EFFICIENT MUSIC IDENTIFICATION BY UTILIZING SPACE-SAVING AUDIO FINGERPRINTING SYSTEM

Guang Yang
Peking University, Beijing, China
kobe@pku.edu.cn

ABSTRACT

Audio fingerprints can be used to implement an efficient music identification system on a million-song library, but the system requires huge amount of memory to hold the fingerprints and indexes. Therefore, for a large-scale music library, memory imposes a restriction on the speed of music identification. In this system, we propose an efficient music identification system which utilizes a kind of space-saving audio fingerprints. For saving space, original fingerprints are sub-sampled and only one quarter of the original data is reserved. In this way, memory requirement is much decreased and the search speed is significantly increased while the robustness and reliability are well preserved.

1. BASIC METHOD

1.1 Philips' audio fingerprinting scheme

Our system is based on Philips' audio fingerprinting [1]. An illustration of Philips' audio fingerprint extraction scheme is shown in Fig 1. An input audio is firstly down sampled to a mono audio stream with the sampling rate of 5kHz. Then the audio signal is segmented into frames every 11.6 milliseconds. The overlapping frames have a length of 0.37 seconds and are weighted by a Hanning window with the overlap factor of 31/32. Since the most important perceptual audio features live in the frequency domain, a spectral representation is computed by performing a Fourier transform on each frame. In order to get a 32-bit sub-fingerprint for each frame, 33 non-overlapping frequency bands are segmented from 300Hz to 2000Hz with a logarithmic spacing. Then the energy in every frequency band can be computed.

A sub-fingerprint represents the fingerprint extracted from a single audio frame, which is computed as follows. Let $E(n,m)$ denote the power of frequency band m of frame n and $F(n,m)$ denote the m-th bit in the sub-fingerprint of frame n. $F(n,m)$ is determined as:

$$F(n,m) = \begin{cases} 1, & \text{if } ED(n,m) > 0 \\ 0, & \text{otherwise} \end{cases}, \qquad (1)$$

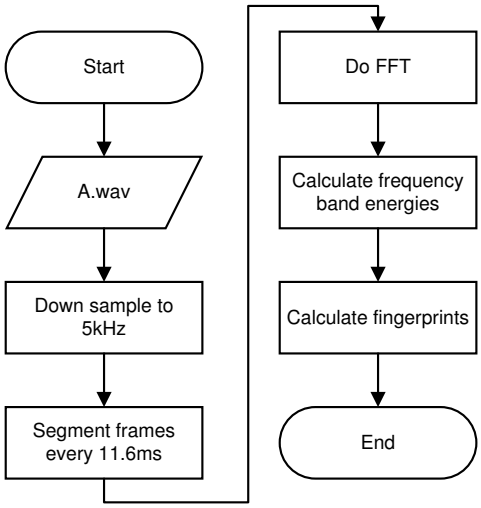

Figure 1. Audio fingerprint extraction process in [1]

where

$$ED(n,m) = E(n,m) - E(n,m+1) - (E(n-1,m) - E(n-1,m+1)). \qquad (2)$$

Experimental results prove that the energy differences between successive frequency bands are effective to identify music and robust to all kinds of distortion and corruption.

1.2 Original search algorithm

To avoid brute-force search in the database, an index-based pre-processing is conducted to enhance the search speed. A hash map is constructed to serve the search in the method of [1]. Every 32-bit sub-fingerprint is stored as an entry in the hash map and each entry points to a list of pointers to the positions in the fingerprint database where the respective 32-bit sub-fingerprints are located.

A single 32-bit sub-fingerprint does not contain enough information to match the original audio. Hence, a *fingerprint block* is used to compare the similarity between two fingerprint blocks. Let $F_Q(n,m)$ and $F_O(n,m)$ respectively denote the sub-fingerprints extracted from the query audio Q and the original audio O. The Hamming distance between each corresponding sub-fingerprint is calculated. Then, the bit error rate $BER(Q,O)$ between fingerprint

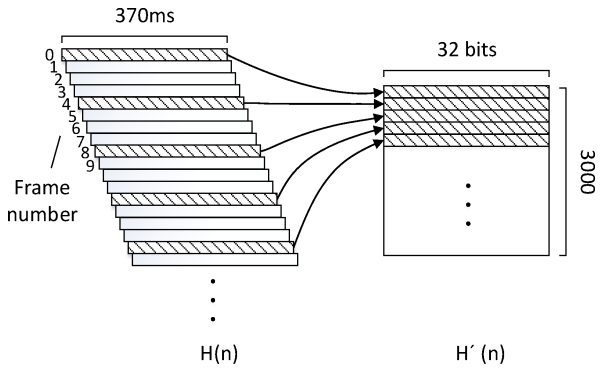

Figure 2. An illustration of sub-sampling scheme with $M = 4$

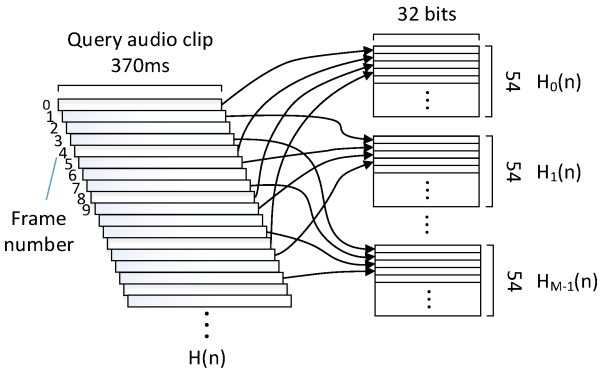

Figure 3. Interleaving process of a query audio clip with $M = 4$

blocks of length N is calculated as:

$$BER(Q,O) = \frac{\sum_{n=1}^{N}\sum_{m=1}^{32} F_Q(n,m) \oplus F_O(n,m)}{32N}, \quad (3)$$

where $\oplus$ denotes bitwise operation XOR (exclusive or). Then select the top 1 result with the smallest BER.

2. THE PROPOSED SCHEME

2.1 The improved extraction method

As mentioned above, a drawback of the prior-art fingerprinting system is the huge size of index. In the method [1] proposed by Haitsma *et al.*, the audio signal is segmented into frames of $0.37s$ long with an overlap factor of $31/32$. This results in a single frame every $11.6ms$ and then a 32-bit sub-fingerprint is extracted. For every index entry, it uses the 32-bit sub-fingerprint (4 bytes) as the key and a combination of song identification (4 bytes) and frame number (4 bytes) as the value. In this way, one index item will occupy 12 bytes. Therefore, a 5-minute song needs approximately 300 kb. If a database contains 1,000,000 songs, the index size is about 300GB, which will cost a huge amount of memory space for a real system.

2.2 Sub-sampling indexing scheme

Inspired by [2], we propose a sub-sampling indexing scheme. In the original audio fingerprint block, not every sub-fingerprint is indexed. The whole sequence $H(n)$ of sub-fingerprints is sub-sampled by a sub-sampler with a factor M. It produces a new sub-sequence $H'(n)$, which contains one out of every M sub-fingerprints of the original fingerprint block. The new sub-sequence $H'(n)$ is not only used to build index entries but also stored as a new fingerprint block for comparison in the next phase. An illustration of the sub-sampling scheme with $M = 4$ is shown in Fig. 2. Under this circumstance, a 5-minute song requires approximately $3000 \times 12bytes$ index memory capacity, which reduces that of the prior-art system without sub-sampling by 75%.

Now we describe the retrieval process. The query audio clip is processed by the same fingerprint extraction method as introduced above. This process extracts a full fingerprint block for each clip. For a 5-second clip, this operation yields a series of approximately 216 sub-fingerprints. The fingerprint block is applied to an interleaving process, which divides it into M interleaved sub-blocks as $H_0(n)$, $H_1(n)$, ... , $H_{M-1}(n)$, where M is the same integer as used in the sub-sampling process. Fig. 3 illustrates the interleaving process with $M = 4$. The M sub-blocks are successively applied to the database for retrieval. Then return the top 1 result with the smallest BER.

2.3 Further search

In the original search algorithm, it is assumed that there is at least one sub-fingerprint unchanged. But in a real application environment, there can be various kinds of noise and quality reduction. So it is possible that any sub-fingerprint in the query clip cannot be found in the index.

Therefore, we take actions to alter sub-fingerprints to generate more candidate positions. For a single bit in a sub-fingerprint, it can be flipped only from 0 to 1 or from 1 to 0. So, we set the possible flipped number of bits in a sub-fingerprint as F. At first, we flip one bit with respect to all the sub-fingerprints in the fingerprint block. This will result in 32 times more fingerprint comparisons, which is acceptable. If all the sub-fingerprints have been used to generate candidate positions and no match below the threshold has been found, we repeat the process by flipping 2 to F bits. If all possible F bits have been flipped and still no match is found, the algorithm decides that it cannot identify the song. This approach will lead to C_{32}^2 + ... + C_{32}^F times more fingerprint comparisons in theory, which seems unacceptable. However, the audio fingerprint extraction occupies a majority of time in the whole search process, while the time of generating candidate positions and computing BER is tiny. Experiments show that when the flipped number F is set to 2, the total search time increases slightly but the recall rate reaches an extremely high level.

3. CONCLUSION

Here, we proposed an efficient music identification system which utilizes a kind of space-saving audio fingerprints. Our method is based on Haitsma *et al.* [1]. We improved the method by sub-sampling the original fingerprint block every four sub-fingerprints and using the new sub-block to build the index and evaluate similarity. Also, we tried to flip every two bits in each sub-fingerprint to generate more candidates for further identification.

4. REFERENCES

[1] Haitsma, Jaap and Kalker, Ton: "A Highly Robust Audio Fingerprinting System.," *ISMIR*, 2002.

[2] Haitsma, Jaap Andre and Kalker, Antonius Adrianus Cornelis Maria and Schimme: "Efficient storage of fingerprints," *Journal of New Music Research*, US Patent 7,477,739, 2009.