ISBN 978-981-14-1455-8
Proceedings of 2019 the 9th International Workshop on Computer Science and Engineering
WCSE_2019_SPRING
Yangon, Myanmar, February 27-March 1, 2019, pp. 48-57
doi:10.18178/wcse.2019.03.009

Compact and Robust Audio Fingerprinting for Speedy Music Identification

Myo Thet Htun [1] [+] and Twe Ta Oo [1]

[1] Faculty of Computer Systems and Technologies, University of Computer Studies, Yangon, Myanmar

Abstract. An audio fingerprint, which is a compact content-based digest of an audio signal, is widely used to quickly locate perceptually similar songs in an audio database. For a million-song library, memory imposes a restriction for speedy and correct music identification and thus demands a compact fingerprinting system. This paper focuses on reducing memory requirement of fingerprint storage while preserving the robustness of fingerprints to common distortions such as compression, noise addition, etc. In this system, a 3-sec audio clip is represented by a 2712-bit fingerprint block. It significantly reduced the storage when compared with the Philips Robust Hashing (PRH), one of the dominant researches of audio fingerprinting, where a 3-sec audio clip was represented by an 8192-bit fingerprint block. Experimental results also showed that the reliability and robustness of the proposed fingerprinting system outperforms the PRH under various distortions, especially linear speed changes and pitch shifting.

Keywords: audio fingerprint, a million-song library, music identification, Philips Robust Hashing.

1. Introduction

Music is one of the most popular types of online information these days and billions of audio data are streaming through the content providers such as iTunes, Netflix, Pandora, and YouTube. These trends have posed a major challenge for searching, retrieving, and organizing music contents for million-song libraries. Music information retrieval (MIR) has gained its popularity in this multimedia age and its practical services include music identification, automatic broadcast monitoring, and detection of unauthorized music sharing.

Audio fingerprinting, a compact content-based signature of an audio recording, is best known for its ability to link unlabeled audio to its corresponding metadata (e.g. artist and song name), regardless of the audio format. It is a smart technology to identify the relevant contents correctly from a small piece of query music, which is generally kept only 3~5 seconds duration. Even though the query may have various types of noise and distortion, the underlying source signal is identical to the matching segment of the database. Efficient fingerprints and matching algorithms can identify the distorted versions of a recording as the same audio content. Audience measurement, broadcast monitoring, naming the tune, metadata collection, and finding duplicates are the well-known applications of audio fingerprinting technology.

A wide variety of audio fingerprinting methods have been proposed in the literature based on different acoustic features. In 2000, Logan [1] proposed Mel frequency cepstral coefficients (MFCCs) based method for music modelling. The author demonstrated that de-correlated MFCC vectors were appropriate for both speech and music spectra. Allamanche et al. [2] proposed a new methodology for audio fingerprinting – spectral flatness. As per experimental results, spectral flatness measure (SFM) features only perfectly worked under clean environments. Haitsma et al. [3] developed a well-known fingerprint extraction 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. Wang [4] who works for Shazam also proposed an algorithm by using energy peaks in a frame and forming spectral pair

[+] Corresponding author. Tel.: +95 9 448019015; fax: +95 1 610633.
E-mail address: myothethtun@ucsy.edu.mm.

landmarks. The local maxima within a defined section were grouped into pairs and nine spectral peaks were considered as a match score. Ke et al. [5] improved the performance of a fingerprinting scheme by utilizing the AdaBoost computer vision technique although it needed relatively longer query clips. Park et al. [6] introduced alternatives to the frequency-temporal filtering combination. Their method achieved robustness to background noise in a real situation, but there was no synergy of the filtering combination anywhere. Yao et al. [7] improved the scalability of big audio data by applying sampling and counting method and inverted index for audio sub-fingerprints. Although the method increased computational complexity for sampling and counting, their audio retrieval time was desirable.

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. In this paper, we modify the PRH method of Haitsma et al. [3] with the aim of generating a more compact fingerprint database for speedy music retrieval with acceptable accuracy.

The rest of the paper is organized as follows. Section 2 describes the literature review in which we focus the PRH method as our mainly cited literature. Section 3 discusses the space-saving architecture of the proposed method in detail. Section 4 presents the comparative analysis of the reliability and robustness of the proposed method and the PRH. Finally, Section 5 concludes the proposed research work.

