Feature Analysis and Normalization Approach for Robust Content-Based Music Retrieval to Encoded Audio with Different Bit Rates

Shuhei Hamawaki[1], Shintaro Funasawa[1], Jiro Katto[1], Hiromi Ishizaki[2], Keiichiro Hoashi[2], and Yasuhiro Takishima[2]

[1] Waseda University, 3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555, Japan
{hamawaki,shint,katto}@katto.comm.waseda.ac.jp
[2] KDDI R&D Laboratories Inc, 2-1-15 Ohara, Fujimino-shi, Saitama 356-8502, Japan
{ishizaki,hoashi,takisima}@kddilabs.jp

Abstract. In order to achieve highly accurate content-based music information retrieval (MIR), it is necessary to compensate the various bit rates of encoded songs which are stored in the music collection, since the bit rate differences are expected to apply a negative effect to content-based MIR results. In this paper, we examine how the bit rate differences affect MIR results, propose methods to normalize MFCC features extracted from encoded files with various bit rates, and show their effects to stabilize MIR results.

Keywords: Mel-Frequency Cepstral Coefficient (MFCC), Content-based MIR Normalization.

1 Introduction

The recent development of various audio encoding formats such as MP3 (MPEG-1 Audio Layer-3), WMA (Windows Media Audio), and AAC (Advanced Audio Coding) have enabled efficient compression of music files with high sound quality. This technology has made possible the development of large-scaled online music distribution services. Furthermore, it has also become popular for customers of such services to share their personal "playlists," *i.e.*, lists of their favorite songs, on the Web. Such developments may lead to the realization of the "celestial jukebox," an application which accumulates all existing music in the world, and makes them accessible to application users.

Obviously, content-based music information retrieval (MIR) is an essential technology to make such an application usable. Therefore, many research efforts have been presented in this area. However, when considering a music collection accumulated for the celestial jukebox, it is clear that the collection consists of songs (audio files) encoded in various formats and/or bit rates. Therefore, content-based MIR must compensate the divergence of the features that are to be extracted from such songs. To the best of our knowledge, this problem has not been seriously considered in existing MIR research, since evaluations of such research are mainly conducted on data sets

B. Huet et al. (Eds.): MMM 2009, LNCS 5371, pp. 298–309, 2009.

individually constructed by the researchers, thus do not contain songs of various formats.

The objective of this research is two-fold. Mainly focusing on MFCC, a representative acoustic feature utilized in many existing work in the content-based MIR research area, we will examine influence of diverse audio file formats to MFCC feature extraction, and prove that the distortion of audio due to encoding cannot be ignored to develop an effective content-based MIR system. Secondly, we propose and evaluate MFCC normalization methods to compensate for the differences of MFCC features, which aim to reduce the effects of diverse bit rates to content-based MIR results.

2 Use of MFCCs in Music Retrieval

Mel-Frequency Cepstral Coefficients (MFCC) are acoustic features which are known to represent perceptually relevant parts of the auditory spectrum. Therefore, MFCC has been commonly used for speech recognition systems [7]. Furthermore, MFCCs have also been increasingly utilized in the field of content-based music analysis, such as genre classification, and audio similarity measures [8].

Spevak *et al.* [4] performs pattern matching on the sequences of MFCC to select a specific passage within an audio file. Deshpande *et al*. [5] convert MFCC features to a gray-scale picture, and use image classification methods to categorize audio files. MARSYAS [6] is a popular software framework for audio analysis, which uses MFCC as one of the features extracted from music for genre categorization, etc.

Furthermore, Sigurdsson *et al*. [1] have analyzed the robustness of MFCCs extracted from MP3 encoded files, and have concluded that MFCCs are sufficiently robust features, which can be utilized for content-based MIR. However, they have not provided any analysis about the effect of extracted MFCCs to content-based MIR results, which is the focus of this paper.

