DCT BASED MULTIPLE HASHING TECHNIQUE FOR ROBUST AUDIO FINGERPRINTING

Yu Liu, Kiho Cho, Hwan Sik Yun, Jong Won Shin, and Nam Soo Kim

School of Electrical Engineering and INMC
Seoul National University, Seoul 151-742, Korea
Email: {yliu1125, khcho, hsyun, jwshin}@hi.snu.ac.kr, nkim@snu.ac.kr

ABSTRACT

Audio fingerprinting techniques should successfully perform content-based audio identification even when the audio files are slightly or seriously distorted. In this paper, we present a novel audio fingerprinting technique based on combining fingerprint matching results for multiple hash tables in order to improve the robustness of hashing. Multiple hash tables are built based on the discrete cosine transform (DCT) which is applied to the time sequence of energies in each sub-band. Experimental results show that the recognition errors are significantly reduced compared with Philips Robust Hash (PRH) [1] under various distortions.

Index Terms— Audio Fingerprinting, Content-Based Audio Identification, Robust Hashing, Discrete Cosine Transform (DCT)

1. INTRODUCTION

An audio fingerprint is a compact content-based digest of an audio signal. It provides the ability to identify short, unlabeled audio excerpts and link them to the corresponding metadata (e.g. music name, album and artist). An ideal fingerprinting system should be able to recognize audio items regardless of the various distortions they may suffer from [2]. Also, it should be able to identify the excerpts only a few seconds long, and it should be computationally efficient since searching for the best match is performed for a huge database consisting of more than hundreds of millions of fingerprints [2]. Such properties of an audio fingerprinting system give rise to difficulties in both the fingerprint-extracting and matching phases, and many practical issues have been studied.

Among the various algorithms, Philips Robust Hash (PRH) [1] is considered one of the most famous content based audio identification techniques of which the performance is mathematically analyzed [3, 4]. With the assumption that at least one of the sub-fingerprints is invariant to noise, it is shown that efficient matching in the database is possible via a robust hashing algorithm. However, although this assumption makes sense in most mild conditions (e.g. MP3 compression, down-sampling, equalization), it fails in some noisy environments such as playing and recording, resulting in serious performance degradations.

This work was supported in part by the Brain Korea 21 Project and the Korea Science and Engineering Foundation (KOSEF) grant funded by Korea government (MEST) (No. R0A-2007-000-10022-0).

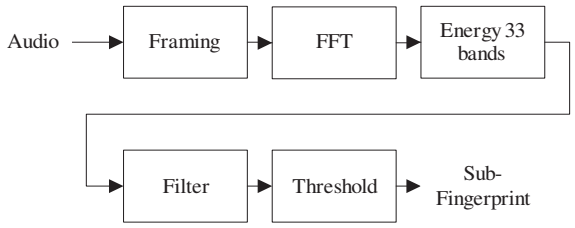

Fig. 1. Fingerprint extraction stage of PRH [6]

For such reasons, various improvements have been made to increase the robustness of PRH. There are several approaches which viewed extracting sub-fingerprints from the spectrogram as time-frequency domain 2-D filtering and tried to improve the robustness of sub-fingerprints by substituting the filters [5, 6, 7]. However, no one has tried to adopt multiple hash tables although it is considered to evidently enhance the robustness. On the other hand, in [8, 9, 10], the spectrogram was treated as a corrupted 2-D image and image processing based approaches were used.

In this paper, we propose a novel audio fingerprinting technique based on combining fingerprint matching results for multiple hash tables in order to improve the robustness of audio fingerprinting. Specifically, we apply discrete cosine transform (DCT) on the time sequence of energies in each sub-band, and build a hash table for each component of DCT. Experimental results showed that the proposed approach outperformed PRH under various environments.

2. SCHEME OF THE PRH ALGORITHM

Before getting into the proposed multiple hashing (MLH) technique, we will give a brief introduction to the PRH algorithm [1]. PRH consists of two phases: first is the fingerprint extraction stage, and second is the matching stage.