2. Literature Review

The PRH method [3], whose overall scheme is shown in Fig. 1, is one of the most influential works on audio fingerprinting. In that method, fingerprint extraction is done for windowed time intervals (i.e. frames); thus, an input audio is segmented into frames, each with a length of approximately 0.4 seconds. The frames are then weighted by a Hanning window to smooth signal discontinuity with an overlap factor of 31/32. Then, Fourier transform is computed on every frame and only the absolute value of the spectrum is retained as many important audio features live in the frequency domain and the Human Auditory System (HAS) is relatively insensitive to phase as well. Then, in order to get a 32-bit sub-fingerprint for each frame, 33 non-overlapping and logarithmically spaced frequency bands are segmented from 300Hz to 2kHz (the most perceptible range by the HAS). Energy in each frequency band is then computed and a 32-bit hash string, i.e. sub-fingerprint, is obtained by computing the sign of the energy differences (simultaneously along the time and frequency axes) as defined by Eq. 1.

$$H(n,m)=\begin{cases}1, & \big(EB(n,m)-EB(n,m+1)\big)-\big(EB(n-1,m)-EB(n-1,m+1)\big)>0,\\ 0, & otherwise,\end{cases} \tag{1}$$

where $EB(n,m)$ is the energy of band m of frame n and $H(n,m)$ is the m-th hash bit of the frame n. A single 32-bit sub-fingerprint does not contain enough information to match the original audio. Thus, a fingerprint block is composed by combining all 256 sub-fingerprints for a 3-sec audio recording.

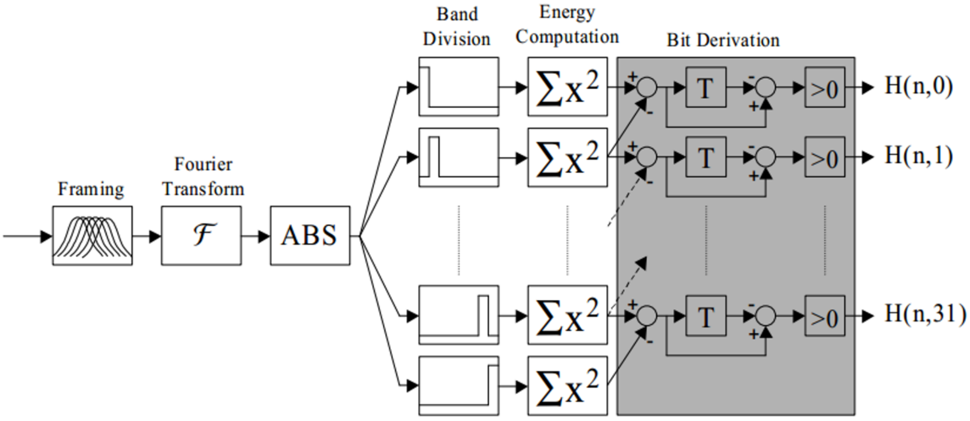

Fig. 1. Overview scheme of PRH fingerprint extraction

As the sizes of today song libraries are increasing, some flaws of the PRH method have already been pointed out in the literature.

- The first problem is the fingerprint block size of 8192 bits (=32 x 256) for a 3-sec audio clip. It needs huge amount of memory allocation.
- Another problem is the big index size of the 32-bit Lookup Table (LUT) which is used for matching process. The 2^{32} (=4G) entries in the LUT are too large to be resident in memory.
- The PRH also assumes that at least one of the 256 sub-fingerprints is error-free under 'mild' signal degradations. It ignores heavy signal degradation.
- Another problem of the PRH method is the 'single match principle' algorithm. It ignores the multiple occurrences of matching.

In this paper, we focus our attention on solving the first two problems of the PRH: reducing the size of the fingerprint block and the LUT. The proposed system chooses the MFCC features over Fourier transform 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. Details of the proposed method are explained in the following section.

3. Proposed Method

The comparative system flow of the proposed method and the PRH is shown in Fig. 2. As in the PRH, the proposed method extracts a sub-fingerprint block from each 11.6 ms frame. The main difference is that the proposed method considers the human ear scale-based Mel features as the fingerprint and whereas the PRH uses the FFT-based spectral information. The detailed framework of the proposed audio fingerprint extraction is shown in Fig. 3.

