Analytical Approach to MFCC Based Space-Saving Audio Fingerprinting System

Myo Thet Htun
University of Computer Studies, Yangon, Myanmar
myothethtun@ucsy.edu.mm

Abstract

Audio fingerprinting is a smart technology to identify music contents relevant to the query by comparing the content-based hash (fingerprint) of the query to known hashes in the fingerprint database. For a million-song library, size of the fingerprint database restricts speedy and correct music identification. In this paper, we present a space-saving audio fingerprinting system based on Mel Frequency Cepstral Coefficients and analyze in detail. For a number of cepstral coefficients with 12 as default, the system yields a 2712-bit fingerprint for a 3-sec audio clip, a significant reduction in storage compared to the 8192-bit fingerprint of well-known Philips Robust Hashing method. Experimental results also show that the more number of input value for cepstral coefficients, the larger the fingerprint size. The choice of Mel coefficients also affects the identification performance of the system: 8 to 12 coefficients gives the best similarity rates while preserving robustness of the fingerprints to signal distortion such as pitch shifting.

Keywords*: audio fingerprint, a million-song library, Mel Frequency Cepstral Coefficients, music identification, Philips Robust Hashing.*

1. Introduction

In this multimedia age, music is one of the most popular online information and billions of audio data are streaming through the content providers such as iTunes, Netflix, Pandora, and YouTube. Music information retrieval (MIR) has attracted much attention in the areas of automatic broadcast monitoring, music identification, and detection of unauthorized music sharing.

Audio fingerprinting is best known in the field of MIR for its ability to link unlabeled audio to its corresponding metadata. When a query audio clip comes, its fingerprint is first calculated and matched against those already stored in the audio fingerprints database. The most similar audio is the one with the highest match score. Audio fingerprinting systems have various advantages such as guaranteeing the correct identification even if the query clips suffer from some kind of distortion and regardless of the format. Efficient fingerprint matching algorithms can identify the distorted versions of a recording as the same audio content.

A variety of audio fingerprinting methods have been proposed in the literature based on different acoustic features. In 2000, Logan [1] proposed that the de-correlated Mel frequency cepstral coefficients (MFCCs) vectors were appropriate for both speech and music modeling. Allamanche et al. [2] proposed the spectral flatness measure (SFM) features for audio finger-printing even though the SFM features perfectly worked only under clean environments. In particular, Haitsma et al. [3] developed a well-known fingerprinting method, namely Philips Robust Hashing (PRH), in which each 11.6 ms frame was represented by a 32-bit sub-fingerprint calculated based on the energy band differences both in time and frequency domains. Afterwards, many researchers refined the PRH method in different ways. Wang [4] who works for Shazam proposed an algorithm by using energy peaks in a frame and forming spectral pair landmarks for fingerprint matching. Ke et al. [5] improved the performance of the audio fingerprinting scheme by utilizing AdaBoost computer vision technique although it needed relatively longer query clips. Park et al. [6] introduced alternatives to the frequency-temporal filtering combination. Yao et al. [7] also improved the scalability of big audio data by applying sampling and counting method and inverted index for audio sub-fingerprints.

Most of the former researches focused on the accuracy of music identification rather than the size of fingerprint database and retrieval speed. However, both of those aspects are increasingly important these days as the size of song libraries are tremendously growing day by day.

Thus in this paper, we present a space-saving audio fingerprinting system based on the MFCC, which can work with large-scale music libraries. In

order to find out the best implementation of the proposed method for various music genres, detailed analysis is also carried out regarding the effect of the choice of the number of cepstral coefficients (i) on the size of fingerprint output and (ii) for the most robust and accurate fingerprint matching.

The rest of the paper is organized as follows. Section 2 discusses the proposed fingerprinting method in detail. Section 3 presents the detailed analysis of the reliability and robustness of the proposed method by playing MFCC parameters. Finally, Section 4 concludes the research work.

2. MFCC-Based Space-Saving Fingerprinting System

We implemented MFCC feature extraction by using Matlab code namely "HTK MFCC MATLAB" [8] written by author Kamil Wojcicki. This function produces a matrix of values for each sample sound that is the number of MFCC by the number of frames. 13 MFCC coefficients are resulted for 227 frames of 3-sec audio clip. It provides to achieve 13x227 feature vectors by keeping the default number of cepstral coefficients as 12 using *mfcc* function in [8]. Resulting feature vectors are depending on the other parameters such as windowing size, frame shift duration, etc. To analyse the specific effects upon the changes of output size and robustness regarding extracted features for audio clips, we mainly focus to use different parameters of cepstral coefficients in the range of 8 to 16. After that, we compute coefficients rows columns difference to transform binary representation from feature vectors by way of PRH method [3].

