphase filters are used to compute the MP3 features for segmentation. Then the phrase clustering technique is applied to group similar phrases into a cluster. The phrases in a cluster can be considered as repeating patterns in the MP3 song. Finally, the RP-tree algorithm [4] is applied to combine repeating phrases into repeating segments and to find the nontrivial

* This work was supported by the Republic of China National Science Council under Contract No. NSC-92-2213-E-216-015.

repeating segments such as chorus, refrains, themes, and motives of the song as the MP3 music summary.

## 2. ARCHITECTURE OF AN MP3 MUSIC SUMMARIZATION SYSTEM

Traditionally, music summaries are provided by CD publishers as the advertisements for promoting their new published albums. Now this technique is required for automatically summarizing music content on the Internet. Music summaries can be visual or auditory.

- Visual Music Summaries: The most common things to attract CD buyers are the cover images or posters of the albums. The information provided by the cover images are album name, artists, and song list. However, users can hardly get any idea about the music styles or genres of the songs with an album image. The other typical music summary of a song is its *score*. However, it is inadequate for naïve users. It is still a great challenge to design certain kinds of fingerprints or thumbnails to visually represent the content of a song.
- Auditory Music Summaries: The straightforward way to illustrate the content of a song is to play the most representative or beautiful parts of it. These representative parts are generally called *key phrases*. Typically, the key phrases are *motives* or *themes* for classical music. While for pop songs, the key phrases are their *refrains* or *choruses* in general. No matter the key phrases are motives, themes, refrains, or chorus, their common property is that they all repeat more than once in a music object. This property is the basic assumption in all music summarization methods proposed.

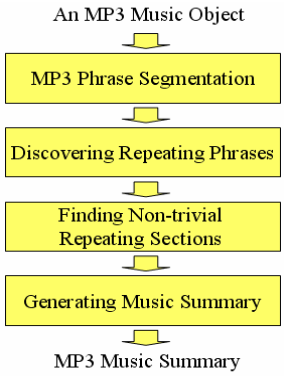


**Figure 1. The architecture of an MP3 music summarization system.**

Figure 1 shows the architecture of an MP3 summarization system. The music summary for an MP3 music object is constructed in four stages. In the first stage, an MP3 music object is automatically segmented into a sequence of MP3 phrases. For each MP3 phrase, three kinds of MP3 features are computed to represent it. Based on the MP3 features, similar MP3 phrases are grouped into a cluster and can be considered as repeating phrases. We can assign each cluster a unique cluster ID to label it. The original MP3 music object can be transformed into a string of cluster IDs. Thus, RP-tree algorithm can be applied to find all non-trivial repeating patterns within the string of cluster IDs. Typically, the longest non-trivial repeating pattern is their theme (refrain) while the shorter but more repeated patterns are their motives. We then output these theme and motives as the music summary for the MP3 music object.

## 3. PREPROCESSING

### 3.1. MP3 Feature Extracting

To reduce computing overhead, it is highly required that the music features should be extracted from the compressed domain. Therefore, the MP3 features used in our system are extracted during the MP3 decoding process. The basic unit of an MP3 bitstream is a *frame*. According to the standard [1], an MP3 bitstream is unpacked and dequantized, frame by frame, into *MDCT* (modified discrete cosine transform) *coefficients*. The MDCT coefficients (576 frequency lines) are then mapped into subsamples (32 subbands) using inverse MDCT. These subsamples also called *the polyphase filter coefficients*. Finally, the subsamples are synthesized into the original audio signal (PCM audio). Both the MDCT coefficients and the polyphase filter coefficients can be used to compute the MP3 features.

### 3.2. MP3 Phrase Segmentation

In the first step of the segmentation procedure, the non-vocal parts of a song to be segmented are removed using a vocal/non-vocal discriminating technique. Then the polyphase filter coefficients are extracted for each MP3 frame. Let *S[i][j]* represents the *j*-th subsample at the *i*-th sunband. The frame energy (FE) of each frame can be computed by the following equation.

$$PC_i = \sum_{j=1}^{36} (S[i][j])^2 \quad (1)$$

$$FE = \sum_{i=1}^{32} PC_i \quad (2)$$

The FE of each frame can be used as the feature for detecting MP3 phoneme break points by applying a heuristic rule which is developed based on the ADSR (attack, decay, sustain, and release) property of the phoneme envelope. Finally, an MP3 phrases can be found by grouping several continuous phonemes. Three cost factors, i.e., the position of the final, the average length of phrase, and the length of the last phrase, can be used to

develop a cost function for finding the best way to segment (or group) a sequence of adjacent MP3 phonemes into a sequence of MP3 phrases.

