Audio Fingerprinting based on Wavelet Spectral Entropy

Kyi Chan Nyein Linn
University Of Computer Studies, Yangon
kyichan88@gmail.com

Abstract

Audio-Fingerprints (AFPs) are essential characteristics of digital audio streams used to score the perceptual similarity between audio signals. Audio-fingerprinting systems extract features from the signal normally on a frame by frame basis. In this paper, a robust audio-fingerprint (AFP) approach is developed based on spectral entropy in wavelet domain. To extract the fingerprints of a song, Shannon's entropy is determined from the spectral coefficients of each one of the first 24 critical bands according to the Bark scale. The performance of this AFP system is evaluated on a music database containing various genres. The robustness of the system is validated through degraded music signals in 4 different ways: white noise addition, lossy compression, lowpass filter and resampling.

1. Introduction

The prime objective of multimedia fingerprinting is an efficient mechanism to establish the perceptual equality of two multimedia objects: not by comparing the (typically large) objects themselves, but by comparing the associated fingerprints (small by design). In most systems using fingerprinting technology, the fingerprints of a large number of multimedia objects, along with their associated meta-data (e.g. name of artist, title and album) are stored in a database. The fingerprints serve as an index to the meta-data. The meta-data of unidentified multimedia content are then retrieved by computing a fingerprint and using this as a query in the fingerprint/meta-data database. The advantage of using fingerprints instead of the multimedia content itself is three-fold: 1. Reduced memory/storage requirements as fingerprints are relatively small; 2. Efficient comparison as perceptual irrelevancies have already been removed from fingerprints; 3. Efficient searching as the dataset to be searched is.

Audio fingerprinting is the process by which are able to identify some piece of music, given just a few seconds of input to go on. A fingerprinting system basically consists of two parts: fingerprint extraction and an algorithm to search for matching fingerprints in a fingerprint database. In an audio content, audio fingerprinting extracts some identifiable features, i.e., the fingerprint, from a piece of audio and stores it in a database. When the system is presented with an unidentified piece of audio, its fingerprint is extracted and matched against those stored in the database. Using fingerprints and matching algorithms, distorted version of a recording can still be identified as the same audio signal. The efficiency is related to the computational costs of both fingerprint extraction and search algorithms, the size of fingerprints and the query granularity. Compact fingerprint can reduce database storage, and moreover speed up the search, as most of the data can be stored in the main memory.

In audio fingerprinting system, accuracy is the most important requirement. It mainly depends on robustness of audio fingerprints and similarity measures. The characteristics of an AFP includes

- **Robustness:**
  Audio signals may be subject to a variety of signal degradations.
- **Compactness**
  Some applications need to store the AFP of every song from a possible big collection, other applications need to transmit the AFP over the internet.
- **Time Complexity**
  The AFP should be determined with as little computer effort as possible.
- **Granularity**
  Some Music Information Retrieval applications require the ability of identifying a song using only a small excerpt.
- **Scalability**
  The ability of an audio fingerprinting system to operate with large database.

Audio fingerprinting systems have been widely used in applications such as

- **Broadcasts Monitoring**
  The assessment of sponsorship effectiveness may be done by computers equipped with multi channel FM/TV cards.

- **Duplicate detection**
  Detecting duplicates is very important for maintaining the integrity of any multimedia database.
- **Automatic labeling**
  Modern MP3 players provide the user with tools for organizing songs which can be automatically filled using fingerprinting techniques.
- **Querying by example**
  A song may be identified using a small excerpt of audio captured by a mobile phone.
- **Filtering in p2p networks**
  When music is transmitted in a peer to peer network, the audio-fingerprint is determined from the packets and searched for in a list of copyrighted songs to prevent illegal copies.

2. Related Work

In the audio hashing approaches Haitsma and Kalker [1,2], the input signal is transformed into frequency domain using Fast Fourier Transform (FFT), and describe a system that uses the energies of 33 bark-scaled bands to obtain 32-bit sub-fingerprints which are the sign of the energy band differences(in both time and frequency axes). In [2], P. Shrestha and T. Kalker reported that when music is transmitted in a peer to peer network, the audio-fingerprint is determined from the packets and searched for in a list of copyrighted songs to prevent illegal copies. Musicbrainz [3] proposed automatic labeling that Modern MP3 players provide the user with tools for organizing songs, they rely in the contents of meta-data labels (e. g Album's title), when these labels are empty they can be automatically filled using fingerprinting techniques. J.S. Seo [4] presented an audio fingerprinting method based on the normalized spectral sub-band centroid (SSC). In [5,6], A. C. Ibarrola and E. Chavez proposed an audio fingerprinting method to link unlabelled audio to metadata such as the song's title or the singer's name, other uses of AFP's include duplicate detection in Multimedia Database and Monitoring Radio Broadcasts. S. Sukittanon and E. Atlas [7] showed that audio-fingerprinting systems extract the signal features in the frequency domain using a variety of linear transforms such as the Modulation Frequency Transform. Cano et al. [8] use Mel-Frequency Cepstrum Coefficient (MFCC) which is a widely used feature that closely approximates the human auditory system's response. Herre et al. [9] use Spectral Flatness Measure (SFM) which is an estimation of tone-like or noise-like quality for a band in the spectrum. Wang [10] generates fingerprints in the form of hash values of pairs of spectrum peaks. First, a constellation map is generated by spectrum peak detection on spectrogram. Then, each peak point is sequentially paired with points within its associated target zone. Finally, the two frequency values of point pair plus the time difference of this pair are hashed into a 32-bit unsigned integer. In [11], the feature vector sequences are converted into strings, and edit distance is applied. Among these similarity measures, some are sensitive to amplitude distortions, i.e., Euclidean distance; some are computational expensive, i.e., DTW and edit distance.

