A Robust Feature Extraction Algorithm for Audio Fingerprinting

Jianping Chen[1], Tiejun Huang[2]

[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[2] Key Laboratory of Machine Perception(Ministry of Education), Peking University, Beijing 100871, China
{jpchen, tjhuang}@jdl.ac.cn

Abstract. In this paper, we present a new feature extraction algorithm which can generate robust and reliable feature in a fingerprint system. This algorithm is referred to as weighted ASF (WASF). The feature in our algorithm is extracted based on a MPEG-7 descriptor-Audio Spectrum Flatness (ASF) and Human Auditory System (HAS). It also applies several effective filters to improve the feature robustness and uses another MPEG-7 descriptor: Audio Signature (AS) to reduce the feature dimension and increase the feature compactness. The smooth filter bank can efficiently resist the noise distortion in addition to some other common distortions such as sampling rate change and amplitude normalization, while the first order inverse filter can effectively resist the speed-change distortion with 90.1% discrimination for the 5% speed acceleration distortion. This algorithm is tested under several audio distortions: sampling rate change, noise addition, data compression and speed-change and so on. For these distortions, the WASF algorithm can get discrimination more than 90%. The MFCC feature and another MPEG-7 descriptor-Audio spectrum Centroid (ASC) are also considered.

Keywords: Audio fingerprinting, weighted ASF, Audio Spectrum Flatness, filter bank, inverse filter.

1 Introduction

The increasing number of audio resources, especially in the network, and the intensity of Intelligent Property (IP) protection has increased the interest in techniques for automatic audio identification. There are two main approaches: watermarking and fingerprinting. In the last few years, the fingerprinting technique has brought much more attention. The audio fingerprinting technique can be used in many applications [1], such as file sharing services, broadcast monitoring and so on. In digital rights management (DRM) system [2], the fingerprinting technique is also urgently required for the protection of Intelligent Property of the owner of media rights.

In general, a fingerprinting system needs to have the following properties: robustness, reliability, compactness and scalability. The robustness indicates that the fingerprinting system can resist various common audio distortions. The reliability

indicates the fingerprinting system should give continuous right results over a wide variety of inputs. The compactness indicates the fingerprinting data should be small and need small storage. The scalability indicates the system can be not only run in large devices but also in resource-constrained devices.

Recently, there are some researches on this topic. In [1], Haitsma and Kalker calculate the energy difference of the inter-frame and intra-frame and convert it to bit value and then use a sequence of bits to form an audio fingerprint. In [2], the square root of the mean energy across the time concatenating the standard deviation of the RMS power is used to form a fingerprint. The MPEG-7 audio descriptors-Audio Spectrum Flatness and Audio Signature are used to form the fingerprint in [3]. And in [4], a two-layer OPCA technique is used to generate the noise-resistant fingerprinting. In [5], the normalized spectral sub-band moments has been used to generate an efficient fingerprint. Computer vision and image process methods are also introduced into the audio process in [6] [7]. For these algorithms, they are mostly aimed to several distortions and don't efficiently resist the speed-change distortion. The speed-change distortion is referred to in [8]; it is based on the work of [1].

In this paper, we use the weighted MPEG-7 descriptor: Audio Spectrum Flatness [3] [9] to generate our audio feature because the perceptual feature computed using ASF can efficiently characterize the audios and be robust to a variety of audio distortions. Otherwise, we use many effective filters to reduce distortions, especially the noise and speed-change, and make use of two ear process functions in Human Auditory System (HAS) [10] to enhance the property of the audio data. In order to compact the fingerprint, we use MPEG-7 descriptor -Audio Signature. This descriptor can efficiently compact the data and maintain the feature robustness.

The rest of this paper is organized as follows. After this introduction, section 2 describes the proposed fingerprinting extraction algorithm in detail. Section 3 shows the experimental results. Finally, the conclusion of the work and the acknowledgment are given in section 4 and section 5 respectively.

2 Proposed Audio Fingerprinting Algorithm

In this section, we describe the fingerprinting extraction algorithm of this system. The framework is shown in Fig.1. This framework can be partitioned into three parts: front-process, feature computation and end-process.