### 3.3. MP3 Phrase Clustering

Having defined the similarity metrics for the comparison of two MP3 phrases, the last step for finding repeating phrases is to group similar MP3 phrases into a cluster. In this paper, a modified *k-means* clustering algorithm is applied. The phrases in a cluster should follow the constraint that the distance between each pair of them must be smaller than a predefined radius *R*.

## 4. THE CONSTRUCTION OF MP3 MUSIC SUMMARY

### 4.1. Finding Non-Trivial Repeating Sections

After the MP3 phrases are clustered, we can find all repeated phrases in a song. The last problem is how to choose the representative phrases as the music summary. Consider the song "Yesterday" by The Beatles. It will be segmented into 27 MP3 phrases. The first phrase (intro) and last phrase (coda) will be removed since they are non-vocal. Applying phrase clustering technique, seven clusters are constructed as shown in Table 1. Each cluster is labeled with a letter to identify it. We can choose the most repeated phrases, i.e., cluster A, B, C, D, and E, as the music summary. However, if we check the structure of this song, we can find that the combination of these five clusters, i.e. A-B-C-D-E, will also repeat in this song. In fact, this song has two themes: A-B-C-D-E and F-G as shown in Figure 8. Any part of a theme will also repeat in this song. Therefore, we should find a more compact while meaningful representation for the clustering result.

**Table 1. Seven clusters constructed for the song "Yesterday".**

| *Cluster* | *Number of phrases* | *Phrase IDs* |
|---|---|---|
| *A* | *4* | *1, 6, 13, 20* |
| *B* | *4* | *2, 7, 14, 21* |
| *C* | *4* | *3, 8, 15, 22* |
| *D* | *4* | *4, 9, 16, 23* |
| *E* | *5* | *5, 10, 17, 24, 25* |
| *F* | *2* | *11, 18* |
| *G* | *2* | *12, 19* |

RP-tree is a data structure for efficiently finding all non-trivial repeating patterns in a string [4]. If we represent the song "Yesterday" using its cluster IDs, the original song can be transformed into the *phrase string* "A-B-C-D-E-A-B-C-D-E-F-G-A-B-C-D-E-F-G-A-B-C-D-E-E". Then we can apply RP-tree algorithm on this phrase string to find the *non-trivial repeating patterns* in this string. A trivial repeating pattern *X* is a substring of another repeating pattern *Y* and *freq(X) = freq(Y)*, where *freq(X)* and *freq(Y)* represent the number of appearance of X and Y, respectively. Since a trivial repeating pattern can be directly derived by taking the substring of the repeating pattern that contains it, it is less representative. Therefore we call it trivial and it can be removed.

We use an example to illustrate the procedure for building the RP-tree. In the first step, for each cluster of phrases, we create a leaf in the RP-tree to represent the repeating patterns of length 1. Then the repeating patterns of length 2 can be found by applying the *string-join* operation on the repeating patterns of length 1. Similarly, the repeating patterns of length 4 can be found by applying the string-join operation on the repeating patterns of length 2. After the repeating patterns whose length is power of 2 are found, we can derive the longest repeating pattern "ABCDEFG" by string-joining the repeating patterns "ABCD" and "DEFG". The constructing result is shown in Figure 2.

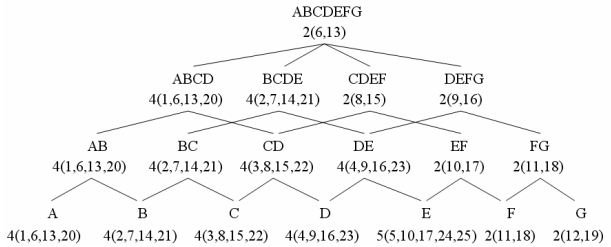


**Figure 2. The construction of the RP-tree for representing the repeating phrases in the song "Yesterday".**

After the RP-tree is constructed, the next step is to remove the trivial repeating patterns. The trivial repeating patterns are removed from the RP-tree in bottom-up manner. We first remove the trivial repeating patterns of length 1 by checking whether they are trivial. For example, since phrase string "A-B" contains phrase string "A", and both "A-B" and "A" repeat four times in the song, "A" is trivial and should be removed from the RP-tree. Similarly, "B", "C", "D", "F", and "G" are trivial and removed. The trivial repeating patterns of length 2 and length 4 are removed in the same way. We also find "ABCD" and "BCDE" can be merged into another non-trivial repeating pattern "ABCDE". Finally, only three non-trivial repeating patterns "ABCDEFG", "ABCDE", and "E" are left in the RP-tree as shown in Figure 3.