Fig. 1 shows the framework of the proposed space-saving fingerprinting system. With the inspiration from the PRH method, fingerprint extraction in this system is done for windowed time intervals (i.e. frames). Unlike the PRH, the proposed system chooses the MFCC features over Fourier spectral information to compose a fingerprint. The reasoning behind is that the MFCC is based on the Mel-scale which is the human ear scale. Thus, it should be more appropriate for extracting a compact digital summary of a sound that can well approximate the human perception.

2.1. Pre-processing

Down sampling: Input (3-sec) audio is firstly down-sampled to a mono Pulse Code Modulation 16-bit audio stream with the sampling rate of 5512 Hz. This process compresses the signal so that more compact fingerprints can be achieved, e.g. it just retains only about one-eighth of the original samples for a 48 kHz sampled signal. This process also eliminates the effect of different playback speeds and thus improves the accuracy of the derived fingerprints.

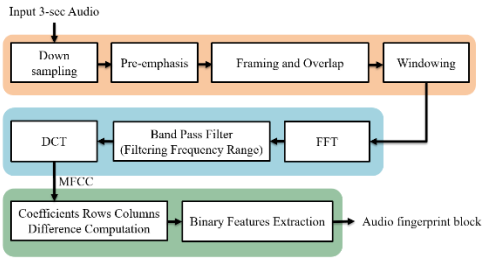

Figure 1. Framework of the MFCC-based space-saving fingerprinting system

Pre-emphasis: A pre-emphasis filter is then applied on the down-sampled signal to balance the frequency spectrum by boosting the signal energy in high frequencies.

$$y(t) = x(t) - \alpha x(t-1), \quad (1)$$

where the typical value for the filter coefficient α is usually between 0.9 and 1.0, and we set as 0.97 in our experiments.

Framing and overlap: The resulting signal is then split into short-time frames: 370 ms frames with 11.6 ms frame shift duration.

Windowing: In order to reduce discontinuities between frames or to smooth the first and last points in a frame, the Hanning window defined by Eq. 2 is applied on each frame.

$$w(n) = 0.5(1 - \cos 2\pi(n/N)), 0 \le n \le N-1, \quad (2)$$

where ***N*** is the window length.

2.2. MFCC Feature Extraction

Fast Fourier Transform (FFT): The FFT is then applied on each frame of the windowed signal to extract the spectral information. A good approximation of the frequency contours of the signal is obtained by concatenating adjacent frames.

Band pass filter: Frequency spectrum yielded by the FFT is then warped according to the Mel-scale in order to adapt the frequency resolution to the properties of the human ear. The spectrum is segmented into a number of critical bands ranging from 300Hz to 2kHz (the most relevant spectral range in the Human Auditory System) by means of a Mel filterbank which typically consists of overlapping triangular filters. Those filters capture the energy at each critical band and give a rough approximation of the spectrum shape. Mel scale for a given frequency ***f*** in HZ is computed by using Eq. 3. The mapping between the frequency in

Hz and Mel scale is linear below 1kHz and logarithmic above 1kHz.

$$F(mel) = 2595 * \log_{10}\left[1 + \frac{f}{700}\right]. \quad (3)$$

Discrete Cosine Transformation (DCT): DCT is then applied to the logarithm of the filterbank outputs to convert the log Mel spectrum into time domain. The result is a set of Mel coefficients, also called acoustic or feature vectors and its size depends on the number of cepstral coefficients. For ***n*** number of cepstral coefficients, this system generates the ***n*** x 227 MFCC feature vectors for a 3-sec audio clip.

2.3. Audio Fingerprint Extraction

For a compact fingerprint representation, the MFCC features are converted to a binary string by computing the sign differences between the features of the adjacent rows and columns of the MFCC feature vectors.

$$f = \begin{cases} 1, & (m(r,c) - m(r,c+1)) - \\ & (m(r-1,c) - m(r-1,c+1)) > 0, \\ 0, & \text{otherwise}, \end{cases} \quad (4)$$

where $\boldsymbol{m}(\boldsymbol{r},\boldsymbol{c})$ is the Mel coefficient of row ***r*** and column ***c*** of the feature vectors and ***f*** is the resulting fingerprint bit.

