AN IMPROVED LANDMARK-BASED METHOD FOR ROBUST AUDIO FINGERPRINTING

ABSTRACT

In this paper, we propose an efficient landmark-based audio fingerprinting method for query-by-example under noisy environments. To increase the robustness of the audio fingerprinting from the experimental observation, we propose to apply a high pass filter in each frame from the spectrogram. Then we examine the horizontal and vertical peaks simultaneously to get more discriminative peak pairs. Landmarks are represented by peak pairs using the temporal and the spectral distances between two adjacent peaks. Finally, the landmarks are mapped to hash values to represent audio fingerprinting. In this system, we proposed method outperforms a well-known baseline audio fingerprinting method regarding noise-robustness. Furthermore, it is computationally efficient as it achieves such improvement with a much smaller number of audio fingerprinting.

1 Baseline

The baseline system, is based on Wang [2]. Because the code is not available, adaptive code was produced by Dan Ellis [1]. The baseline system is illustrated in Fig. 1.

1.1 Spectrogram Signal Processing

First, all the songs and audio clips are first converted into mono and downsample to 8KHz ($\boldsymbol{x} = \{x_1, ..., x_n\}$). Then these signal was convert into spectrogram by Short-Time Fourier Transform (STFT), where the frame length is 64ms and the frame shift is 32ms. Finally, we get the absolute value of the spectrogram:

$$X_{ft} = |\text{spectrogram}(\boldsymbol{x})| \tag{1}$$

where t is the t-th value of the frame in the spectrogram and f is the f-th value of the channel in the spectrogram. To reduce the background noise in audio clips. Thus, it is limited by dynamic range in the spectrogram (find a lower bound, is the maximal value divide by 10^6 from Eqn. 1) and compression of energy signals using natural logarithm:

$$A_{ft} = \ln(\max(X_{ft}, \text{LB})), \quad \text{LB} = \frac{\max(X)}{10^6} \tag{2}$$

Finally, A_{ft} is normalized by subtracting the mean value, so that the average value of A_{ft} is 0:

$$B_{ft} = A_{ft} - \mu, \quad \mu = \frac{\sum_{f=1}^{F}\sum_{t=1}^{T} A_{ft}}{F \times T} \tag{3}$$

1.2 High Pass Filter

The purpose of the high-pass filter allows high frequencies to go through and cutting low frequencies. The baseline system will produce B_{ft} from the previous step, using high-pass filter for each row(each channel) of B_{ft} :

$$Y_{ft} = B_{ft} - B_{f(t-1)} + 0.98 \times Y_{f(t-1)} \tag{4}$$

1.3 Landmarks in acoustic representation

For the data points in each frame, if one data point is higher than two neighbors in the channel, it will be treated as a peak. In order to obtain more discriminating peaks, therefore, it will use some smoothing techniques to extraction the peaks. The steps are: Establish envelope, forward smoothing and backward smoothing.

1.3.1 Establish envelope

1. Before generating the envelope, find a set of data points first. Then extract the first ten columns (the first ten frames) of Y_{ft} to get the $F \times 10$ matrix and find out the maximum of each row to get a $F \times 1$ column vector $\boldsymbol{a} = [a_1, ..., a_F]^{\mathbf{T}}$ from the matrix.

2. Local maximum points Identification: In vector $\boldsymbol{a}$, which is found by the previous step, if the data point a_f in a is higher than two neighbors data points (a_{f+1}, a_{f-1}), it will be treated as a local maximum point. The size of the local maximum point is expressed as $\boldsymbol{\rho} = \{\rho_1, ..., \rho_M\}$. The position of channel of the local maximum point is expressed as $\boldsymbol{\ell} = \{\ell_1, ..., \ell_M\}$, where M is the number of local maximum points.

3. GaussianTable generation : GaussianTable is a $F \times M$ matrix. Each column in GaussianTable represents a local maximum point multiplied by the Gaussian kernel. Gaussian kernel function is defined as :

   $$G(f, \ell_m) = \exp(-\frac{1}{2} \times (\frac{f - \ell_m}{30})^2) \tag{5}$$

   GaussianTable is calculated as follows:

   $$\text{GaussianTable}_{fm} = \rho_m \times G(f, \ell_m) \tag{6}$$

   Where $f = 1, ..., F$,$m = 1, ...M.$

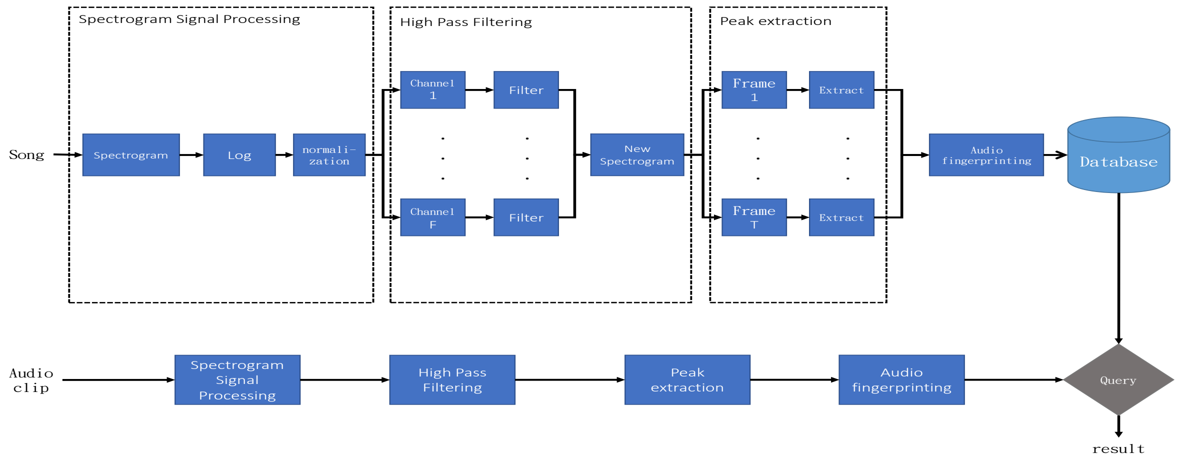

Figure 1. Overview of the baseline system.

4. Envelope generation : According to GaussianTable, find the maximum value of each row to get $F \times 1$. **Envelope**$=$ $[\text{Envelope}_1, ..., \text{Envelope}_F]^{\mathbf{T}}$.

1.3.2 Forward smoothing

1. Find peaks : First, extract the first column (the first frame) of Y_{ft} to get the $F \times 1$ vector. For these data points, if it is a local maximum point and greater than the envelope ($Y_{f1} > \text{Envelope}_f$) will be treated as a peak. Next, these peaks will be in descending order according to peak size and expressed as $p = \{p_1, ..., p_K\}$. Where $l = \{l_1, ..., l_K\}$ is the channel position of peak, K is the number of peaks, and K not more than five, i.e., each column(frame) retain only the top five highest peak.

2. Envelope update : The peaks multiplied by the Gaussian kernel and compared with the envelope to update envelope:

$$\text{localmax}_f = p_k \times G(f, l_k) \tag{7}$$

$$\text{Envelope}_f = \max(\text{localmax}_f, \text{Envelope}_f) \tag{8}$$

Where $f = 1, ..., F$, $k = 1, ...K$.

3. Record peak : For peak retained, the frame position, the channel position and the data point Y_{ft} will be stored in dataTable1.

4. When processing the next frame, the envelope will be multiplied by 0.9943 to decay envelope. When the process is completed for all frame, dataTable1 records all peak information.

1.3.3 Backward smoothing

1. Calculation the envelope : The way with the same sectoion 1.3.1, but it is calculated to produce a set of data points from the first end frame.

2. Peak selection : First, choose a peak from the last data of datatable1. When the current frame and the frame of which position is the last data of datatable1 are the same, then compared to the size. If the last data of datatable1 is greater than or equal the envelope ($Y_{ft} \geq \text{Envelope}_f$), then this data will be retained and updates the envelope. The update methods and forward smoothing envelope updated are in the same way. If not, it will be moved to the frame position which dataTable1 is currently pointing at , then compare the envelope, and each time to process a front frame ,the envelope will be multiplied by 0.9943 to attenuate the envelope.

3. Access peak information : The peaks which are retained by previous step will be stored the belong frame and the channel positions in dataTable2.

4. When the data dataTable1 are screened, these data are retained in dataTable2, that is a piece of music represented by the signal peak.

1.4 Peak Pair Analysis

Each fingerprint represented by one landmark in the music. For a landmark, first find a peak information from dataTable2, known as anchor point. As shown in Fig. 2. The dotted square represents a target zone. The target zone range is from the frame position of anchor point plus 64 frames and the channel position of anchor point plus and minus 32 channels. Then find the three nearest peak from the target zone and composition with anchor point to get peak pairs. The peak-pair will be represented in the form of the landmark :

$$\text{Landmark} = (t_1, f_1, f_2, t_2 - t_1)$$

After all peak in dataTable2 have been the anchor point, it will get all the fingerprints of a song.