The overall block diagram for the fingerprint extraction stage is illustrated in Fig. 1. First, the audio signal is divided

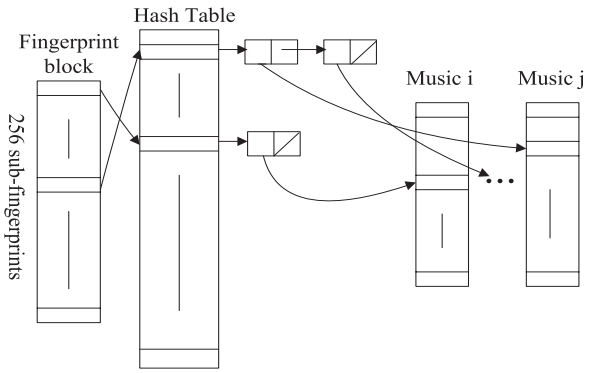

Fig. 2. Fingerprint matching stage of PRH

into overlapping frames with the length of about 370ms, and the frame shift is 1/32 of the frame length. Second, FFT is applied and power spectrum is obtained. Third, the energies for 33 non-overlapping logarithmically spaced sub-bands (e.g. Bark Scale) covering the frequency range of 300Hz to 2000Hz are calculated. These sub-band energies are then filtered by a time-frequency filter:

$$ED(n,m) = E(n,m) - E(n,m+1) - (E(n-1,m) - E(n-1,m+1)), \quad (1)$$

where $E(n,m)$ denotes the m-th sub-band energy of n-th frame, and $ED(n,m)$ is the output of the filter that represents the difference between energies from successive frames and neighboring frequency bands. Finally a 32-bit representation for each frame (which is referred to as a *sub-fingerprint*) is obtained by a thresholding process:

$$F(n) = [F(n,0), \cdots, F(n,31)], \quad (2)$$

$$F(n,m) = \begin{cases} 1, & ED(n,m) > 0 \\ 0, & ED(n,m) \leq 0, \end{cases} \quad (3)$$

where $F(n)$ is the sub-fingerprint of frame n and $F(n,m)$ is the m-th bit of it. The fingerprint consists of 256 consequent sub-fingerprints, which amounts to about 3 seconds.

The specific positions in the audios in the database which corresponds to each of the 256 sub-fingerprints are obtained using a hash table with sub-fingerprints as keys, as depicted in Fig. 2. Each entry of the hash table stores a list of pointers that point to the positions in the audio related to the sub-fingerprint. For a query audio, 256 sub-fingerprints are extracted and utilized as keys to the hash table to find the candidates which match at least one sub-fingerprint. The candidates are then evaluated through a comparing process: since we get a block of $256 \times 32 = 8192$ bits from the query signal, we also calculate the fingerprint of 8192 bits from the candidate position; the bit error rate (BER) between the two blocks is computed and compared with a threshold which was set to 0.35 in [1]. If the BER is less than that threshold the two signals are considered similar and the candidate audio is returned as the result.

3. DCT BASED MULTIPLE HASHING ALGORITHM

The matching algorithm of PRH relies on the assumption that there is at least one error-free sub-fingerprint from the query audio signal. Although the authors of [1] claim that for 'mild' degradations the assumption will always hold, that assumption may not work for seriously distorted audio signals, resulting in the significant degradation of the performance.

To enhance the robustness of the audio fingerprinting, we propose a multiple hashing algorithm. Multiple sets of sub-fingerprints are extracted using DCT coefficients of the time sequence of band energies in each band instead of band energies themselves. The fingerprint extraction stage of MLH system is illustrated in Fig. 3. The first three parts, i.e. framing, FFT and band energy calculation are the same as those in PRH. However, in contrast to PRH, L-point DCT is performed on L consecutive sub-band energies $E(n,m), E(n+1,m), \cdots, E(n+L-1,m)$. Among the output L coefficients, we only retain the first K values. Since just one frame is shifted for each DCT, we obtain K coefficients for the frame n and sub-band m by $C_k(n,m)$, $k = 1, 2, \cdots, K$. Then, they are filtered in much the same way as in the PRH:

$$ED_k(n,m) = C_k(n,m) - C_k(n,m+1) - (C_k(n-L,m) - C_k(n-L,m+1)). \quad (4)$$

Here $ED_k(n,m)$ represents the k-th filter output from sub-band m at frame n. Note that we use $C_k(n-L,m)$ and $C_k(n-L,m+1)$ instead of $C_k(n-1,m)$ and $C_k(n-1,m+1)$ to ensure that they are obtained based on band energies which do not overlap with those used to compute $C_k(n,m)$ and $C_k(n,m+1)$. In the final stage, the k-th sub-fingerprint at frame n, $F_k(n)$, is derived through the thresholding process:

$$F_k(n) = [F_k(n,0), \cdots, F_k(n,31)], \quad (5)$$

$$F_k(n,m) = \begin{cases} 1, & ED_k(n,m) > 0 \\ 0, & ED_k(n,m) \leq 0. \end{cases} \quad (6)$$

As a result, K sub-fingerprints are obtained for each frame, which are used as keys for the K distinct hash tables.

