A Real-time Peak Discovering Method for Audio Fingerprinting

Tao Jiang[1], Rihui Wu[2], Jiahong Li[2], Kang Xiang[2], Feng Dai[1]

[1] Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS)
{jiangtao01, fdai}@ict.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, 100190, China
{wurihui, lijiahong, xiangkang}@ict.ac.cn

ABSTRACT
Discovering peaks is a premise of audio fingerprinting algorithms for audio information retrieval, which focus on spectrogram peak pairs. In this paper, we discuss finding peaks in a two-direction scanning method which use a dynamic threshold vector to find local maximal point as peaks. And then an improved method using a slide window with two grids is proposed for discovering peaks at real-time. In this method, two key steps are executed alternately: scanning and sliding. In each scanning step, new peaks are found. When the slide window method is used, experiments on a database with 400,000 songs show that the average query time is shorten and the recall rate is slightly decreased.

Categories and Subject Descriptors
H.5.5 [**Information Systems**]: Information Interfaces and Presentation—Sound and Music Computing

General Terms
Algorithms, Experimentation

Keywords
Fingerprint, Peak, Real-time, Slide Window

1. INTRODUCTION
In audio information retrieval applications, there is a need to extract fingerprint from an audio clip, which is a compact content-based signature. Audio fingerprint uniquely represents the features of an audio recording, which is used to build up effective schemes to compare the quality of two clips of audio content. Audio fingerprinting is now attracting more attentions, since it can identify an audio recording independently of its format, without the need of metadata or watermark embedded [1]. There are numbers of audio fingerprinting associated applications, including music recognition [2][3][4], integrity verification, watermark support [1], copyright detection [5], personalized entertainment and interactive television without extraneous hardware[6].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ICMICS'13, Aug. 17-19, 2013, Huangshan, Anhui, China.
Copyright 2013 ACM 978-1-4503-2252-2/13/08 …$15.00.

The fingerprints are typically extracted from the spectrogram – a time frequency representation of the audio. There are mainly two groups of state-of-the-art fingerprinting algorithms [7]. The first group is based on computer-vision features of spectrogram. Those algorithms compute fingerprints over relative large spectrogram images. Using a computer vision technique, Ke et al. [8] train AdaBoost classifiers based on box-filters, which is often used in face detection. Baluja and Covell [5] propose a method based on wavelet of spectrogram. The overlapping spectrogram images are transformed into a sparse wavelet representation and the min-hash technique [9] is used to speed up the matching process. In contrast to the first one, Wang [2][3] proposes a scheme that only focus on spectrogram peaks. Ellis et al. [10] relies on the relative timing between note onsets that successfully detected in the audio.

Baluja and Covell [5] use a database with 10,000 songs and get a higher accuracy rate than Wang. 10,000 songs database is much smaller than database of Shazam [2][3] with over 1.8M songs. The scheme proposed by Wang is effective in large database. According to Wang, if a time-frequency point has a higher energy, it is a candidate peak. In this paper, we use local maximal points as peaks to calculate fingerprints. A slide window is used to discover peaks at real-time. With the window sliding, new peaks are found at real-time. This method can be used in fingerprinting schemes which base on peaks in feature domain. Beside these two group algorithms, other algorithm such as [11][12] are also used.

The remainder of this paper is organized as follow. Section 2 first illustrate an example of audio retrieval application and then discusses our method in details. The experiments and the results are described in Section 3. Section 4 summarizes this paper.

2. METHOD DESCRIPTION
2.1 Process of Audio Information Retrieval Based on Peaks
Before audio information retrieval, fingerprints are calculated as templates and stored in database with metadata (e.g. song and artist name) of audio signal files. When a clip of unlabeled audio content is used as a query, fingerprints are calculated from it. Then the query is matched against templates stored in the database. The query may be distorted by noise or microphone. Finally, queries are linked with metadata correspondent to the template which contains the undistorted version.