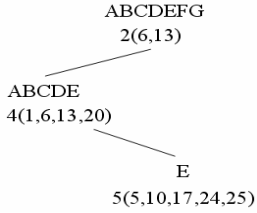


**Figure 3. The procedure for removing trivial repeating patterns in the RP-tree.**

### 4.2. Generating MP3 Music Summary

The leaves in an RP-tree are typically the most repeated while the shortest parts of a song. On the other hand, the root in an RP-tree is the longest repeating part of a song. According to the music theory, themes are defined as longer repeating patterns which represent subjects of a composition while motives are defined as shorter repeating patterns that are recognizable in a composition. Typically, the top nodes are the themes and the leaf nodes are the motives of the song. In our example, the longest repeating pattern "ABCDEFG" is composed of two themes "ABCDE" and "FG", and the most repeated pattern "E" is a motive of this song. The music summary can be represented in many ways. First, we can store them in the music databases to support content-based music data retrieval and browsing. Second, we can arrange them in a web page. Finally, we can use the audio descriptors and description schemes defined in MPEG-7 to store the summary result.

## 5. EXPERIMENTS

To show the effectiveness of the proposed method, a series of experiments are performed and analyzed. 100 famous songs were randomly picked to perform the following experiments.

The effectiveness of the MP3 music summarization technique is measured by *the precision rate* and *the recall rate*. The precision rate is defined as the number of repeating MP3 phrases correctly discovered divided by the total number of MP3 phrases discovered in an MP3 music object. The recall rate is defined as the number of repeating MP3 phrases correctly discovered divided by the total number of MP3 phrases found manually by music experts in an MP3 music object. The judgments are made by manually examining the original score information of the corresponding songs.

There are three kinds of feature vectors *FA* (based on polyphase filter coefficients), *FB* (based on MDCT coefficients), and *FC* (based on perceptually normalized MDCT coefficients) for measuring the similarity between two MP3 phrases. The goal of the first experiment is to find which kind of MP3 features is the most adequate one for MP3 music summarization. All MP3 features perform well in finding repeating MP3 phrases. However, the perceptually normalized MDCT coefficients out perform the other MP3 features in both the precision rate and the recall rate. The average precision rate is 79.4% while the average recall rate is 85.5% for the 100 test songs.

An MP3 song can be segmented into a sequence of MP3 phrases manually or automatically. The second experiment investigates the effect of phrase segmentation methods. The average recall rates using automatic and manual phrase segmenting techniques are 67% and 92%, respectively; while the average precision rates using automatic and manual phrase segmenting techniques are 64% and 86%, respectively. That is, about 25% recall drop and 22% precision drop will be paid for automatic phrase segmentation.

## 6. CONCLUSION

In this paper, we propose an approach to automatically summarize MP3 music objects. Compared with other music summarization methods proposed, three major advantages can be provided in our approach. First, since the features we used are extracted from the compressed domain, computing overhead is reduced. Second, instead of randomly segmenting a song into fixed-length sections, we segment an MP3 song into a sequence of MP3 phrases. Therefore, the themes and motives can be soundly kept in the summary more likely. Third, the RP-tree algorithm is applied to make the music summary more compact and representative. For the future work, we will try to find the most adequate features to represent an MP3 song. Second, we will analyze of the variation works of J.S. Bach and their characteristics in MP3 features.

## 7. REFERENCES


[1] ISO/IEC 11172-3:1993, "Information Technology — Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s — Part 3: Audio."

[2] Cooper, M., and J. Foote, "Automatic Music Summarization via Similarity Analysis," *in Proceedings of 3rd International Conference on Music Information Retrieval*, 2002.

[3] Hirata, K., and S. Matsuda, "Interactive Music Summarization based on GTTM," *in Proceedings of 3rd International Conference on Music Information Retrieval*, 2002.

[4] Hsu, Jia-Lien, Chih-Chin Liu, A.L.P. Chen, "Discovering Nontrivial Repeating Patterns in Music Data," *IEEE Transactions on Multimedia*, Vol.3, Issue 3, pp.311-325, Sept. 2001.

[5] Liu, Chih-Chin and Po-Jun Tsai, "Content-based Retrieval of MP3 Music Objects," *in Proceedings of the 10th ACM International Conference on Information and Knowledge Management*, 2001.

[6] Liu, Chih-Chin and Chuan-Sung Huang, "A Singer Identification Technique for Content-Based Classification of MP3 Music Objects," *in Proceedings of the 11th ACM International Conference on Information and Knowledge Management*, 2002.

[7] Logan, Beth and Stephen Chu, "Music summarization using key phrases," *in Processings IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2000,Vol.2,pp. 749-752, 2000.

[8] Peeters, G., A. L. Burthe, and X. Rodet, "Toward Automatic Music Audio Summary Generation from Signal Analysis," in Proceedings of 3rd International Conference on Music Information Retrieval, 2002.

[9] Xu, Changsheng, Yongwei Zhu, and Qi Tian, "Automatic Music Summarization based on Temporal, Spectral and Cepstral Features," in *Proceedings of IEEE International Conference on Multimedia and Expo*, 2002.