Generally, as the research efforts of the above conventional works indicate, MFCCs have generally been utilized as features, which express the timbral characteristics of music. While other aspects of music, *e.g.*, rhythm, harmony, and melody, are also essential to develop effective MIR systems, MFCCs can be assumed as a representative feature for content-based MIR.

In the following sections, we first investigate the influence of MFCC features extracted from differently encoded audio files, and show that the influence is not neglectable for content-based MIR. We also propose methods to compensate this problem, and reduce influence to MIR results that are caused by encoding distortion.

3 Analysis of MFCCs of Variously Encoded Music

3.1 Influence on MFCC Values

First, we examine the variance of MFCCs extracted from MP3 files encoded in different bit rates, and compared them with MFCCs extracted from raw audio files. All MFCC values are calculated with window size 25ms, window interval 10 ms and 13-dimension (12-dimention+power). LAME 3.97 is used for encoding and decoding, and the Hidden Markov Model toolkit [9] is used for calculating MFCC. For each

MFCC dimension, we compared the MFCC values extracted from raw audio files (hereafter referred as *Raw_MFCC*) with MFCC values extracted from MP3-encoded files (hereafter referred as *MP3_MFCC*), with bit rates of 128kbps (44.1kHz) and 64 kbps (24kHz).

MFCC values of each dimension are extracted from the same portion of the raw and encoded files. Figure 1 illustrates the MFCC values of the 1st, 6th, and 12th dimensions, extracted from two Japanese pop songs (SONG_A: male artist, SONG_B: female artist). It is clear from this Figure that the values of MFCC extracted from raw audio and MP3 files are different from each other. For example, let us focus on the MFCC value differences of the 1st dimension (shown on the first two graphs of Figure 1). For SONG_A, the MFCC values extracted from the 128kbps MP3 file are generally higher than that of the raw audio file, while the MFCC values of the 96kbps MP3 file are lower. However, the general distribution of MFCC values is different for SONG_B. Namely, the MFCC values of the 96kbps MP3 file are closer to those of the raw audio file, which is clearly different from the above observations of the MFCC values extracted from SONG_A.