As an example, this process yields a 2712-bit (=12x226) fingerprint block for a 13x227 MFCC feature vectors of a 3-sec audio clip. It was a big reduction in storage compared to the fingerprint size of the PRH, where each 3-sec clip was represented by an 8192-bit fingerprint block [3].

A good fingerprinting system needs not only to be compact but also to provide accurate music identification. The following section presents the analysis results of the effect of the choice of the number of cepstral coefficients on the size of fingerprint output and for the most robust and accurate fingerprint matching.

3. Result and Discussion

Although music modeling of Logan [1] used MFCCs as audio features, there was no enough analysis over the choice of MFCC parameters. In this section, we analyze the proposed method regarding how the choice of input number of cepstral coefficients affect the size, reliability, and robustness of the resulting audio fingerprints.

3.1. Experimental Setup

Operating system: Microsoft Windows 10 Enterprise 64-bit

Processor: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz (8 CPUs)

Memory: 4096MB RAM

Research aided tools: MATLAB R2018a and Microsoft Visual Studio 2017 for simulating the proposed method and Audacity 2.3.0 for various audio editing functions such as pitch shifting, adding background noise, speed changes, etc.

3.2. Dataset

Eight short audio excerpts are used as testing data in our experimental results and which are selected by different popular music genres (Pop, Rock, Jazz, Classical, Hard Rock, Hip Hop, Acoustic, and Traditional). The songs used in the experiments are officially granted for research purpose by Myanmar Music Store (MMS) [9] which is the biggest music company in Myanmar. Table 1 lists the audio excerpts used in the experiments. The song code is used instead of the song title for protection of copyright.

Table 1. Audio clips for experiments

Audio Clips for Experiments			
No.	Song ID	Musical Genres	Duration (min:sec)
1	*S01649*	*Acoustic*	*4:16*
2	*S00172*	*Classical*	*2:39*
3	*S05031*	*Hard Rock*	*5:33*
4	*S58104*	*Hip Hop*	*4:21*
5	*S05868*	*Jazz*	*3:01*
6	*S13971*	*Pop*	*3:43*
7	*S00079*	*Rock*	*3:53*
8	*S00015*	*Traditional*	*4:48*

3.3. Analysis of the Choice of Cepstral Coefficients Number Regarding Fingerprint Size

As previously discussed, the proposed system yields MFCC feature vectors with different sizes depending on the number of cepstral coefficients. In this experiment, the size of output fingerprint is analyzed for the different number of cepstral coefficients in the range of 8 to 16. Table 2 clearly shows that more number of cepstral coefficients yields larger fingerprint size.

Table 2 also compares the proposed method and PRH from the space-saving point of view. It can be clearly seen that the proposed method is more space-saving than the PRH, which yields to speedy music retrieval and thus more appropriate for million-song libraries.

Even though the experimental results show that the less the cepstral coefficients number, the more

space-saving, robustness of the fingerprints to common signal distortions should also be considered. The following section analyzes how the choice of the cepstral coefficients number affects the fingerprint robustness.

Table 2. Number of cepstral coefficients vs. size of fingerprint

Fingerprint size (bit) for 3-sec audio			
Proposed Method			PRH Method
Input number of cepstral coefficients	Resulted MFCC feature vectors	Final output after rows columns difference computation	
8	*9 x 227*	*8 x 226 = 1808*	*8192*
9	*10 x 227*	*9 x 226 = 2034*	
10	*11 x 227*	*10 x 226 =2260*	
11	*12 x 227*	*11 x 226 = 2486*	
12	*13 x 227*	*12 x 226 = 2712*	
13	*14 x 227*	*13 x 226 = 2938*	
14	*15 x 227*	*14 x 226 = 3164*	
15	*16 x 227*	*15 x 226 = 3390*	
16	*17 x 227*	*16 x 226 = 3616*	

3.4 Analysis of the Choice of Cepstral Coefficients Number Regarding Fingerprint Robustness

In order to answer the next theoretical question of how robust these space-saving audio fingerprints are, resilient experiments for various signal degradations are carried out.