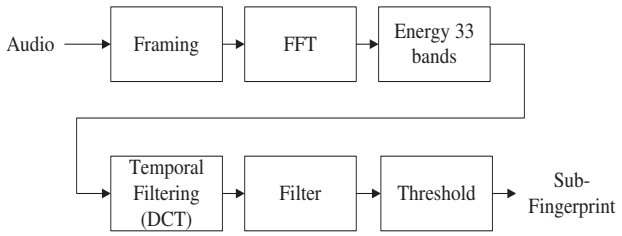

Fig. 3. Fingerprint extraction stage of MLH

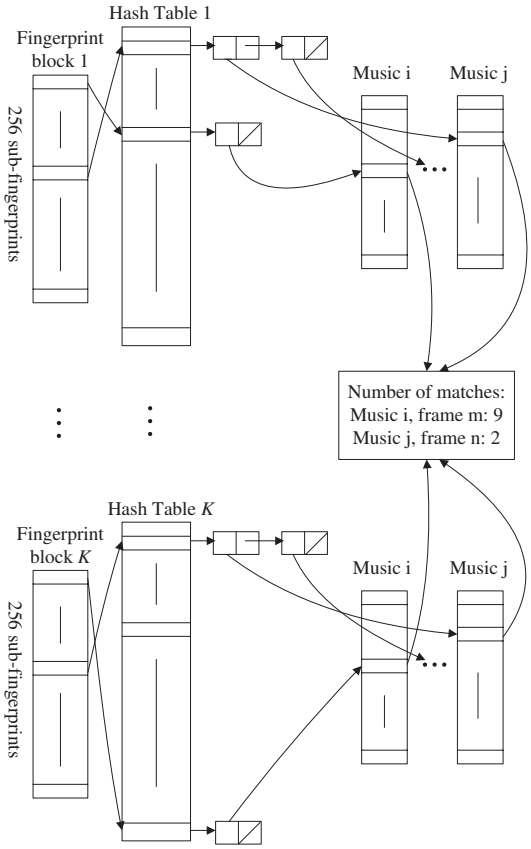

Fig. 4. Fingerprint matching stage of MLH

The fingerprint matching stage is illustrated in Fig. 4, K sub-fingerprints are extracted from the query audio for each frame. The set of the k-th sub-fingerprints for all 256 frames are represented as fingerprint block k in Fig. 4. For each hash table, the candidates are selected in the the same way as PRH. All the candidates chosen in K hash tables are sorted in the order of number of matches of which the maximum value is 256 times K. BERs from the query block for the candidates are computed in this order and the first candidate with BER less than specified threshold is returned.

In the proposed algorithm, DCT is adopted to construct separate hash tables since DCT has two desired properties. First, it has a strong energy compaction property [11], which means that most of the signal information tends to be concentrated in a few low-frequency components of DCT; second, among all the orthogonal transforms, the decorrelation performance of DCT is closest to the Karhunen-Loéve transform which is optimal in the decorrelation sense [12]. The energy compaction property together with the slow-varying nature of sub-band energies guarantees the information's concentration in only a few low-frequency coefficients, enabling the reduction of the number of hash tables. The decorrelation property ensures that the hash tables we build contain distinct information with each other so that the combination of the tables may improve the performances.

Table 1. Recognition rates (%) of PRH and MLH

Algorithm	Hash tables used	Set 1	Set 2
PRH	0	96.83	34.75
MLH	1	97.17	37.75
	2	94.83	30.50
	3	94.17	25.08
	4	96.25	29.67
	1,2	98.17	52.33
	1,3	98.25	50.67
	1,4	98.92	52.67
	2,3	96.83	42.17
	2,4	98.25	46.83
	3,4	98.00	43.17
	1,2,3	98.50	60.67
	1,2,4	99.17	62.92
	1,3,4	99.17	62.00
	2,3,4	98.67	55.25
	1,2,3,4	99.33	69.58

4. EXPERIMENTAL RESULTS

There are several parameters that should be determined for the implementation. The DCT length L was set to be 16, which considered to provide a good compromise between the frequency and time resolution. As for K, we used $K = 4$ since more than 90% of the total energy was found to concentrate on the first 4 coefficients from the experiment. The threshold for BER was set to be 0.35 as in [1]. Finally, to speed up the computation, we applied a running DCT algorithm [13] to further reduce the computation load since computation of DCT shifts one sample each time.