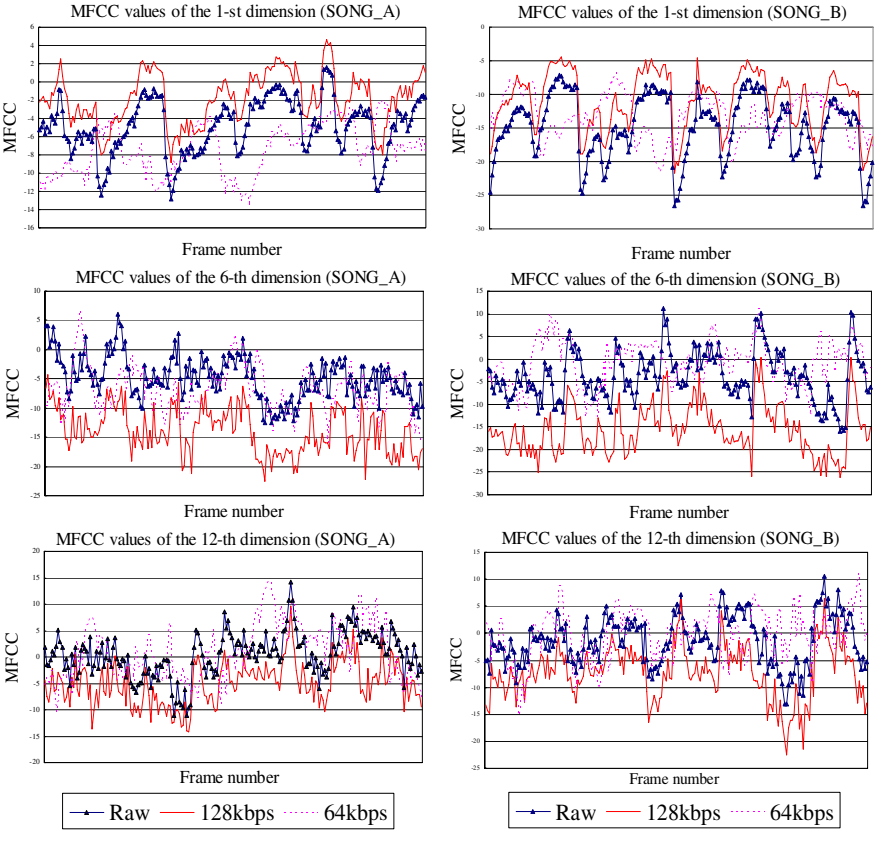

Fig. 1. Comparison of MFCC values extracted from MP3 files with various bit rates

Overall, the results of Figure 1 indicate that, even if the target song is the same, the MFCC values extracted from variously encoded files are different. Furthermore, it is also clear from the results of Figure 1, that the difference of MFCC values is dependent not only to the bit rate of MP3 encoding, but also to the acoustic features of the target song. These differences are expected to apply a significant impact to content-based MIR systems which utilize MFCC-based features, especially for music collections which consist of songs in various formats.

3.2 Influence on MIR

Next, we examine the effects of the difference of MFCC features to content-based MIR, based on an experimental content-based MIR system, and a music collection consisting of MP3 files with various bit rates.

For this experiment, we have developed a prototype MIR system, based on the MIR method proposed by Hoashi *et al*. [2], which utilizes the tree-based vector quantization (TreeQ) algorithm proposed by Foote [3]. TreeQ constructs the feature space to vectorize music, based on training data, *i.e.*, music with category labels. The training audio waveform is processed into MFCCs, and TreeQ recursively divides the vector space into bins each of which corresponds to a leaf of the tree. Once the quantization tree has been constructed, it can be used to vectorize input music data. We used the songs and sub-genre information of the RWC Genre Database [10] as the initial training data. Then, the method of Hoashi *et al.* is applied to automatically derive the training data set from the music collection for TreeQ, based on the results of clustering songs. This method enables the extraction of features that optimally express the characteristics of songs in any given music collection.

The music collection of our prototype MIR system consists of songs with various bit rates. Namely, the music collection consists of 2513 MP3 files of Japanese and Korean pop songs, whose bit rates range from 96kbps to 192kbps. Details of distribution of bit rate for that dataset are 96kbps (708files), 128kbps (589files), 160kbps (1195files), and 192kbps (21files).

First, we analyze the distribution of the vectors of songs with various bit rates, by plotting all song vectors on a two dimensional feature space. Namely, the dimensions of the song vectors are reduced to two, by selecting the first two elements of principal component analysis conducted on the vectors of the songs in the music collection.

Figure 2 shows the distribution of all vectors in our music collection on the two-dimensional feature space. From Figure 2, it is clear that, the vectors of 96 kbps songs are densely located in a small area of the feature space, whereas the song vectors of songs with higher bit rates are scattered evenly, regardless of the actual acoustic features of the songs.

This result indicates that song vectors used for content-based MIR are severely affected by the bit rates of the songs in the collection. An example of a problem which may occur as a result of this result, is a situation where a user submits a query song, which happens to be encoded with low bit rate. While songs which are perceptually similar to the query song may exist in the music collection, such songs may not be successfully retrieved by the MIR system, simply because the bit rates of the songs in the collection differ to that of the query song.