Fingerprinting schemes focus on spectrogram peaks, and they are based on pairs of peaks [3][10]. One typical audio fingerprinting system is described in [2][3]. Wang [3] proposed a combinatorial hashing technology settled on spectrogram peaks only, because of their approximate linear superposability and robustness. Figure 1 gives an example of this fingerprinting scheme. Using each

selected pair of peaks (t_1, f_1) and (t_2, f_2), the fingerprint is calculated on a triplet of$((t_2 - t_1), f_1, (f_2 - f_1), t_1)$.

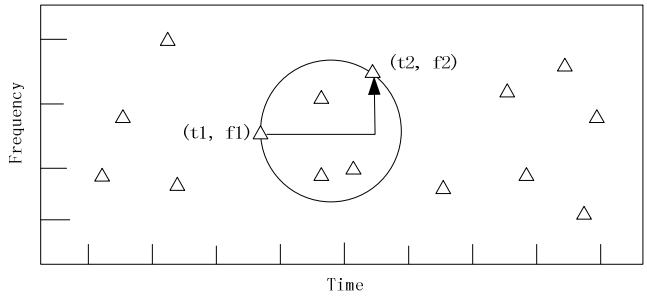

Figure 1: Illustration of fingerprinting proposed by Wang [3]

In the matching part, the first thing to do is to find the matching hashes. Each matching hash found in database corresponding to a time pair, one for the time offset of query and one for the time offset of database file. A scatterplot is used to represents those time pairs. If an example matches a database file, there should be a set of points in the scatterplot form a diagonal line. Wang [3] proposed a simple and useful technique to get this diagonal in approximately $N * \log(N)$ time, where N is the number of points in the scatterplot. It is assumed that the slope of the diagonal line is 1.0. Time pairs of matching features between matching files have a relationship

$$t'_k = t_k + offset \quad (1)$$

Where t'_k is the time coordinate of database sound recording file and t_k is the time coordinate of sample clip. For each(t'_k, t_k), Wang calculate

$$\Delta t = t'_k - t_k \quad (2)$$

Then a histogram is built up by Δt, and the score is the number of the matching points in the histogram peak. The value of a threshold is chosen by experiments.

2.2 Two-direction Scanning Method

The Peaks of the audio clip can be gotten by using a two-pass scanning method. The audio clip is firstly transformed into a feature vector series. In this method, the audio clips are scanned from the beginning to the end. Then the scanning direction is reversed. In the scanning processes a threshold vector is dynamically updated. After the threshold vector initialized, the comparing step and the decaying step are taken alternately.

In the comparing step, the threshold vector is compared with the feature vector in each dimension. One example of the comparing process is shown in Figure 2. Little circles are candidate peaks have been found; little discs are new candidate peaks found in this comparing step since they are greater than the corresponding dimensions of threshold vector; litter squares are values in the threshold which will be replaced by discs next to them.

In the degrading step, each dimension of the threshold vector degraded according to a given function. In this paper, the degrading scheme is multiple a constant small than 1.

Values in a given dimension of a series of feature vectors are described as the curve in Figure 3. The little circle, disc and little squares are candidates found in the two pass of scanning respectively. The little circles and disc are found in the first scanning from left to right. Little triangles are peaks which are candidates in both scanning. On the rising edge of curve in Figure 3, a candidate is discovered for it is greater than the threshold value in the corresponding comparing step. So it is greater than its left neighbor, or the threshold value equals to the left one's and the right one would not be chosen as a candidate. Similarly, when a candidate is found on the falling edges, candidates are greater than their right neighbor. The points, which are chosen as candidates in both pass of scanning, are local maximal points.

After the first local maximal is chosen as a candidate in a comparing step, it is also chosen as a threshold value. Then the value degrades as discussed above. New candidates are found in the next scanning step. That is the reason why the disc and black triangle can be found although they are smaller than the hollow triangle. The black square is founded in a similar way.