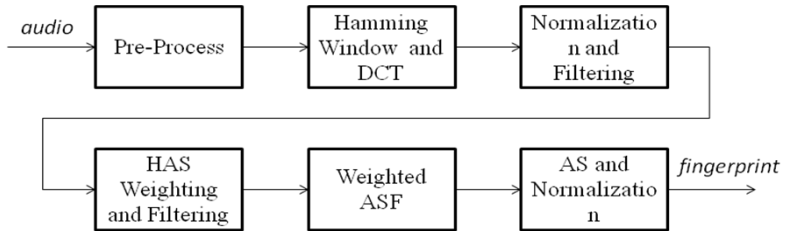


Fig.1- the framework of the feature extraction algorithm

2.1 Front-Process

This step contains pre-process, framing, time frequency transformation and data filtering. A stereo waveform should be converted into a mono waveform in the pre-process phase because this proposed algorithm is aimed to the mono waveforms. In order to extract robust feature from the dynamic audio data, a framing window should be applied to the audio waveform to obtain relatively static audio clips. There are some optional windows, such as rectangular window, hanning window, hamming window and blackman window. In our algorithm, we select hamming window to frame the audio $H(i) = 0.54 - 0.46\cos(\frac{2\pi i}{N-1})$, where N is the number of samples in each window frame such that $0 \le i < N$.

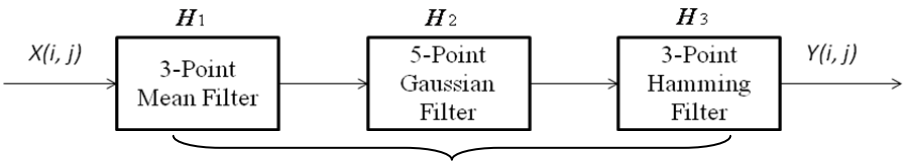
We have tested several frame lengths and found that longer frame length can give more perceptual information but take more time. In our method, we set each frame length 90ms and inter-frame overlap rate 2/3. In this way, we can reduce the discontinuity of the data. Usually, the overlap rate should be set larger than 1/2 to get better continuity. Then, we apply the Discrete Cosine Transform for each frame to generate the frequency spectrum. After the transformation, a normalization process is needed; it is the combination of two methods as follows:

$$Y(i, j) = \frac{X(i, j) - \mu}{\sigma} \quad (1)$$

$$Z(i, j) = \frac{Y(i, j) - \min}{\max - \min} \quad (2)$$

Where X(i, j) is the j^{th} sample data of the i^{th} frame, μ and σ is the mean and standard deviation of the i^{th} frame respectively, min and max is the minimum and maximum data of Y(i, j). These two functions make the audio data from different audio clips in the same range [0, 1].

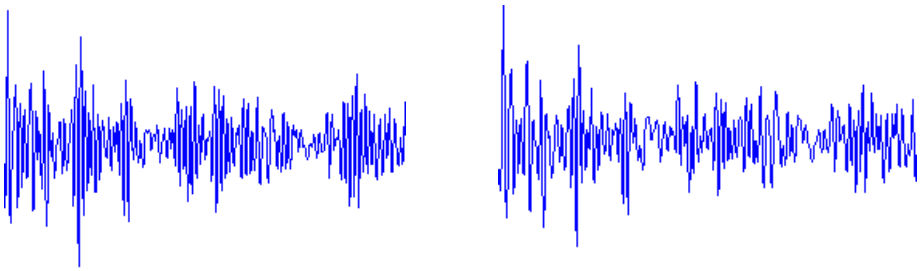
In order to reduce the noise distortion efficiently, whatever white noise or Gaussian noise, we use a smooth filter bank shown in Fig.2 to filter the data. This filter bank is composed of three smooth filters: a 3-point mean filter, a 5-point Gaussian filter and a 3-point hamming filer.



$$Y(i, j) = X(i, j)\, H_1\, H_2\, H_3 \quad (3)$$

Fig.2-smooth filter bank

From our experiment, we find this smooth filter bank is efficient to white and Gaussian noise in our weighted ASF algorithm. Fig.3 shows the result of an audio segment with 20% Gaussian noise addition distortion and processed by the smooth filter bank.



(a) (b)

Fig.3-(a) segment with 20% Gaussian noise addition (b) audio after smooth filter bank process

From Fig.3, we can see the segment with 20% Gaussian noise addition has been smoothed and the main perceptual property is maintained after the process of smooth filter bank. Of course, the more the number of the smooth filters in the filter bank, the smoother the audio frequency spectrum, but more local perceptual information will be weakened. Therefore, three smooth filters are enough.