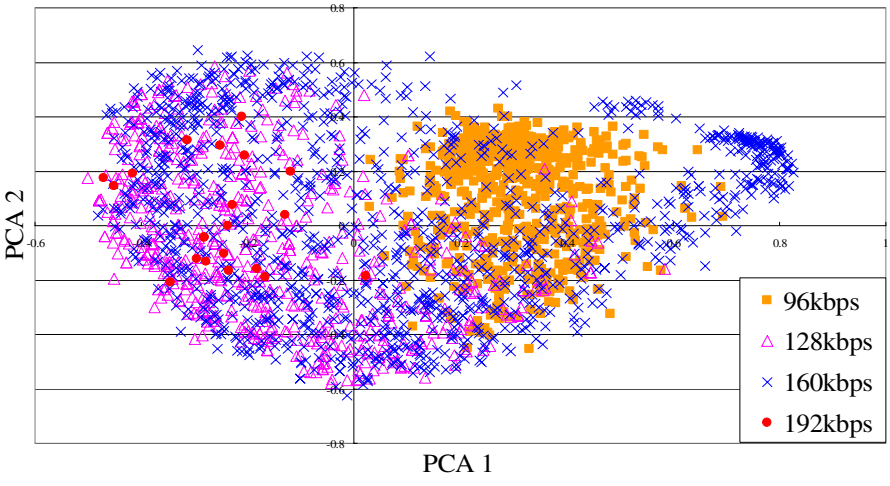

Fig. 2. Distribution of song vectors in 2D feature space

3.2.1 Experimental Data

Next, we examine how the divergence of song vectors extracted from songs with various bit rates will affect content-based MIR results. For the following experiment, we have prepared a music collection which consists of 96 Japanese pop songs. From this music collection, we generated four sets of music audio files: Raw (wav files with no compression), and MP3 files encoded in 192kbps (44.1kHz), 128kbps (44.1kHz), and 64kbps (24kHz).

3.2.2 Experiment Method

From the previous four sets of music audio files, we have constructed content-based MIR systems, following the method proposed in [2]. Namely, the feature space for each set (hereafter referred as: *Raw_hist*, *192_hist*, *128_hist*, and *64_hist*, respective to the format/bit rate of the music collection) is generated by the method of [2], and all songs in each data collection are vectorized based on the corresponding feature space. Furthermore, in order to simulate a music collection composed of songs in various formats, we have also generated a "mixed" data collection of MP3 files, by randomly selecting the bit rate of each song evenly in the experimental data set. Vectors of all songs in the mixed collection are also generated in the same way. We will refer to this feature space as "*mix_hist*."

3.2.3 Evaluation Measures

In order to analyze the difference between MIR results for song collections with various formats, we select a song as the MIR query, and calculate the similarity between the selected query and all other songs in each collection. The MIR results of *Raw_hist* are utilized as a reference, to which the results of the other MIR systems are compared. Query-to-song similarity is calculated based on the cosine distance between the query and song vectors.

The difference between the MIR results of the raw data set and the encoded data sets ({*192, 128, 64, mix*}*_hist*) is measured by calculating the correlation coefficient between the MIR results, *i.e*., the list of songs and their similarity to the query. Correlation coefficient (*r*) between MIR results of *Raw_hist* (*R*) and other feature space (*H*), which consists of *N* (=95) cosine distance scores, are calculated by the following formula.

$$r = \frac{\sum_{K=1}^{N}\left(R_k - \overline{R}\right)\left(H_k - \overline{H}\right)}{\sqrt{\sum_{K=1}^{N}\left(R_k - \overline{R}\right)^2}\sqrt{\sum_{K=1}^{N}\left(H_k - \overline{H}\right)^2}} \tag{1}$$

where R_k denotes cosine distance between the query and the *k*-th song in *Raw_hist*, and $\overline{R}$ denotes the average cosine distance of all data from the query in *Raw_hist*. Similarly, H_k denotes the cosine distance between the query and the *k*-th song, and $\overline{H}$ denotes the average cosine distance for each system ({*192, 128, 64, mix*}*_hist*).

Moreover, as another criterion, we calculate the ratio of songs which appear in the top ten songs of the MIR results of both *Raw_hist*, and {*192*, *128*, *64*, *mix*}*_hist*. This ratio is hereafter referred to as *Coin* (which stands for "*coin*cidence"). In the following experiment, we calculate the correlation coefficient and *Coin* for each of the 96 songs, and use the average values for overall comparison of MIR results.