In this experiment, firstly we calculate Mel feature vectors for each audio clip in Table 1, by using cepstral coefficients number in the range of 8 to 16. After calculating coefficients difference computation as defined by Eq. 4, we get the final bit streams in binary representation form. Those final bit streams are stored as fingerprints in a database. Then, the audio clips are edited in Audacity to simulate signal distortions such as

- **Liner Speed Changes:** -4% to +4%,
- **Distortions:** Hard Clip, Soft Clip, Heavy Overdrive, Valve Overdrive, and Blues Drive Sustain,
- **Pitch Shifting:** -4% to +4%,
- **Noise Additions:** White Noise, Pink Noise and Brownian Noise, and
- **Signal Compression:** 128 kbps, 64 kbps, 32 kbps, 16 kbps and 8 kbps.

The edited audio clips are then assumed as query and their fingerprints are matched against those in the fingerprint database. We analyze which feature vector is the most robust to various signal distortions to find the best implementation of the proposed method. Robustness and reliability is determined by means of the bit error rate (BER), defined by Eq. 5. The BER is calculated by comparing the transmitted sequence of bits to the received bits and counting the number of errors. It is used to estimate the similarity between two audio clips.

$$BER = Number\ of\ errors\ /\ Number\ of\ bits. \quad (5)$$

If the BER between the query fingerprint block and one fingerprint segment stored in the database beforehand is lower than the threshold $\boldsymbol{T}$, it is considered to be a reliable match. A number of experiments have proved that when the BER is less than $\boldsymbol{T}$=0.35, matching results can be regarded as effective [3]. BER calculation upon the MFCC coefficients range 8-16 well preserved their similarity rates respectively. We present the final results of average BER after testing with following signal degradations using Audacity-

Firstly, robustness of the proposed method to 'linear speed changes' of the audio clips is evaluated by changing the speed of the audio clips from -4% to +4% in Audacity. Those speed changes affect both the tempo and pitch of the original songs. The edited audio clips are then assumed as query and their fingerprints are matched against those extracted from the original songs. The resulting BERs for the proposed method are shown in Table 3 and also visualized in Fig. 2. The proposed method is well robust against the speed changes under threshold value 0.35 for all musical genres. Especially, MFCCs value 8 and 10 give the most similarity rates than others.

The robustness of the proposed method to various kinds of signal distortions is also tested by editing the audio clips by adding the effects of Hard Clip, Soft Clip, Heavy Overdrive, Valve Overdrive, and Blues Drive Sustain. These distortions are implemented with the factory presets values of Audacity. The resulting BERs are shown in Table 4 and illustrated in Fig. 3. The results show that the proposed method preserves its robustness very well: all the BER values are under threshold. In this signal degradation, MFCCs value 13 shows higher reliability rates by two times.

Robustness of the proposed method to 'pitch shifting' is also shown in Fig. 4 and Table 5. The query clips are edited by shifting their pitch from -4% to +4%. It can be clearly seen from Fig. 4 that the proposed method perfectly preserves its robustness under threshold as well. For this experiment, MFCCs value 8 is the most sufficient value for all genres.

Robustness of the proposed method to 'signal compression' is also analyzed for various compression rates: 128 kbps to 8 kbps by using LAME MP3 encoder. The resulting BER values are shown in Table 6 and illustrated in Fig. 5. It can be seen that the

similarity rates of all musical genres to compression are robust under threshold. Moreover, MFCCs value 12 is the most acceptable for this experiment.

The robustness results of the proposed method to different noise effects are shown in Table 7 to Table 9 and illustrated in Fig. 6 to Fig. 8. For white and pink noise additions, the proposed method well preserves the robustness except Jazz and Rock. Among the noise types, the proposed method is more robust to the brownian noise rather than the white and pink noises. Among the different MFCCs coefficients, value 11 is the most robust for all musical genres in these noise types.