After the smooth filtering, we should apply the HAS ear functions. According to the HAS, the functions in the inner ear and middle ear are respectively shown as:

i) Outer ear:

$$A_{db}(f_{kHz}) = -2.184(\frac{f}{1000})^{-0.8} + 6.5e^{-0.6(\frac{f}{1000}-3.3)^2} - 0.001(\frac{f}{1000})^{3.6} \quad (4)$$

ii) Middle ear:

$$W(f) = 10^{A_{db}(f)/20} \quad (5)$$

Where f is the frequency of each sample data in Hz.
In addition, there is a scaling factor for each sample data:

$$G_L = \frac{10^{L_p/20}}{\gamma(f_c)\frac{A_{\max}}{4}(N_F - 1)} KN_F \quad (6)$$

Where K is the energy compensation coefficient and is relative to the window function used when framing the audio, A_{max} is the maximum amplitude of the sample data, L_p is set to 92db, and N_F is the number of samples in a frame and $\gamma(f_c)$ varies from 0.84 to 1.

So for each sample data, we get a weight as follows:

$$WS(f)=G_L W(f) \quad (7)$$

If the sampling rate of an audio is 11.025 kHz, the weight curve of a clip with 0.09s length is shown in Fig.4:

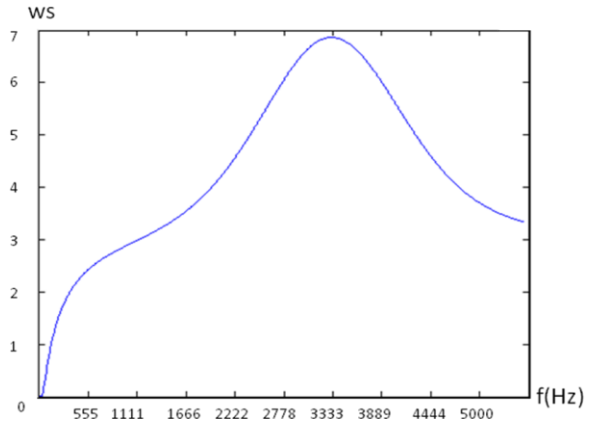


Fig.4-the weight curve generated by the HAS ear functions

From the Fig.4, we can see the weight increases nonlinearly in the frequency range about 250Hz-2000Hz. This weighted operation can enhance the perceptual property of this sensitive frequency range.

For the speed-change distortion, it causes misalignment both in the time domain and the frequency domain [8]. Common methods cannot efficiently resist this distortion. To resist the speed-change distortion, we should consider the distortions in the two domains. We find the all-zero first order inverse filter is efficient to this distortion in our algorithm and its z transformation is as follows:

$$A(z)=1+a_1 z^{-1} \quad (8)$$

This inverse filter can flatten the frequency response and get a good effect on signal-to-quantization-noise ratio versus frequency [11]. In our experiment, we set the

first-order coefficient a_1 to 0.95. This function can get a good result for the speed-change distortion.

However, it can weaken the efficiency of the noise distortion process. Therefore, an additional operation should be applied to each audio frame to avoid the distortion possibly brought by the inverse filter. Prior to the use of inverse filter, we use a hamming-like window function to generate a weight function increased by degrees, which is represented below:

$$w_i = (0.54 - 0.46\cos(\frac{2\pi i}{2(N-1)}))^2, \quad 0 \le i < N \tag{9}$$

This function can reduce the influence of the data in the low frequency range and maintain the tone-like property of the processed data. It is a better choice to balance the performance of noise addition process against speed-change process and still get a good result both in the case of these two distortions.

This process of the hamming-like window weight function and the inverse filter can bring about some noise to the resulting audio data, so a smooth filter bank should be applied to the audio data to reduce this distortion. In this smooth filter bank, we don't use the mean filter but a hamming filter, and then this filter bank contains two 3-point hamming filters.