3. Background

3.1 Bark Scale

The human ear perceives better the lower frequencies than the higher ones. A basic feature of psychoacoustics is the concept of critical bands. It is assumed that the sound is analyzed in the hearing system by a bank of filters. When placing the critical bands next to each other, the critical band-scale or Bark-scale is produced. The Bark scale defines 25 *critical bands.* Equation (1) is used to convert Hertz to Barks.

$$z = 13\tan^{-1}\left(\frac{0.76\,f}{1000}\right) + 3.5\tan^{-1}\left(\frac{f}{7500}\right)^{2} \qquad (1)$$

where f is the frequency in Hertz. z is the frequency in Barks. An advantage of the Bark-scale is its linear relation to physiological features of the human hearing system, namely the length of the basilar membrane in the inner ear. The Bark scale ranges from 1 to 24 Barks, corresponding to the first 24 critical bands of hearing. The published Bark band edges are given in Hertz as [0, 100, 200, 300, 400, 510, 630, 770, 920, 1080, 1270, 1480, 1720, 2000, 2320, 2700, 3150, 3700, 4400, 5300, 6400, 7700, 9500, 12000, 15500]. The published band centers in Hertz are [50, 150, 250, 350, 450, 570, 700, 840, 1000, 1170, 1370, 1600, 1850, 2150, 2500, 2900, 3400, 4000, 4800, 5800, 7000, 8500, 10500, 13500].

3.2 Entropy of a signal

The entropy of a signal is a measure of the amount of information the signal carries. Entropy (or an entropy-based feature) can be computed from any finite set of values, e.g. a parametric vector, a discrete spectral density estimate, or directly from a segment of a digital signal.

Shannon's entropy is computed using Eq (2) and its continuous version called "differential entropy".

$$H(x) = -\sum_{i=0}^{n} p_i \ln(p_i) \qquad (2)$$

where p_i is the probability for any sample of the signal to adopt value ***i*** being ***n*** the number of possible values.

By processing an audio-signal in frames of computing Shannon's entropy every frame, a sequence of entropy values could be obtained. This sequence is referring to the entropy curve.

4. Proposed Method

The audio fingerprinting method presented in this paper has three aspects: extraction of audio features, modeling of representative audio features into finger codes and matching a query in the database using a distance measure. In this section, details of the proposed method are presented.

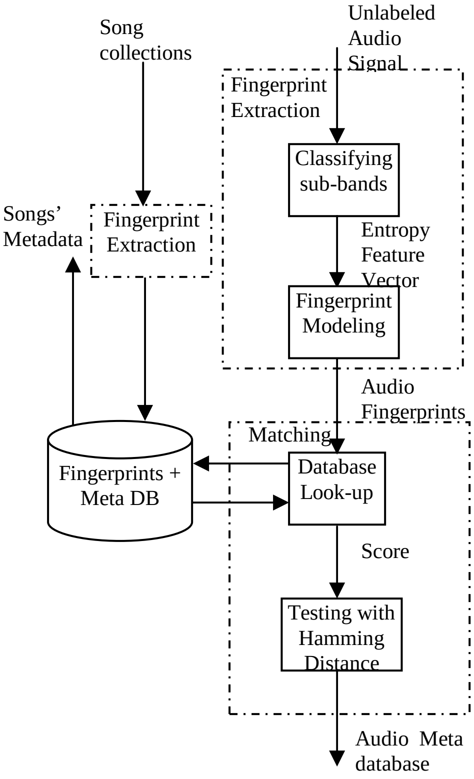

Figure1. Framework of audio fingerprinting system

4.1 Audio Fingerprint (AFP) Extraction

Stereo audio signals of .mp3 file format are first converted to .wav monaural of sampling frequency of 44100 Hz with 16 bits resolution. The signal is processed in frames of 370 ms. Frames are overlapped fifty percent for the spectral entropy sequence extraction. To each frame the Hann window is applied and then its DFT is determined. Using Eq. (2) Shannon's entropy is computed for the first 24 critical bands according to the Bark scale, discarding only the 25th critical band.

4.1.1 Computing the Entropy Signal

For any given band *b*, the elements of the DFT corresponding to *b* are used to build two histograms, one for the real parts and another one for the imaginary parts. The histograms are used to estimate the probability distribution functions. Shannon's entropy for the real (h_{br}) and imaginary parts (h_{bi}) of the DFT are computed separately. The entropy h_b for band *b* is determined as the sum of (h_{br}) and (h_{bi}).