In this curve, the false local maximal are ignored since they are useless to the matching. They may come from noise or distortion and should not be treated as peaks.

This method can discover peaks robustly. Peaks could not be found in the first pass of scanning. And peaks in the beginning of the clips would be found in the end of second pass of scanning. Matching process taken at after that those peaks found. Delay of matching relay on the length of clips.

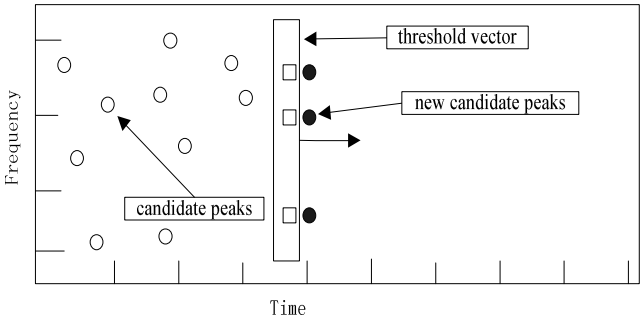

Figure 2: Picking out candidate peaks

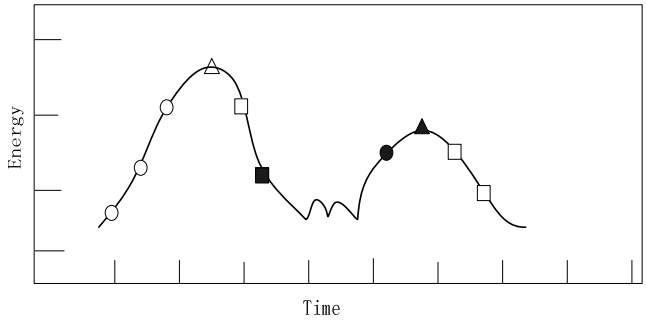

Figure 3: Discovering peaks

2.3 Real-time Method with Slide Window

In order to get peaks at real-time, we propose an method using a slide window with two grids with equal width, as shown in Figure 4. There are two key steps executed iteratively in this method: scanning and sliding. The scanning method is the same as shown in section 2.2, starting from the edge between the two grids towards two directions.

Figure 4 illustrates a result after scanning and before sliding. Little triangles are peaks, which are picked out as candidate peaks in both grids. Little circles and little squares represent candidate peaks found in left grid and right grid respectively. There are little squares in the left grid，which are found in the former scanning step . Little squares only picked out when the right grid is scanned.

Then the slide window move towards right, the sliding step, as shown in Figure 5. The slide window stops until its left edge meets the first candidate peak, which used to be in the right grid.

In this process, no new candidates are founded, and peaks and candidate peaks can only be discovered in the scanning steps.

Figure 6 shows the next scanning. Four little squares turned into little triangle since they are also candidates when scanning the left grid.

The method above can get peaks at real-time. Candidate peaks in the right grid of Figure 4 come from a left to right scanning. And those ones in the left grid of Figure 6 come from a right to left scanning. According to method discussed in section 2.2, little triangles in Figure 6 are peaks. In our method, new peaks are found with the shift of slide windows at realtime.

By this method, fingerprints can be extracted at real time with the uploading of audio stream, which is important to applications that require timeliness. When using the method shown in section 2.2, no peaks picked out before first pass of scanning finished. The delay of the method shown in section 2.2 is constrained by the length of clips. If the clip is too short, it may contains not enough peaks; if it is too long, the users have to wait for a long time before the result comes out. The time delay is the upload time plus the time of one pass of scanning. The slide window method is not constrained by the length of clip, when the number of peaks is more than the threshold, the uploading can be stopped. The slide window method can get peaks and fingerprints at real-time, and the matching can be processed at realtime. So the time delay is decide by the time of scanning the slide window.

In the beginning of executing this method, the left grid have to be scanned in two directions before the first moving of the slide window, or peaks in the beginning of audio clips would lost.