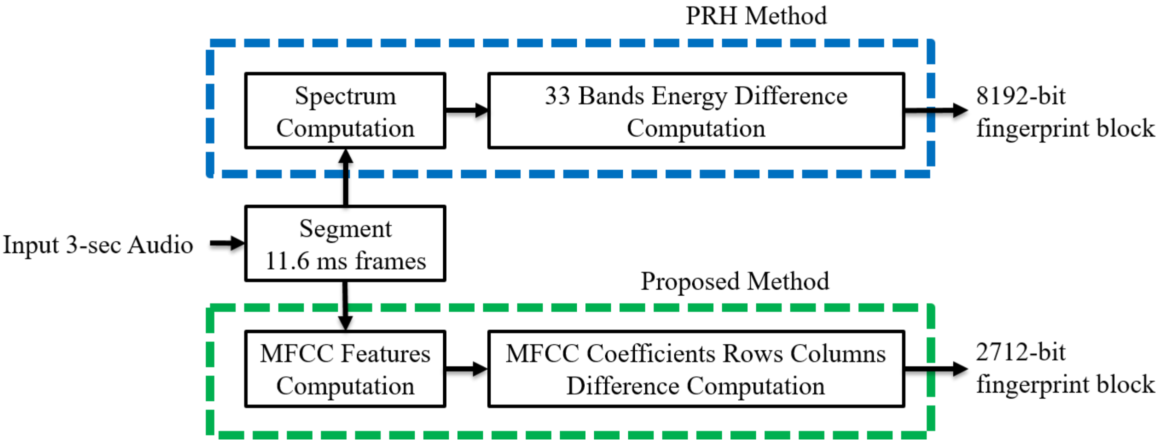
PRH Method
Spectrum Computation
33 Bands Energy Difference Computation
8192-bit fingerprint block
Input 3-sec Audio
Segment 11.6 ms frames
Proposed Method
MFCC Features Computation
MFCC Coefficients Rows Columns Difference Computation
2712-bit fingerprint block

Fig. 2. Comparative system flow for fingerprint extraction of the PRH and the proposed method

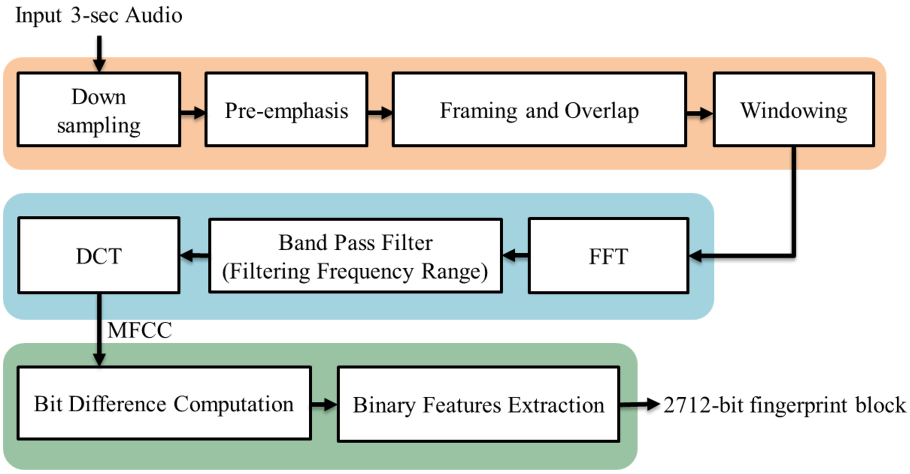
Input 3-sec Audio
Down sampling
Pre-emphasis
Framing and Overlap
Windowing
DCT
Band Pass Filter (Filtering Frequency Range)
FFT
MFCC
Bit Difference Computation
Binary Features Extraction
2712-bit fingerprint block

Fig. 3. Proposed framework for audio fingerprint extraction

3.1. Pre-processing

- **Down sampling:** Input audio is firstly down-sampled to a mono Pulse Code Modulation (PCM) 16-bit audio stream with the sampling rate of 5512 Hz. This process eliminates the effect of different playback speeds and thus improves the accuracy of the derived fingerprints. Moreover, this process also compresses the signal so that more compact fingerprints can be achieved, e.g. it just retains only about 1/8 of the original samples for a 48kHz sampled signal.
- **Pre-emphasis:** As defined by Eq.2, 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.

  $$y(t) = x(t) - \alpha x(t-1), \tag{2}$$

  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:** After pre-emphasis, the resulting signal is 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. 3 is applied on each frame.

  $$w(n) = 0.5(1 - \cos 2\pi(n/N)), 0 \le n \le N-1, \tag{3}$$

  where N is the window length.