3.2.4 Result

Table 1 shows the average correlation coefficient, and *Coin* of the MIR results of *Raw_hist* and each MP3-based music collection. Furthermore, Figure 3 illustrates the differences between the MIR result of Raw_hist and the other MIR results. In Figure 3, all songs are sorted in descending order, based on their similarity to a specific query song in *Raw_hist*, and the query-to-song similarity of the other MIR systems are plotted according to the order of the sorted songs.

The results of Table 1 indicate that the difference of MIR results to *Raw_hist* increases along with the decrease of encoding bit rates. Another notable observation is the severe difference between the MIR results of *mix_hist* and *Raw_hist*, which can be

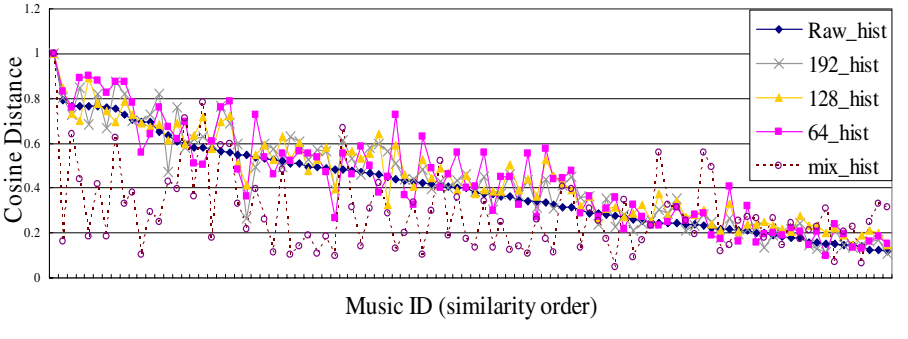

Fig. 3. Comparison of cosine distance of songs sorted by similarity in *Raw_hist*

Table 1. Average of correlation coefficient and *Coin* of MIR results

Feature space	Correlation	*Coin*
192_hist	0.900	6.28/10
128_hist	0.899	6.25/10
64_hist	0.850	5.25/10
mix_hist	0.219	2.23/10

observed from the low correlation coefficient and *Coin* of *mix_hist* in Table 1, and the scattered cosine distance of *mix_hist* results in Figure 3.

3.2.5 Discussions

As clear from the results of Table 1 and Figure 3, the difference of MFCC values extracted from audio files with different bit rates applies a significant impact to content-based MIR results. This is especially notable for music collections composed of a mixture of songs encoded with various bit rates, as can be observed from the results of *mix_hist*.

If the music database for a "*celestial jukebox*" is to be accumulated by collecting music data from various record companies and/or Web users, the resulting collection will consist of songs in various formats. The previous experimental results indicate that, existing content-based MIR methods, such as the method utilized in our previous experiments, must be able to handle the diversity of feature values that are expected to be extracted from songs with different formats.

The results in Table 1 show that, when the bit rate of songs in the music collection is fixed, the correlation to the original MIR results is high. Therefore, a naïve solution to resolve the difference of features extracted from mixed song collections is to unify the bit rate of all songs, prior to feature extraction. This bit rate unification can be conducted by adjusting the bit rate of all songs to the lowest bit rate of all songs in the music collection. By this method, the feature space is expected to be closer to the original feature space, than directly utilizing mixed features. However, if the minimum bit rate of songs in the music collection is extremely low, the amount of information to be lost in the bit rate adjustment process will be huge, especially for songs with high bit rates. Moreover, if a music file whose bit rate quality is lower than the minimum of the music files in the database is added to the collection, the feature space must be re-constructed.

In order to solve this problem, we propose methods to normalize MFCC values. Such methods are expected to resolve the variety of MFCCs, while avoiding the risky process to unify bit rate quality. Details of this method are described in the following chapter.