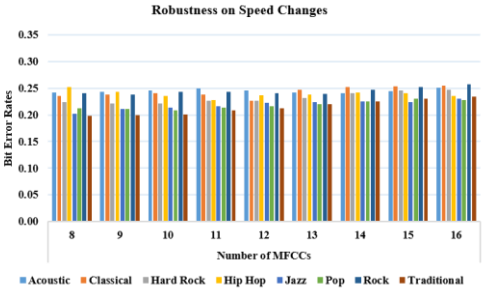

Figure 2. Robustness on speed changes

Table 3. Robustness on speed changes

Robustness on Speed Changes									
Musical Genres	Number of MFCCs								
	8	9	10	11	12	13	14	15	16
Acoustic	*0.2419*	*0.2439*	*0.2465*	*0.2493*	*0.2456*	*0.2415*	***0.2412***	*0.2440*	*0.2513*
Classical	*0.2357*	*0.2381*	*0.2404*	*0.2382*	***0.2272***	*0.2470*	*0.2519*	*0.2534*	*0.2544*
Hard Rock	*0.2246*	*0.2218*	***0.2218***	*0.2264*	*0.2272*	*0.2317*	*0.2404*	*0.2464*	*0.2475*
Hip Hop	*0.2526*	*0.2436*	*0.2359*	***0.2279***	*0.2369*	*0.2381*	*0.2424*	*0.2407*	*0.2354*
Jazz	***0.2021***	*0.2112*	*0.2138*	*0.2158*	*0.2225*	*0.2234*	*0.2255*	*0.2235*	*0.2301*
Pop	*0.2118*	*0.2115*	***0.2084***	*0.2138*	*0.2167*	*0.2204*	*0.2253*	*0.2301*	*0.2277*
Rock	*0.2403*	***0.2384***	*0.2432*	*0.2429*	*0.241*	*0.2395*	*0.247*	*0.2521*	*0.2573*
Traditional	***0.1984***	*0.1993*	*0.2014*	*0.2086*	*0.2125*	*0.2202*	*0.2257*	*0.2310*	*0.2344*

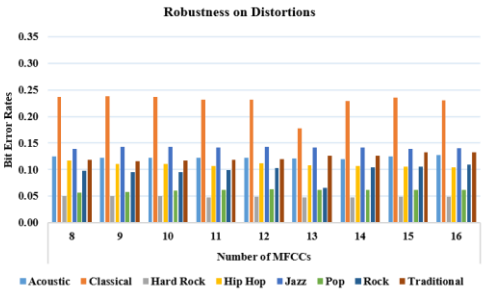

Figure 3. Robustness on distortions

Table 4. Robustness on distortions

Robustness on Distortions									
Musical Genres	Number of MFCCs								
	8	9	10	11	12	13	14	15	16
Acoustic	*0.1246*	*0.1224*	*0.1226*	*0.1228*	*0.1218*	*0.1207*	***0.1195***	*0.1245*	*0.1274*
Classical	*0.2372*	*0.2386*	*0.2363*	*0.2319*	*0.2316*	***0.1779***	*0.2296*	*0.2357*	*0.2308*
Hard Rock	*0.0503*	*0.0500*	*0.0498*	***0.0479***	*0.0488*	*0.0480*	*0.0479*	*0.0490*	*0.0488*
Hip Hop	*0.1170*	*0.1108*	*0.1104*	*0.1073*	*0.1116*	*0.1087*	*0.1064*	*0.1058*	***0.1046***
Jazz	*0.1389*	*0.1434*	*0.1426*	*0.1421*	*0.1435*	*0.1421*	*0.1413*	***0.1384***	*0.1397*
Pop	***0.0566***	*0.0577*	*0.0599*	*0.0619*	*0.0633*	*0.0615*	*0.0611*	*0.0619*	*0.0619*
Rock	*0.0979*	*0.0949*	*0.0958*	*0.0992*	*0.1035*	***0.0652***	*0.1047*	*0.1056*	*0.1093*
Traditional	*0.1181*	***0.1155***	*0.1176*	*0.1186*	*0.1192*	*0.1262*	*0.1263*	*0.1328*	*0.1328*