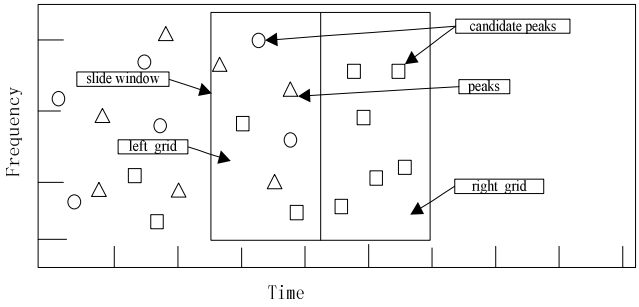

Figure 4: Status after scanning step and before sliding step

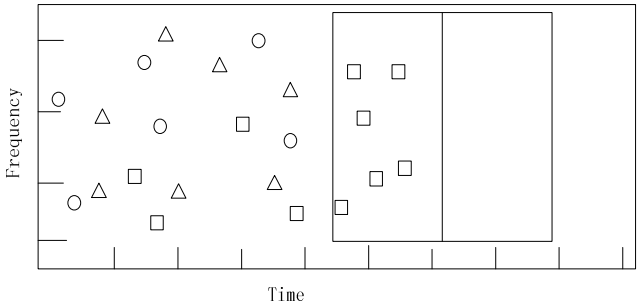

Figure 5: Sliding step

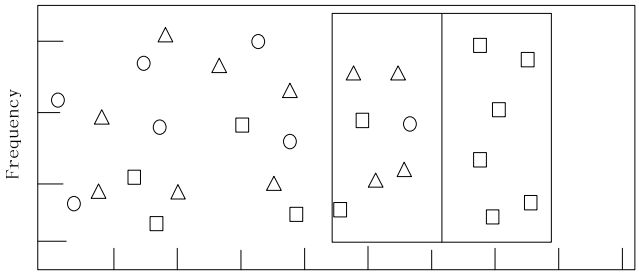

Figure 6: Scanning step

3. EXPERIMENTS AND RESULTS

3.1 Experiments

To evaluate performance of the method proposed, we first set up an audio information retrieval system with a database containing 400,000 tracks of various genres and with a corresponding fingerprint database based on the algorithm proposed by Wang [12]. Simple versions of slide window method are implemented. Each clip is divided into segments. The slide window is as wide as a segment and slides once in a segment length instead of a variable one. Inside the slide window, both grids are scanned in two directions. The performance is evaluated by average response time and recall rate.

In experiment I, 1413 mono tracks are used, which sampled at 16 kHz, 16-bits sampling. Each clip is extracted from a track beginning at a random position and last for 10 seconds. After extraction, gaussian white noise is added to all the clips, gaining 0 dB SNR. Clip set is used to created test sets. There three test sets, the first one contain 10 seconds clips undivided; in the second one, each clip is used as a query, and it is divided into two 5 second segments; in the third one, segments are 4 second, 4 second and 2 second.

In experiment II, clips are divided into 4 seconds length of segments. There are three test sets cut from 783 songs recorded by mobile services such as cell phone, mp3 player, voice recorder, and computer microphone in real environments. Songs are played by cell phones and recorded in noisy situation such as supermarket, street, canteen and so on. The test sets contain clips of 12 seconds length. Different test sets are cut from different parts of those songs and clips in the same set are cut from the same part of songs. Test clips include beginning part, middle part start from random place, and end part of those songs.

Tests are done on first segments, first two segments, all three segments and 12 seconds clips undivided. Distribution of test set is shown in Table 1. The first column indicates which segments are used as queries. The three columns in the middle of Table 1 are number of clips are tested. First lines of these columns shown which part of songs the clips come from. The last column calculates the total of clips.

Table 1. Distribution of test set

Segments	Beginning	Middle	End	Total
4s*1	783	782	775	2340
4s*2	782	782	773	2337
4s*3	782	782	773	2337
12s	782	782	775	2340

3.2 Results

Results of experiment I are shown as Table 2.