4 MFCC Normalization

We examine three normalization techniques, Cepstral Mean Normalization, Cepstral Variance Normalization, and Mean and Variance Normalization, in order to compensate the difference of MFCC values extracted from mixed song collections. All of the three

methods aim to make MIR results close to the original feature space, where MFCC features are extracted from raw audio files. These normalization methods are used to reduce the influence of different environmental conditions in the field of speech recognition [11]. In the following methods, the mean and variance of MFCC values for each MFCC dimension are calculated for each song, based on all MFCCs extracted from the song in question. Details of the three methods are as follows.

4.1 Normalization Method

CMN: Cepstral Mean Normalization

CMN normalizes Cepstral vector by subtracting the average Cepstral vector from the original vector. This method can be expressed in the following formula.

$$\hat{C}(i) = C(i) - \mu(i) \tag{2}$$

where $\hat{C}(i)$ denotes the i-th dimensional Cepstram after normalization, $C(i)$ denotes the i-th dimensional Cepstram before normalization, and $\mu(i)$ denotes the average of i-th-dimensional Cepstral vector.

CVN: Cepstral Variance Normalization

CVN normalizes Cepstral vector by dividing the original Cepstral vector by the standard deviation.

$$\hat{C}(i) = \frac{C(i)}{\sigma(i)} \tag{3}$$

$\sigma(i)$ is the i-th dimensional Cepstral standard deviation .

MVN: Mean and Variance Normalization

MVN normalizes Cepstral vector by Cepstral average and standard deviation.

$$\hat{C}(i) = \frac{C(i) - \mu(i)}{\sigma(i)} \tag{4}$$

4.2 Result of Normalization

We first compare the *MP3_MFCC* (128kbps, 64kbps) and *Raw_MFCC* (Raw) values after normalization. Figure 4 shows the result of MVN normalization of the MFCC values of the same songs and MFCC dimensions presented in Figure 1.

From comparison of Figures 1 and 4, it is clear that the normalized MFCC values extracted from MP3 files (64kbps, 128kbps) have moved closer to the original MFCC (Raw) value, especially for MFCC values of 128kbps MP3 files are more overlapped with those of Raw in each graph. Similar results are also observed for other songs and normalization methods. This analysis proves that MFCC normalization is effective to reduce the difference of MFCC values extracted from variously encoded music files.