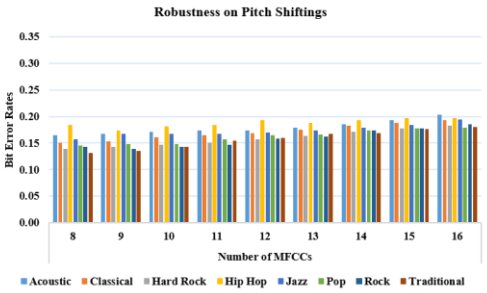

Figure 4. Robustness on pitch shiftings

Table 5. Robustness on pitch shiftings

Robustness on Pitch Shiftings									
Musical Genres	Number of MFCCs								
	8	9	10	11	12	13	14	15	16
Acoustic	***0.1649***	*0.1678*	*0.1713*	*0.1739*	*0.1738*	*0.1789*	*0.1853*	*0.1931*	*0.2034*
Classical	***0.1505***	*0.1527*	*0.1615*	*0.1644*	*0.1688*	*0.1747*	*0.1833*	*0.1884*	*0.1932*
Hard Rock	***0.1387***	*0.1427*	*0.1469*	*0.1504*	*0.1567*	*0.1638*	*0.1709*	*0.1778*	*0.1831*
Hip Hop	*0.1838*	***0.1732***	*0.1812*	*0.1846*	*0.1926*	*0.1885*	*0.1927*	*0.1967*	*0.1969*
Jazz	***0.157***	*0.167*	*0.1677*	*0.167*	*0.1705*	*0.1742*	*0.1785*	*0.1841*	*0.1942*
Pop	***0.1453***	*0.1475*	*0.1483*	*0.1572*	*0.1644*	*0.166*	*0.1732*	*0.1776*	*0.179*
Rock	*0.1427*	***0.1389***	*0.1435*	*0.1462*	*0.1586*	*0.162*	*0.1732*	*0.1779*	*0.1848*
Traditional	***0.1317***	*0.1347*	*0.1432*	*0.1541*	*0.1593*	*0.1672*	*0.1686*	*0.1768*	*0.1808*

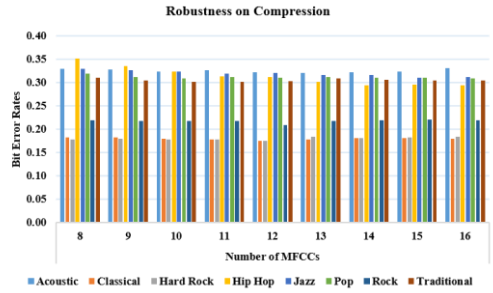

Figure 5. Robustness on MP3 compressions

Table 6. Robustness on MP3 compressions

Robustness on Compression									
Musical Genres	Number of MFCCs								
	8	9	10	11	12	13	14	15	16
Acoustic	*0.329*	*0.3282*	*0.3243*	*0.326*	*0.3216*	***0.3205***	*0.3219*	*0.3238*	*0.3309*
Classical	*0.1822*	*0.182*	*0.18*	*0.1777*	***0.1757***	*0.1776*	*0.1807*	*0.1815*	*0.1801*
Hard Rock	*0.1777*	*0.1798*	*0.1783*	*0.1784*	***0.1757***	*0.1837*	*0.1814*	*0.1829*	*0.1837*
Hip Hop	*0.3516*	*0.3359*	*0.3235*	*0.3131*	*0.3114*	*0.3023*	***0.2936***	*0.2954*	*0.2947*
Jazz	*0.3301*	*0.3268*	*0.323*	*0.3192*	*0.3202*	*0.3169*	*0.3158*	***0.3098***	*0.3119*
Pop	*0.3196*	*0.3125*	***0.3087***	*0.3125*	*0.3105*	*0.3119*	*0.3103*	*0.31*	*0.3096*
Rock	*0.2197*	*0.2175*	*0.2183*	*0.2182*	***0.2095***	*0.2183*	*0.2196*	*0.2207*	*0.2198*
Traditional	*0.3102*	*0.3045*	*0.3023*	***0.3016***	*0.303*	*0.3086*	*0.3055*	*0.304*	*0.304*