Table 2. Results of Experiment I

Test set	average query time	recall rate
10s	10s	92.5%
5s*2	6.02s	91.2%
4s*2+2s	5.30s	87.4%

Results of experiment II are shown as Table 3. Each percentage is a recall rate of a test.

Table 3. Results of Experiment II

Segments	Beginning	Middle	End	Total
4s*1	40.1%	45.7%	41.4%	42.4%
4s*2	63.0%	63.8%	59.3%	62.0%
4s*3	69.1%	71.1%	68.3%	69.5%
12s	74.0%	77.2%	75.6%	75.6%

As shown in Table 2, along with dividing clips into segments, the average query time shortens, and the recall rates decrease.

Since peaks are found in experiment I after one pass of scanning of each segment, not the undivided clip, the average query time is shorter. Using a smaller window, the average query time would be shorter. Fingerprints are computed inside each slide window. As discussed in section 2.1, in fingerprinting schemes focus on spectrogram peaks, fingerprints are calculated with pairs of peaks. Peaks pairs would lose if peaks are in different sides of a slide window edge. So the recall rates decrease.

Two experiments above are simplifying versions of real-time slide window method. Both grids of the slide window are scanned in two directions. New peaks are found after first pass of scanning and all the new peaks are picked out in the end of second pass of scanning. Compared to original method, double time is used to find new peaks and brings more delay. The original method would not calculate fingerprints inside the slide window since number of peaks found each time is dynamically. Then no pairs of peaks would lose and the recall rates are higher than the simplifying versions.

The performance of original method would be better than using all the three segments and worse than using the clips undivided. The average query time would shorter than the average query time of both the two testing and the recall rates would be between theirs.

4. CONCLUSION

This paper demonstrates a accelerate process in fingerprinting algorithm of audio information retrieval. In this work, a slide window method can find peaks at real-time is proposed. By executing scanning and sliding steps alternately, new peaks are found. If a smaller slide window is used, the method brings a smaller time delay. The experiment results shown that this method would produce a decrease in recall rate.

5. ACKNOWLEDGMENTS

This work is supported by National Nature Science Foundation of China (61102101, 61273247), National Key Technology Research and Development Program of China (2012BAH06B01, 2012BAH39B02), Co-building Program of Beijing Municipal Education Commission.

6. REFERENCES

[1] Cano, P., Batle, E., Kalker, T. and Haitsma, J. *A review of algorithms for audio fingerprinting.*, 2002.

[2] Wang, A. The Shazam music recognition service. *Commun Acm*, 49, 8 2006), 44-48.

[3] Wang, A. and others *An industrial strength audio search algorithm.*, 2003.

[4] http://www.midomi.com/

[5] Baluja, S. and Covell, M. Content fingerprinting using wavelets2006).

[6] Fink, M., Covell, M. and Baluja, S. *Social-and interactive-television applications based on real-time ambient-audio identification.*, 2006.

[7] Chandrasekhar, V., Sharifi, M. and Ross, D. A. *Survey and evaluation of audio fingerprinting schemes for mobile query-by-example applications.*, 2011.

[8] Haitsma, J. and Kalker, T. *Speed-change resistant audio fingerprinting using auto-correlation.*, 2003.

[9] Seo, J. S., Haitsma, J. and Kalker, T. *Linear speed-change resilient audio fingerprinting.*, 2002.

[10] Ellis, D. P., Whitman, B. and Porter, A. *Echoprint: An Open Music Identification Service.*, 2011.

[11] Park, M., Kim, H. and Yang, S. H. Frequency-temporal filtering for a robust audio fingerprinting scheme in real-noise environments. *Etri J*, 28, 4 2006), 509-512.

[12] Ke, Y., Hoiem, D. and Sukthankar, R. *Computer vision for music identification.*, 2005.