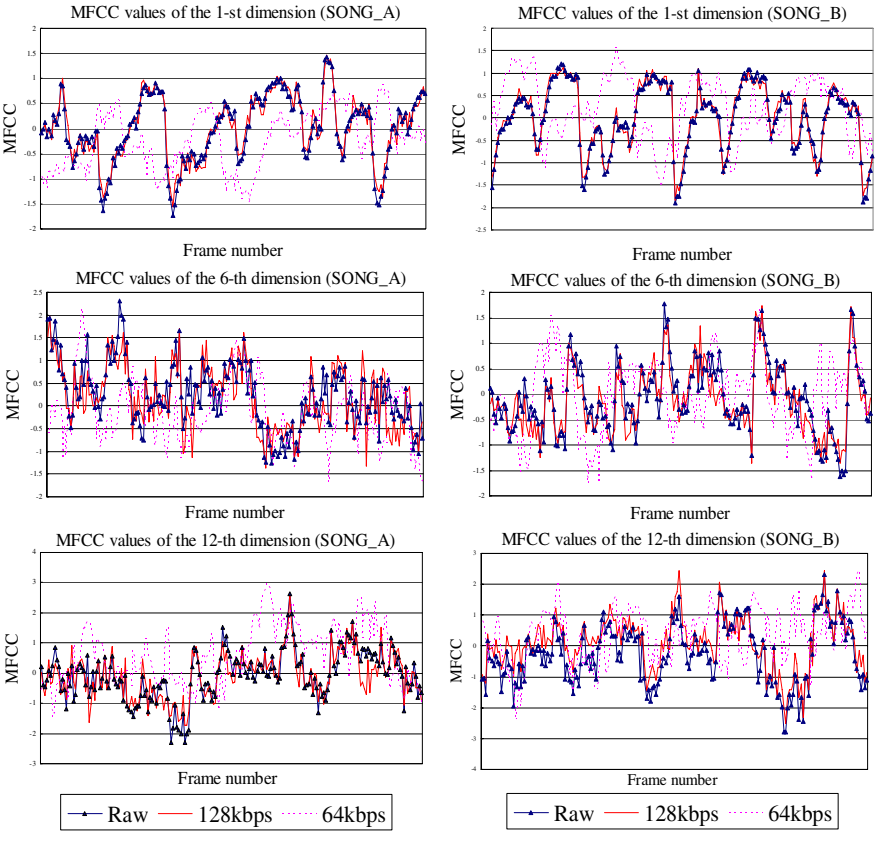

Fig. 4. Comparison of normalized MFCC values

Next, in order to examine the effects of MFCC normalization to the vectorization of music, we generated song vectors from MFCC values extracted from music files of the same database used in Figure 2 by MVN, and plotted all vectors on a two-dimensional graph generated by principal component analysis. This result is illustrated in Figure 5.

As obvious from the comparison of Figures 2 and 5, the biased distribution of vectors extracted from music files with different bit rates has been generally resolved by MVN normalization. Similar results are also observed for the other MFCC normalization methods. These results indicate that MFCC normalization is also effective to reduce the difference of MFCC-based song vectors extracted from audio files with variant bit rates.

Finally, in order to analyze how MFCC normalization affects content-based MIR results, we have conducted the same experiment as in Section 3.2.2, using the vectors generated from normalized MFCCs extracted from the mixed music collection. The results are compared with the results with those of *mix_hist* presented in Table 1 and Figure 3.

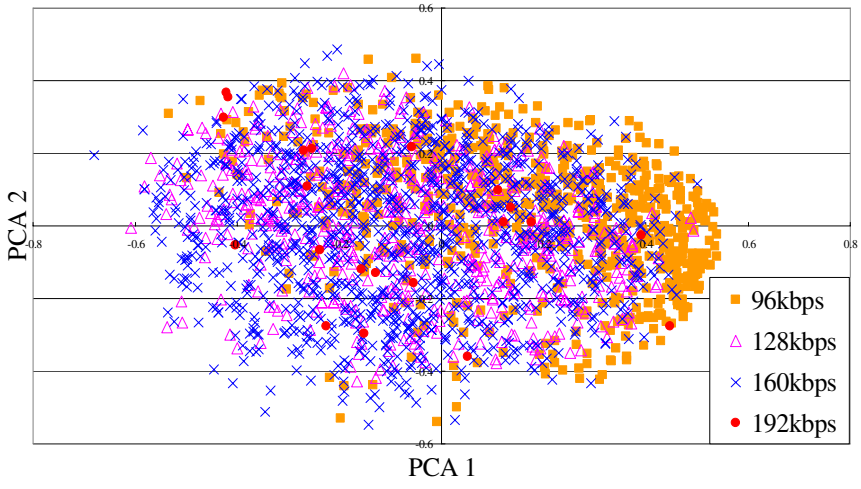

Fig. 5. Distribution of song vectors after normalization (MVN)

Table 2 shows the average correlation coefficient and *Coin* for each normalization method. Figure 6 shows cosine distance between a specific query and other songs, for *Raw_hist* and *mix_hist*, where MFCC values are normalized by MVN.

Table 2. Average correlation coefficient and Coin of MIR results of *mix_hist* with MFCC normalization

Normalization method	Correlation	*Coin*
CMN	0.714	4.04/10
CVN	0.439	2.77/10
MVN	0.859	4.52/10

It is clear from Table 2 that, all normalization methods have led to higher correlation coefficients and *Coin*, compared to the results of *mix_hist* in Table 1. This result indicates that MFCC normalization is effective to reduce the difference of MIR results for mixed music collections. Of the three proposed MFCC normalization methods, MVN has achieved the highest correlation coefficient and *Coin*. Similar conclusions can also be derived from Figure 6, where the difference between the raw and mixed MIR results has been significantly reduced, compared to the results of *mix_hist* presented in Figure 3.