2.2 Feature Computation

After the front-process, we begin to use the weighted ASF descriptor to compute the audio feature. To obtain a robust feature, we should get the most sensitive part of the frequency spectrum. In our experiment, we select the frequency range in 250 Hz-2000 Hz to extract the audio feature. Then the frequency spectrum of each frame is partitioned into bands in a logarithmic spacing and these bands are not overlapped. The number of bands in each frame is defined as follows:

$$bandNum = \frac{\log_2(hiFre / loFre)}{octave\,\mathrm{Re}\,solution} \tag{10}$$

Where hiFre and loFre are the upper and lower frequency limits of each frame, respectively and octaveResolution represents the logarithmic frequency resolution with the recommended range of 1/16 to 8 octaves, and we set it 1/4 in our algorithm.

In order to reduce the data and adapt the frequency resolution to "log" band, power spectrum coefficients above the frequency of 1 kHz are grouped and the average value is taken as a new value. The grouping is defined in the following way: between frequency 1 kHz and 2 kHz, two consecutive power spectrum coefficients are grouped.

Then we use the weighted ASF descriptor to compute the features for each frequency band. The weighted flatness measure is defined as the ratio of the

geometric and the arithmetic mean of the weighted power spectrum coefficients and shown below:

$$WASF = \frac{\sqrt[n]{\prod_{i=0}^{n-1} w_i P_i}}{\frac{1}{n}\sum_{i=0}^{n-1} w_i P_i} \quad (11)$$

w_i is the weight of each power spectrum coefficient. In our experiment, we set this weight as follows:

$$w_i = \frac{P_i}{\sum_{k=0}^{n-1} P_k} \quad (12)$$

In this way, we get a weighted ASF feature vector for each audio frame $WASF_i = [wasf_{i,0}, wasf_{i,1}, ..., wasf_{i,N-1}]$, where i is the index of a frame and N is the band number of the frame i.

2.3 End-Process

The feature generated from one frame is not enough to identify a whole audio clip, so M feature vectors generated from the part 2.2 are integrated to compose a feature block for identifying an audio clip. In our algorithm, we set M to 198 which is about 6- second length. Then we get a WASF matrix:

$$WASF = \begin{bmatrix} wasf_{0,0} & wasf_{0,1} & \cdots & wasf_{0,N-1} \\ wasf_{1,0} & wasf_{1,1} & \cdots & wasf_{1,N-1} \\ . & . & \cdots & . \\ wasf_{M-1,0} & wasf_{M-1,1} & \cdots & wasf_{M-1,N-1} \end{bmatrix}$$

For the feature matrix WASF, we subtract the mean of each row $\overline{m_i}$ ($0 \le i < M$) to make the mean of each frame feature zero in order to maintain the consistency of each frame feature.

For these M frames, the resulting feature is huge. In order to reduce the data, dimension reduction technique should be applied. We consider the MPEG-7 descriptor: Audio Signature. This descriptor uses a scaling factor to condense the audio date. According to [9], this condensation will not weaken the perceptual property of the audio data. This scaling factor is also called decimation factor *df*. In

our experiment, we set this decimation factor 24. Then in the WASF feature matrix, the number of blocks in the time axis is $b = \lceil m/df \rceil$, m is the number of frames. Then we can get a feature matrix with the dimension as b×N. Let S be the resulting feature matrix, then the arithmetic mean of each block is calculated as the new element in matrix S and denoted by:

$$S(k, j) = \frac{1}{df} \sum_{i=0}^{df-1} wasf_{i,j} \text{ , } 0 \leq j < N, \ 0 \leq k < b \tag{13}$$

Where k is the row index of matrix S.

In the end, a normalization process with function (**2**) will be applied to S, and then we obtain the final resulting feature which is denoted as *fingerprint*.

3 Experimental Results

To evaluate the performance of the proposed algorithm, we prepared 203 music audios, containing pop, rock, piano, flute, country music and so on. These source audios are all parameterized with 11025 Hz sampling rate, mono and 16 bits/sample. For each source audio, we make several distortions respectively as follows:

(a) 2s silence addition, (b) 80% amplitude normalization,
(c) sampling rate 22050Hz, (d) sampling rate 32000Hz, (e) sampling rate 44100Hz,
(f) mp3 compactness, (g) 20% white noise addition,
(h) 25% Gaussian noise addition, (i) 20% Gaussian noise addition,
(j) 5% speed acceleration, (k) free distortion.

In our experiment, we get a 6-second clip beginning from the location of 10s in each audio as our test clip. In this way, we get 2233 test clips and 203 source clips in all.