3.2. 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.
- **Bandpass filter:** The 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 HAS) 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. 4. The mapping between the frequency in Hz and Mel scale is linear below 1kHz and logarithmic above 1kHz.

  $$F(mel) = 2595 * \log_{10}\left[1 + \frac{f}{700}\right]. \tag{4}$$

- **Discrete Cosine Transformation (DCT):** The 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 frequency cepstral coefficients that is called acoustic vectors. For a 3-sec audio excerpt, this system generates the 13x227 MFCC feature vectors. The size of the feature vectors depend on the frame size, frame shift duration, windowing method, and pre-emphasis values.

3.3. Audio Fingerprint Extraction

For a compact fingerprint representation, the MFCC features are converted to a binary representation as follows. With an inspiration from the bit derivation process of the PRH, sign differences between the MFCC features of the adjacent rows and columns of the 13x227 feature vectors are calculated. After this process, a 2712-bit (=12x226) fingerprint block is obtained for a 3-sec audio clip, and it can later be used for matching and identifying the query audio clips.

As stated by the PRH, these binary features have effectual advantages because they can be faster to compute, more efficient to compare, and more compact to store. Compared to the PRH, the proposed method reduces the 8192-bit fingerprint block for a 3-sec audio clip of the PRH to 2712-bit. By this way, memory requirement for fingerprint storage is much decreased and retrieval speed is increased. However, a good fingerprinting system needs not only to be compact but also to provide accurate music identification. The following section presents the reliability and robustness analysis of the proposed method.

4. Experiments

4.1. Research Aided Tools

- **Matlab R2018a:** Most of the experiments are simulated in Matlab.
- **Audacity 2.3.0:** Audacity is a free, open source, cross-platform software that supports a variety of audio editing functions. Audacity is used in this system to edit the audio clips by injecting common signal distortions such as adding background noise, pitch shifting, speed changes, etc.
- **Microsoft Visual Studio 2017:** Some parts of the proposed method such as re-sampling audios, converting to mono, framing, etc are implemented in C# by using Microsoft's famous IDE.

4.2. Runtime Environment

- **Operating System:** Microsoft Windows 10 Enterprise 64-bit
- **Processor:** Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz (8 CPUs)
- **Memory:** 4096MB RAM

4.3. Comparative Analysis of Fingerprint Size

Table 1 lists the four audio excerpts used in the experiments. Aforementioned in this paper, the proposed method extracts a 2712-bit fingerprint block for a 3-sec audio, whereas the PRH extracts an 8192-bit block. The last column of Table 1 states the size of the fingerprints yielded by the PRH and the proposed method in kilobytes. Averagely, the proposed method requires 27.4 kB fingerprint storage for a 4-min and 4-sec long audio clip, and whereas the PRH requires 82.6 kB. The proposed method saves approximately two-third of the storage space required by the PRH. Thus, it proves that the proposed method can provide speedy music retrieval and it is more appropriate for million-song libraries.

Table 1. Audio Clips

No.	Song Name	Artist	Duration (min:sec)	Fingerprint Size (kB)	
				Proposed	PRH
1.	A whole lot of Rosie	AC/DC	5:33	**36.7**	**111**
2.	O Fortuna	Carl Orff	2:39	**17.5**	**53**
3.	Say what you want	Texas	3:53	**25.7**	**77.7**
4.	Success has made a failure of our home	Sinead o'Connor	4:28	**29.6**	**89.3**
Average			**4:38**	**27.4**	**82.6**

4.4. Comparative Analysis of 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 and compared the results with the PRH. The robustness and reliability of the proposed fingerprinting system is evaluated 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.