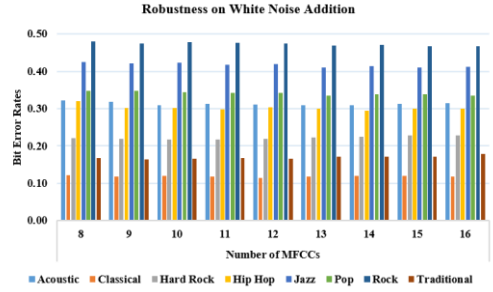

Figure 6. Robustness on white noise addition

Table 7. Robustness on white noise addition

Robustness on White Noise Addition									
Musical Genres	Number of MFCCs								
	8	9	10	11	12	13	14	15	16
Acoustic	*0.3214*	*0.3186*	***0.3084***	*0.3118*	*0.3111*	*0.3094*	*0.3094*	*0.3123*	*0.3149*
Classical	*0.1208*	*0.1184*	*0.1202*	*0.1172*	***0.1149***	*0.1183*	*0.1197*	*0.1198*	*0.1170*
Hard Rock	*0.2202*	*0.2190*	***0.2167***	*0.2171*	*0.2183*	*0.2226*	*0.2252*	*0.2289*	*0.2279*
Hip Hop	*0.3192*	*0.3011*	*0.3025*	*0.2978*	*0.3039*	*0.2990*	***0.2951***	*0.2997*	*0.2995*
Jazz	*0.4257*	*0.4220*	*0.4236*	*0.4184*	*0.4186*	*0.4110*	*0.4133*	***0.4101***	*0.4116*
Pop	*0.3480*	*0.3467*	*0.3445*	*0.3424*	*0.3422*	*0.3353*	*0.3377*	*0.3376*	***0.3348***
Rock	*0.4806*	*0.4754*	*0.4780*	*0.4759*	*0.4753*	*0.4696*	*0.4702*	***0.4667***	*0.4679*
Traditional	*0.1680*	***0.1630***	*0.1648*	*0.1682*	*0.1662*	*0.1708*	*0.1706*	*0.1706*	*0.1787*

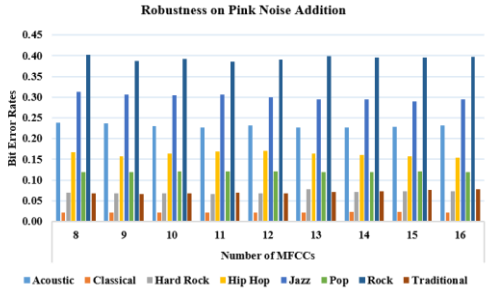

Figure 7. Robustness on pink noise addition

Table 8. Robustness on pink noise addition

Robustness on Pink Noise Addition									
Musical Genres	Number of MFCCs								
	8	9	10	11	12	13	14	15	16
Acoustic	*0.239*	*0.236*	*0.2296*	*0.2276*	*0.2312*	*0.2269*	***0.2262***	*0.2287*	*0.2312*
Classical	***0.0208***	*0.0211*	*0.0217*	*0.0217*	*0.0217*	*0.0221*	*0.0226*	*0.0229*	*0.0222*
Hard Rock	*0.0687*	*0.0683*	*0.0674*	***0.0668***	*0.0678*	*0.0774*	*0.0706*	*0.072*	*0.0721*
Hip Hop	*0.1666*	*0.1573*	*0.1635*	*0.1681*	*0.1699*	*0.1638*	*0.1609*	*0.157*	***0.1537***
Jazz	*0.3124*	*0.3057*	*0.3047*	*0.3063*	*0.2996*	*0.2949*	*0.2944*	***0.2904***	*0.2951*
Pop	*0.1191*	*0.1195*	*0.1207*	*0.121*	*0.1216*	*0.1194*	***0.1187***	*0.1209*	*0.1189*
Rock	*0.4023*	*0.3871*	*0.3919*	***0.3849***	*0.3902*	*0.3991*	*0.3957*	*0.395*	*0.3974*
Traditional	*0.0685*	***0.0663***	*0.0683*	*0.0688*	*0.068*	*0.0711*	*0.0726*	*0.076*	*0.0773*