1.5 Establish a database

In addition to recording song author, lyrics and other information, music database is the most important thing hash table. Hash table contains hash key and hash value.

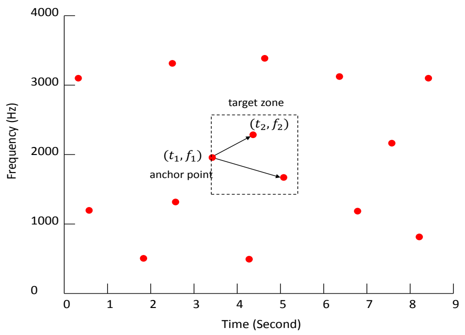

Figure 2. Illustration of audio fingerprinting.

1. Hash key : before fingerprints are stored in the database, it will be converted into the form of hash key. Calculated as follows.

   $$(f_1, \Delta f, \Delta t) = (f_1, f_2 - f_1, t_2 - t_1)$$

   Because short-term Fourier transform taken from 512 points and 256 points per shift, each frequency channel of 8 bits.

2. Hash value : hash value contains Song ID, and landmark t1. Calculated as follows.

   Hash value $=$ uint32(Song ID$\times$16384$+t_1$%16384)

1.6 Song Identification

During searching for the song, first retrieve the fingerprints of audio clip and then calculate hash key and landmark t_1 by the fingerprints. Then find the matching hash keys of hash table in the database, and record the contents of which are included in the hash keys. Because the hash value contains the song ID and t_1, so it could compile statistics to find out the most likely songs in these two conditions.

In addition to the number of comparisons song ID appeared, but it also add an offset time approach. First identify each song appeared offset time from the previous one calculated song ID rankings. Offset time is t_1 of hash value of database (named DBt_1) minus t_1 of audio clip (named qt_1). Then store each song offset time according to the occurrences to obtain results and add the error (plus and minus one frame) of time. After the ranking method via offset time, and then re-rank all the songs, the results can be obtained.

2 Proposed Method

Our implementation of the baseline system [2] is based on the work by Ellis [1]. Our proposed system is illustrated in Fig. 3. The differences between our implementation and the baseline are summarized as follows.

2.1 Change the filter mode

In this system, we change the filter mode and use high-pass filter for each column (each frame) of B_{ft}, As shown in Eqn. 9.

$$Z_{ft} = B_{ft} - B_{(f-1)t} + 0.98 \times Z_{(f-1)t} \tag{9}$$

2.2 Change the peak extraction mode

In addition to peak extraction of the system based on frame, we also change the peak extraction mode from each channel. The steps are: Establish envelope, forward smoothing and backward smoothing.

2.2.1 Establish envelope

In the proposed method, first we will set up the two envelopes. The details are as follows :

1. For the first envelope, we take the first ten column (the first ten frame) data points from Z_{ft} in order to get the $F \times 10$ matrix. And find the maximum value for each column to get a row vector. For the second envelope, first take the first ten row (the first ten channel) data points to get the $10 \times T$ matrix. And find the maximum value for each column to get a row vector.

2. Local maximum points Identification : For these two vectors, if the data point in a is higher than two neighbors data points, it will be treated as a local maximum point.

3. GaussianTable generation : There are two Gaussian Table in our proposed method. Each column in GaussianTable1 represents a local maximum point multiplied by the Gaussian kernel. Each row in GaussianTable2 represents a local maximum point multiplied by the Gaussian kernel.

4. Envelope generation : According to GaussianTable1, find the maximum value of each row to get the envelope, of which size is $F \times 1$ **Envelope1** For GaussianTable2, find the maximum value of each column to get the envelope, of which size is $1 \times T$ **Envelope2**

2.2.2 Forward smoothing

In forward smoothing, the same as the first envelope and the second envelope peak extraction method. The main difference is that one is performed channel-wise peak extraction, the other is performed frame-wise peak extraction and dose not limit the peak (the baseline limit up to five in each frame) in each channel and each frame, as follow :

1. Find peaks : First, extract the first frame and the first channel of Z_{ft}. For the first frame data points, if it is a local maximum point and greater than the envelope then will be treated as a peak. For the first channel data points, if it is a local maximum point and greater than the envelope will be treated as a peak.