Overall, the above experimental results indicate that, MFCC normalization is effective to resolve MIR result differences caused by MP3 files encoded with various bit rates, thus should be implemented for content-based MIR systems with music collections which consist of variously encoded music files.

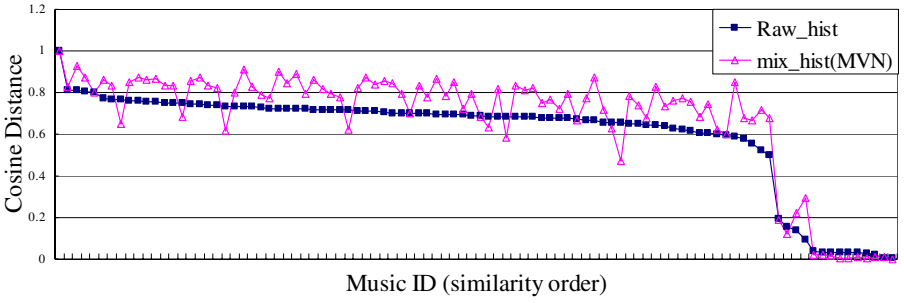

Fig. 6. Comparison of cosine distance of sorted songs after MFCC normalization

5 Conclusion and Future Work

In this paper, we have analyzed the difference of features extracted from music files with various bit rates, and their effect to content-based MIR results. Results of our analysis clearly show that, if the music collection consists of songs whose bit rates are mixed, the MIR results are significantly different from the results of music collections which consist of raw audio files. Furthermore, we confirmed that normalizing MFCC is effective to reduce the difference between MIR results for mixed music collections.

For the next step, we are investigating influences on other spectrum features like Flatness, Centroid, Rolloff and etc by encoding. We would also like to explore more optimal compensation methods, and conduct user-based evaluations of the MIR algorithms.

References

1. Sigurdsson, S., Petersen, K.B., Lehn-Schiøler, T.: Mel Frequency Cepstral Coefficients: An Evaluation of Robustness of MP3 Encoded Music. In: Proceedings of the International Conference on Music Information Retrieval (2006)
2. Hoashi, K., Matsumoto, K., Sugaya, F., Ishizaki, H., Katto, J.: Feature space modification for content-based music retrieval based on user preferences. In: Proceedings of ICASSP, pp. 517–520 (2006)
3. Foote, J.: Content-based retrieval of music and audio. In: Proceedings of SPIE, vol. 3229, pp. 138–147 (1997)
4. Spevak, C., Favreau, E.: SOUNDSPOTTER-A prototype system for content-based audio retrieval. In: Proceedings of the International Conference on Digital Audio Effects, pp. 27–32 (2002)
5. Deshpande, H., Singh, R., Nam, U.: Classification of musical signals in the visual domain. In: Proceedings of the International Conference on Digital Audio Effects (2001)
6. Tzanetakis, G., Cook, P.: MARSYAS: A framework for audio analysis. Organized Sound 4(3), 169–175 (2000)

7. Mermelstein, P.: Distance measures for speech recognition. Psychological and instrumental. Pattern Recognition and Artificial Intelligence, 374–388 (1976)
8. Logan, B.: Mel frequency cepstral coefficients for music modeling. In: Proceedings of the International Symposium on Music Information Retrieval (2000)
9. Slaney, M.: Auditory toolbox, version 2. Technical Report #1998-010, Interval Research Corporation (1998)
10. Goto, M., et al.: RWC Music Database: Music GenreDatabase and Musical Instrument Sound Database. In: Proceedings of the International Conference on Music Information Retrieval, pp. 229–230 (2003)
11. Viikki, O., Laurila, k.: Cepstral domain segmental feature vector normalization for noise robust speech recognition. Speech Communication 25, 133–147 (1998)