To test the performance of MLH, we conducted several experiments. The database used for constructing hash tables included 1500 music files with average length of 4 minutes, extracted from commercial compact discs to guarantee the high quality. The database consist of 3 groups of 500 files from classical, pop musics and rock/roll, respectively. To compare the performance, we also implemented the PRH algorithm [1]. As the query audios, 1200 music clips with the length of 256 frames which amounts to about 3 seconds were chosen from the database. To show the robustness of the algorithm, the following distortions were applied to construct the query sets:

Set 1: Playing and recording in a very quiet environment.

Set 2: Playing and recording in a noisy environment with recorded office noise played at the same time.

The hash tables used in MLH are denoted as HT1, HT2, HT3 and HT4, and the only table used in PRH is denoted as HT0. Here HTk was built from the k-th DCT coefficient; for example, HT1 was constructed from the DC components. For each query set, the MLH algorithms using various combinations of hash tables were tested along with PRH. The recognition rate (in percentage) for each algorithm is given in Table 1. Note that 'HT' of HTk is omitted in the table.

As can be inferred from the description on MLH, since HT1 of MLH was constructed using the DC component, the performance with HT1 only should be similar to PRH using HT0, although the low-pass effect of averaging when using HT1 may enhance the performance under severe noise conditions. When multiple hash tables were used, better results were achieved since the other hash tables provided additional information for the music clips. As can be seen from Table 1., the recognition rates from combinations of hash tables are significantly higher than those using only one of them. It is worth noting that if more hash tables are used, the memory usage and the computational burden increases. Thus it requires careful consideration how many hash tables are needed according to the environment in which the audio fingerprinting would be used.

5. CONCLUSIONS

In this paper, we present a novel audio fingerprinting technique based on multiple hash tables. Instead of using only one hash table as in [1], we rely on combinations of matching results for several hash tables which are built upon lower frequency DCT coefficients of sub-band energies. Experimental results have shown that the proposed MLH scheme outperformed the conventional PRH algorithm under various conditions. Future works may include the investigation of the efficient method for which reduced number of hash tables are used while maintaining the accuracy of MLH.

6. REFERENCES

[1] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *Proc. 3rd Int. Conf. Music Information Retrieval*, Oct. 2002, pp. 107–115.

[2] P. Cano, E. Batlle, T. Kalker, and J. Haitsma, "A review of audio fingerprinting," *Journal of VLSI Signal Processing*, vol. 41, no. 3, pp. 271–284, Nov. 2005.

[3] F. Balado, N.J. Hurley, E.P. McCarthy, and G.C.M. Silvestre, "Performance analysis of robust audio hashing," *IEEE trans. Information forensics and security*, vol. 2, no. 2, pp. 254–266, Jun. 2007.

[4] P.J.O. Doets and R.L. Lagendijk, "Distortion estimation in compressed music using only audio fingerprints," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 302–317, Feb. 2008.

[5] J.S. Seo, J. Haitsma, and T. Kalker, "Linear speed-change resilient audio fingerprinting," in *Proc. 1st Workshop on Model based Processing and Coding of Audio*, Nov. 2002, pp. 45–48.

[6] J. Haitsma and T. Kalker, "Speed-change resistant audio fingerprinting using auto-correlation," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, Apr. 2003, vol. 4, pp. 728–731.

[7] M.S. Park, H.R. Kim, and S.H. Yang, "Frequency-temporal filtering for a robust audio fingerprinting scheme in real-noise environments," *ETRI Journal*, vol. 28, no. 4, pp. 509–512, Aug. 2006.

[8] Y. Ke, D. Hoiem, and R. Sukthankar, "Computer vision for music identification," in *Proc. Computer Vision and Pattern Recognition*, 2005, vol. 1, pp. 597–604.

[9] S. Baluja and M. Covell, "Audio fingerprinting: Combining computer vision and data stream processing," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, Apr. 2007, vol. 2, pp. 213–216.

[10] S. Baluja and M. Covell, "Content fingerprinting using wavelets," in *Proc. Conf. Visual Media Production*, Nov. 2006, pp. 198–207.

[11] K.R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*, Academic Press, 1990.

[12] N. Ahmed, T. Natarajan, and K.R. Rao, "Discrete cosine transform," *IEEE Trans. Computers*, pp. 90–93, Jan. 1974.

[13] J.T. Xi and J.F. Chicharo, "Computing running DCTs and DSTs based on their second-order shift properties," *IEEE Trans. Circuits and Systems-I: Fundamental Theory and Applications*, vol. 47, no. 5, pp. 779–783, May 2000.