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

If the BER between the query fingerprint block and one fingerprint segment stored in the database beforehand is lower than the threshold T, it is considered to be a reliable match. A number of experiments have proved that when the BER is less than T=0.35, matching results can be regarded as effective [3].

Firstly, robustness of the proposed method to 'linear speed changes' of the audio clips is evaluated by changing the speed of the audio clips in Table 1 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 PRH and the proposed method are shown in Table 2 and also visualized in Fig. 4. The proposed method is well robust against the speed changes from -2% to +2%, i.e. BERs are under threshold. Compared with the PRH, it is seen that the proposed method is getting more robust than the PRH when speed changes rates are higher.

The robustness of the proposed method to various kinds of signal distortions is also tested by editing the audio clips in Table 1 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 3 and illustrated in Fig. 5. The results show that the proposed method preserves its robustness very well: all the BER values are under threshold. However, it does not perform well as much as the PRH.

Robustness of the proposed method to 'pitch shifting' is also shown in Fig. 6 and Table 4. The query clips are edited by shifting their pitch from -4% to +4%. It can be clearly seen from Fig. 6 that the proposed method perfectly preserves its robustness under pitch shifting as well. As for the PRH, its robustness is getting decreased when the percentage of pitch shifting is severer.

The robustness results of the proposed method and the PRH to different noise effects are shown in Table 5 to Table 7 and illustrated in Fig. 7 to Fig. 9. For all kinds of noise types, the PRH outperforms the proposed method although both methods well preserve their robustness. Among the noise types, the proposed method is more robust to the pink and brownian noises rather than the white noise.

Robustness of the proposed method to 'signal compression' is also analyzed for various compression rates: 128 kbps to 8kbps by using LAME MP3 encoder. The resulting BER values are shown in Table 8 and illustrated in Fig. 10. It can be seen that the degrees of robustness of the PRH and the proposed method to compression are almost the same. When the compression rate is getting higher, their robustness is getting lower. Both methods can preserve their robustness to compression rate of up to 32kbps.