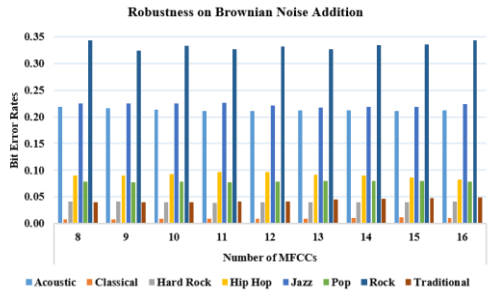

Figure 8. Robustness on brownian noise addition

Table 9. Robustness on brownian noise addition

Robustness on Brownian Noise Addition									
Musical Genres	Number of MFCCs								
	8	9	10	11	12	13	14	15	16
Acoustic	*0.2186*	*0.2157*	*0.2143*	***0.2105***	*0.2107*	*0.2129*	*0.2120*	*0.2117*	*0.2125*
Classical	*0.0082*	***0.0081***	*0.0092*	*0.0093*	*0.0091*	*0.0094*	*0.0106*	*0.0112*	*0.0108*
Hard Rock	*0.0413*	*0.0412*	*0.0402*	***0.0390***	*0.0396*	*0.0395*	*0.0396*	*0.0404*	*0.0411*
Hip Hop	*0.0904*	*0.0904*	*0.0929*	*0.0962*	*0.0970*	*0.0919*	*0.0899*	*0.0858*	***0.0828***
Jazz	*0.2252*	*0.2255*	*0.2259*	*0.2263*	*0.2214*	***0.2171***	*0.2186*	*0.2183*	*0.2234*
Pop	*0.0787*	*0.0775*	*0.0779*	***0.0775***	*0.0791*	*0.0792*	*0.0794*	*0.0793*	*0.0782*
Rock	*0.3439*	***0.3251***	*0.3330*	*0.3273*	*0.3326*	*0.3265*	*0.3352*	*0.3361*	*0.3443*
Traditional	*0.0403*	***0.0396***	*0.0400*	*0.0412*	*0.0414*	*0.0445*	*0.0458*	*0.0479*	*0.0487*

In summary, according to the experimental results discussed above, the reliability and robustness of the proposed method to common signal distortions is satisfactory in general, mostly keeping the BER levels under threshold. The proposed method especially performs for 'distortions'. And also well preserves for 'pitch shifting' distortion types and 'linear speed changes' which is the major challenge in broadcast monitoring systems and. For 'noise addition like brownian noise', the proposed method is highly robust than other noise types. The proposed method also well preserves its robustness against 'compression'.

This paper investigates for the acceptable reason of using MFCC coefficients in audio fingerprint extraction process. Based on the experimental research works, we can assume that the range of 8 to 12 cepstral coefficients gives the most similarity rates for various musical genres. Our proposed method brings effective ways of using MFCC coefficients for music identification area. On the other hand, we present the incontrovertible proof that the size of fingerprint block in our system is considerably reduced than PRH method.

Thus, it can be concluded that the proposed method can perfectly align the tradeoffs between space-saving and robustness of the audio fingerprints.

4. Conclusion

Audio fingerprinting can be used to quickly retrieve perceptual similar songs from a song database. For million-song libraries, not only the correct music identification but also the speedy retrieval rate is also very important. With the aim of achieving speedy music retrieval, the proposed method modifies the Philips Robust Hashing method to reduce its storage requirement for fingerprint database. The experimental results clearly showed that the proposed method can reduce the fingerprint size to one-third of the fingerprint yielded by the PRH. Additional to reducing the fingerprint size, the proposed method is also significantly robust against common signal distortions in different MFCC values for all popular musical genres. Thus, the proposed method can be utilized in broadcast monitoring systems and noisy environment. In addition, it can balance the trade-off between robustness and memory requirements of the fingerprints for large-scale music libraries.

References

[1] B. Logan, "Mel frequency cepstral coefficients for music modeling," International Symposium for Music Information Retrieval, Plymouth, USA, October 2000.

[2] E. Allamanche, J. Herre, O. Hellmuth, B. Froba, T. Kastner, and M. Cremer, "Content-based identification of audio material using mpeg-7 low level description," 2nd International Symposium on Music Information Retrieval, Indiana University, Bloomington, Indiana, USA, October 15-17, 2001.
[3] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," International Symposium for Music Information Retrieval, 2002.
[4] A. Li-Chun Wang, "An industrial strength audio search algorithm," International Symposium for Music Information Retrieval, 2003.
[5] Y. Ke, D. Hoiem, and R. Sukthankar, "Computer vision for music identification," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.
[6] M. Park, H. Kim, and S. H. Yang, "Frequency-temporal filtering for a robust audio fingerprinting scheme in real-noise environments," Electronics and Telecommunications Research Institute Journal, Volume: 28, Number: 4, Page: 509–512, 2006.
[7] S. Yao, B. Niu, and J. Liu, "A sampling and counting method for big audio retrieval," IEEE Second International Conference on Multimedia Big Data, 2016.
[8]https://www.mathworks.com/matlabcentral/fileexchange/32849-htk-mfcc-matlab
[9] http://myanmarmusicstore.com