In addition to the weighted ASF algorithm, we also test the following algorithms:

(1) Audio Spectrum Centroid
(2) MFCC

We use the Euclidean distance to match the two comparing features. We set the fingerprint of the source clip and a distorted clip as S and D respectively, and then the distance between two frames from the source and distorted clips respectively is defined as:

$$distance_i = \sqrt{\frac{1}{n} \sum_{j=0}^{n-1} (S(i, j) - D(i, j))^2} \tag{14}$$

Where n is the number of frames and i is the row index of the fingerprint matrix. Then

the distance of S and D is defined as:

$$Dis\tan ce(D,S) = \frac{1}{b}\sum_{i=0}^{b-1} dis\tan ce_i \quad (15)$$

The experiment result is shown in Table.1.

Table.1-Experiment results

Methods / Distortion	ASC (%)	MFCC (%)	WASF (%)
(a)	90.1	100	100
(b)	100	100	100
(c)	93.4	100	100
(d)	95.6	100	100
(e)	99.0	100	98.0
(f)	88.7	98.5	91.6
(g)	99.5	98.5	99.5
(h)	98.5	99.0	99.5
(i)	97.0	96.6	97.5
(j)	43.3	46.3	90.1
(k)	100	100	100

From Table.1, we can see the weighted ASF descriptor has good performance to various distortions. Especially, the WASF algorithm has 90.1% discrimination to speed-change distortion while the other two algorithms have a lower discrimination less than 50%. It also has 97.5% discrimination to 20% Gaussian noise addition distortion and 99.5% discrimination to 20% white noise addition and 25% Gaussian noise addition. In addition to these distortions, the proposed algorithm has very high recognition rate. Of course, the discrimination of the mp3 compression distortion of weighted ASF is a little lower than that of the MFCC algorithm. On the whole, however, the WASF method can efficiently resist a variety of distortions.

4 Conclusion

For a good fingerprinting system, the extracted feature should be robust to various distortions and have a good reliability property. In this paper, the proposed algorithm weighted ASF is aimed for this purpose. From the experiment results, we can see that the proposed algorithm has over 90% discrimination rate to the ten distortions. Contrary to other algorithms, this proposed algorithm has better performance to many distortions than that of other algorithms. The next work we will do is to apply much

more test clips. The size of the resulting fingerprint is another issue we should pay attention to.

5 Acknowledgment

This work was supported by the National Key Technology R&D Program [2006BAH02A10 and 2006BAH02A13] of China.

References

1. J.Haitsma and T.Kalker, "A highly robust audio fingerprinting system", *Proc. Int. Conf. Music Information Retrieval*, 2002.
2. Vidya Venkatachalam, Luca Cazzanti, Navdeep Dhillon and Maxwell Wells, "Automatic Identification of Sound Recordings", *Signal Processing Magazine, IEEE*, 2004.
3. M.Sert, B.Baykal and A.Yazici, "A Robust and Time-Efficient Fingerprinting Model for Musical Audio", *IEEE Tenth International Symposium*, 2006.
4. Christopher J.C.Burges, John C.Platt and Soumya Jana, "Distortion Discriminant Analysis for Audio Fingerprinting", *IEEE Transaction on speech and processings*, Vol.11, No.3, May, 2003.
5. Jin S.Seo, Minho Jin, Sunil Lee, DalWon Jang, Seungjae Lee and Chang D.Yoo, "Audio Fingerprinting Based on Normalized Spectral Subband Moments", *Signal Processing Letters, IEEE*, 2006.
6. Yan Ke, Derek Hoiem, Rahul Sukthankar. "Computer Vision for Music Identification". *Processings of Computer Vision pattern Recognition*, 2005.
7. Shumeet Baluja, Michele Covell. "Audio Fingerprinting: Combining Computer Vision & Data Stream Processing". *ICASSP*, 2007.
8. J.Haitsma and T.Kalker, "Speed-change resistant audio fingerprinting using auto-correlation", *Acoustics, Speech and Signal Processing*, 2003.
9. ISO/IEC FDIS 15938-4:2001(E), "Information Technology-Multimedia Content Description Interface-Part 4: Audio".
10. P.Kabal, "An Examination and Interpretation of ITU-R BS.1387: Perceptual Evaluation of Audio Quality", *McGill University*, 2003.
11. John D.Markel, "Digital Inverse Filtering-A New Tool for Formant Trajectory Estimation", *Audio and Electroacoustics, IEEE transactions*, June, 1972.