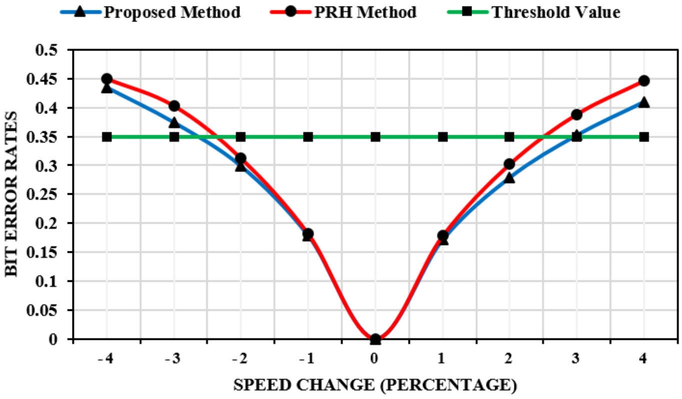

Fig. 4. Comparative test for linear speed changes

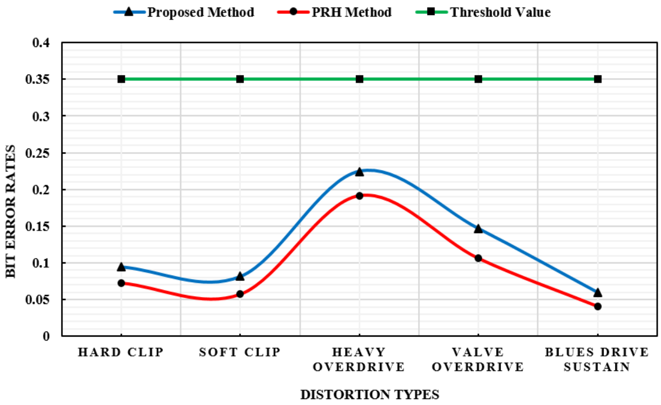

Fig. 5. Comparative test for distortion types

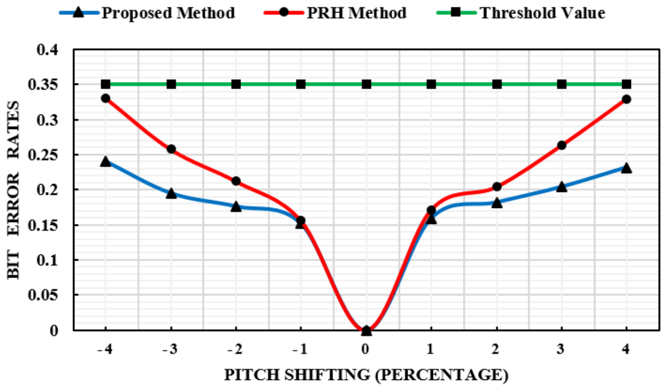

Fig. 6. Comparative test for pitch shiftings

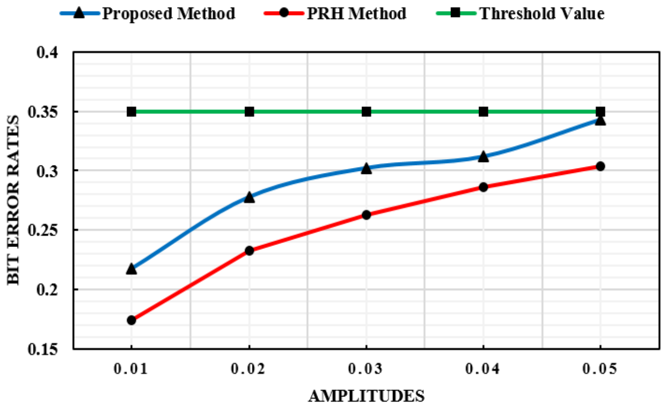

Fig. 7. Comparative test for white noise addition

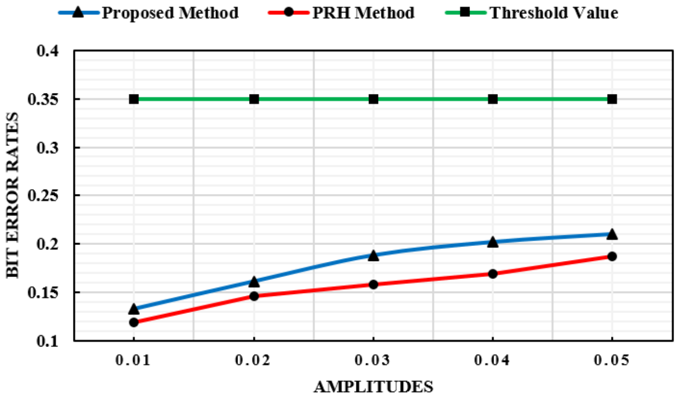

Fig. 8. Comparative test for pink noise addition

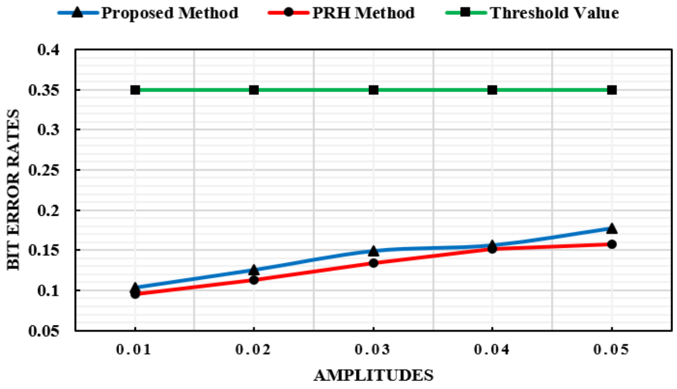

Fig. 9. Comparative test for brownian noise addition

Table. 2. Comparative test for linear speed changes

No.	***Linear Speed Changes (percentage)***	**Bit Error Rates**	
		Proposed Method	***PRH Method***
1.	-4 %	0.4346	0.4499
2.	-3 %	0.3740	0.4029
3.	-2 %	0.2982	0.3120
4.	-1 %	0.1780	0.1812
5.	1 %	0.1710	0.1777
6.	2 %	0.2787	0.3021
7.	3 %	0.3522	0.3884
8.	4 %	0.4099	0.4463
Average		**0.2774**	**0.2956**

Table. 3. Comparative test for different distortion types

No.	***Distortion Types***	**Bit Error Rates**	
		Proposed Method	***PRH Method***
1.	Hard Clip	0.0946	0.0726
2.	Soft Clip	0.0816	0.0575
3.	Heavy Overdrive	0.2245	0.1912
4.	Valve Overdrive	0.1465	0.1059
5.	Blues Drive Sustain	0.0600	0.0409
Average		**0.1214**	**0.0936**