2. Envelope update : For Peak extraction in the frame and channel, the peaks multiplied by the Gaussian kernel and compared with the envelope to update it. The updated of envelope is the same as above.

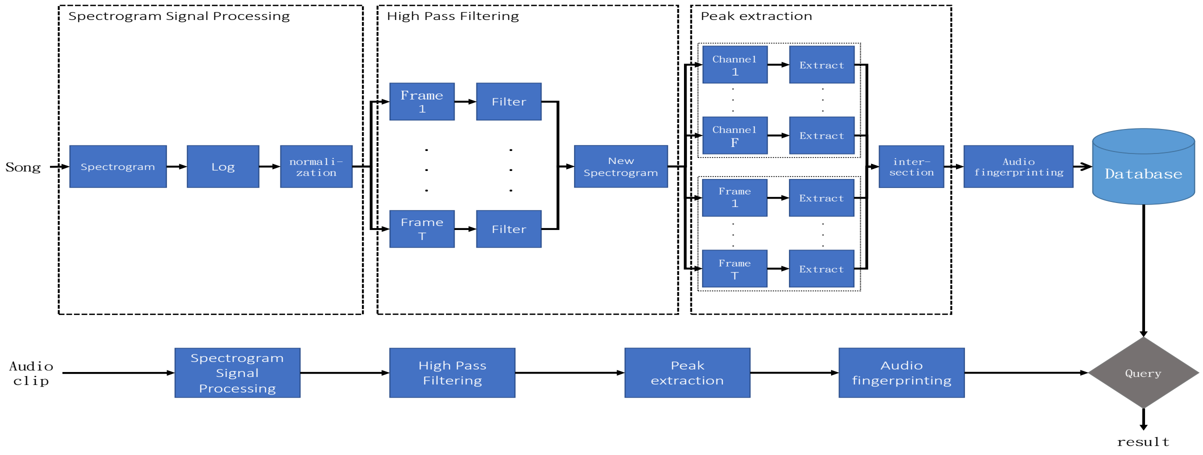

Figure 3. Overview of the proposed system.

3. Record peak : For retained peak based on the frame, the frame position, the channel position and the data point Z_{ft} will be stored in dataTable11. For retained peak based on the frame, then will be stored in dataTable12.

4. When processing the next frame or channel, the envelope will be multiplied by 0.9943 to decay envelope. When the process is completed for all frame or channel. Next, dataTable11 and dataTable12 are intersected, and the contents of intersection is stored in dataTable1.

2.2.3 Backward smoothing

Backward smoothing is similar to the baseline. Details are as follows :

1. The initial envelope calculation : The envelope also has two, is calculated to produce two sets of data points from frame T and channel F, respectively. There are two envelopes, one is generated by frame T and the other is generated by channel F.

2. Peak selection : Peak selection by the frame with the baseline. When the current frame and the frame of which position is the last data of datatable1 are the same, then we compare them with the size. If the last data of datatable1 is greater than or equal to the envelope, then this data will be retained and we update the envelope. If not, it will be moved to the frame position which dataTable1 is currently pointing at, then we compare the envelope. Each time we process a front frame, the envelope will be multiplied by 0.9943. Similarly, when the current channel and the channel of which position is the last data of datatable1 are the same, then we compare them with the size and do peak selection. If not, it will be moved to the channel position which dataTable1 is currently pointing at, then we compare the envelope. Each time we process a front channel, the envelope will also be multiplied by 0.9943.

3. Access peak information : The belong frame and the channel positions of the peaks which are retained by the previous step based on frame will be stored in dataTable21. The belong frame and the channel positions of the peaks which are retained by the previous step based on channel will be stored in dataTable22.

4. After the data in dataTable1 are selected, then dataTable11 and dataTable12 are intersected. And the contents of intersection are stored in dataTable2. These data are retained in dataTable2, that is a piece of music represented by the signal peak.

3 Conclusion and Future Works

We have proposed a 2-dimensional landmark generation methods. This is particularly desired as the order of peak extraction does appear to matter. With using less fingerprints, we achieved better music retrieval accuracy than the baseline method.

4 References

[1] Dan. Ellis. Robust Landmark-Based Audio Fingerprinting. web resource, `http://labrosa.ee.columbia.edu/matlab/fingerprint/.`, 2009.

[2] Avery LiChun Wang. An industrial-strength audio search algorithm. *Proceedings of International Conference on Music Information Retrieval (ISMIR)*, 2003.