Table. 4. Comparative test for pitch shiftings

No.	***Pitch Shifting (percentage)***	Bit Error Rates	
		Proposed Method	***PRH Method***
1.	-4 %	0.2401	0.3304
2.	-3 %	0.1949	0.2573
3.	-2 %	0.1762	0.2119
4.	-1 %	0.1517	0.1558
5.	1 %	0.1594	0.1715
6.	2 %	0.1819	0.2040
7.	3 %	0.2039	0.2631
8.	4 %	0.2317	0.3296
Average		**0.1711**	**0.2137**

Table. 5. Comparative test for white noise addition

No.	***White Noise Level (amplitude)***	**Bit Error Rates**	
		Proposed Method	***PRH Method***
1.	0.01	0.2174	0.1738
2.	0.02	0.2777	0.2322
3.	0.03	0.3022	0.2630
4.	0.04	0.3119	0.2857
5.	0.05	0.3431	0.3039
Average		**0.2905**	**0.2517**

Table. 6. Comparative test for pink noise addition

No.	*Pink Noise Level (amplitude)*	Bit Error Rates	
		Proposed Method	*PRH Method*
1.	0.01	0.1324	0.1184
2.	0.02	0.1609	0.1453
3.	0.03	0.1881	0.1583
4.	0.04	0.2019	0.1686
5.	0.05	0.2101	0.1870
Average		**0.1787**	**0.1555**

Table. 7. Comparative test for brownian noise addition

No.	*Brownian Noise Level (amplitude)*	Bit Error Rates	
		Proposed Method	*PRH Method*
1.	0.01	0.1037	0.0956
2.	0.02	0.1254	0.1132
3.	0.03	0.1487	0.1338
4.	0.04	0.1557	0.1513
5.	0.05	0.1767	0.1578
Average		**0.1420**	**0.1303**

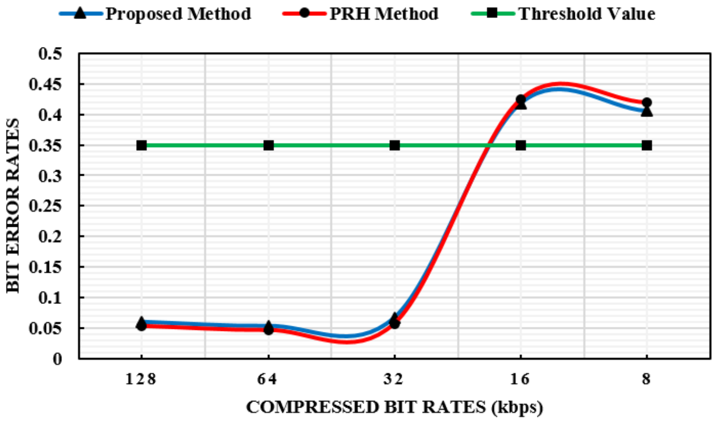

Fig. 10. Comparative test for MP3 compression

Table. 8. Comparative test for MP3 compression

No.	*Compressed Bit Rate (kbps)*	Bit Error Rates	
		Proposed Method	*PRH Method*
1.	128	0.0598	0.0529
2.	64	0.0532	0.0461
3.	32	0.0666	0.0567
4.	16	0.4186	0.4244
5.	8	0.4060	0.4196
Average		**0.2008**	**0.1999**

The robustness of the proposed method is also tested for different kinds of distortion types such as 'Hard Clip', 'Soft Clip', 'Heavy Overdrive', 'Valve Overdrive', and 'Blues Drive Sustain'. These distortion types are selected from factory presets

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 better than the PRH for 'linear speed changes' which is the major challenge in broadcast monitoring systems and 'pitch shifting' distortion types. For 'noise addition' and 'distortion types like hard clip', the PRH outperforms the proposed method. The proposed method also well preserves its robustness against 'compression'. Thus, it can be concluded that the proposed method can perfectly align the tradeoffs between space-saving and robustness of the audio fingerprints.

5. 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 comparably robust against common signal distortions as the PRH. 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.

6. 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.