4.3 Modeling of Audio Finger Codes

For each frame of the audio signal a vector with 24 values of spectral entropy is obtained. The sequence of vectors corresponding to a short excerpt of audio of a few seconds make a matrix 24 rows and a number of columns that depends on the duration of the excerpt. The finger codes of a particular frame are expressed as the difference between adjacent bands using Eq. (3).

$$F(n,b) = \begin{cases} 1 & if \quad [h_b(n) - h_b(n-1)] > 0 \\ 0 & otherwise \end{cases} \qquad (3)$$

4.4 Matching with Hamming Distance

As a distance measure for the spectral entropy sequence of any two songs, the Hamming distance is be used. The Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. It measures the minimum number of substitutions required to change one string into the other, or the number of errors that transformed one string into the other. For example, the Hamming distance between: **1011101** and **1001001** is 2.

5. Evaluation Experiments

Two types of experiments are performed in this experimental study: Experiment I and Experiment II. In Experiment I, song identification with no degraded query songs is performed. Look-ups in the database with degraded versions of query songs are carried out in Experiment II.

5.1 Music Database

The database we used in experiments includes 150 songs grouped by 8 genres: classical, country, disco, hiphop, jazz, metal, pop, and rock. These songs, recorded at 44 kHz sampling rate, with 16 bit resolution with .mp3 file format.

5.2 Performance Metrics

In evaluating the search results, two criteria are used for measuring the performance of the proposed audio fingerprinting method. Common measurements used in audio fingerprinting work are correct identification which is related to percentage of true positives and incorrect identification related to number of false search results.

5.3 Results

5.3.1 Experiment I

Queries of 5s long music excerpts are drawn at random offset from 150 songs. In this experiment, 25 queries of music excerpts have been made. 100% correct identification rate is resulted.

5.3.2 Experiment II

Songs are degraded into different ways: low pass filtering, mixing with noise, mp3 attacking and resampling. For every degradation attack, queries of 5s music fragments are also drawn at random offsets. They are

- low-pass filtering with two cutoff frequencies
- noise addition with 4 levels of SNRs
- resampling into two sampling frequencies,
- mp3 compression into two bit rates

In the following result tables are measured from 25 runs where test songs and their offsets are randomly chosen.

Low-pass filtering

In this experiment, two different cutoff frequencies have been used and result are shown in Table 1. In both filtering attacks, the correct and incorrect identification rates are 96% and 4%, respectively.

Table 1: Accuracy on Low-pass filtering

Cutoff Frequency	Correct Identification	Incorrect Identification
22 kHz	96%	4%
11 kHz	96%	4%

Noise Addition

Here, white noise is mixed to the songs to have a Signal to Noise Ratio (SNR) at 4 different levels: 20 dB, 15 dB, 10 dB, and 5 dB. Table 2 displays the recognition rate of the queries in the database. The SNR is computed using Eq. (4) where ***P* (signal)** is the power of the original signal and ***P* (noise)** is the power of the noise added to the signal.

$$SNR = 20 \log_{10}\left(\frac{p_{signal}}{p_{noise}}\right) \qquad (4)$$

Table 2: Accuracy on Noise Mixing

SNR	Correct Identification	Incorrect Identification
20 dB	100%	0%
15 dB	92%	8%
10 dB	84%	16%
5 dB	52%	48%

Resampling

Original sampling rate of 44100 Hz is converted into lower sampling frequencies: 22 kHz and 16 kHz. No error identification result is found at 25 runs at random selection of starting points.

Table 3: Accuracy on Resampling

Sampling frequency	Correct Identification	Incorrect Identification
22050 Hz	100%	0%
16000 Hz	100%	0%

Lossy Compression

In this test, two bit rates of lossy compression to MP3 are used: a bit rate of 64 kbps and 96 kbps. Zero incorrect identification is also found at this distortion attack as it is shown at Table 4.

Table 4: Accuracy on MP3 Lossy Compression

Bit rate	Correct Identification	Incorrect Identification
96 Kbps	100%	0%
64 Kbps	100%	0%

Bit Error Rate (BER)

The bit error rate between the extracted fingerprint and the fingerprint in the database is used as the similarity measure or threshold value. It is computed as

$$BER = \frac{no.\ \ of\ \ error\ \ bits}{total\ \ no.\ \ of\ \ bits\ \ in\ \ query} \qquad (5)$$

In Table 5, the bit error rates which are used to identify the queries in meta-database at different tests described above are summarized.

Table 5: Bit Error Rates Employed

Tests	BER
Experiment I	0

MP3 (96kbps)	0.2431
MP3 (64kbps)	0.2736
Low-pass (22kHz)	0.2880
Low-pass (11kHz)	0.3203
Noise (20dB)	0.2806
Noise (15dB)	0.3256
Noise (10dB)	0.3718
Noise (5dB)	0.4088
Resample (22050 Hz)	0.0385
Resample (16000 Hz)	0.0813

6. Discussion

In testing with a database of 150 songs, four different degraded versions (resampling, mixing with white noise, low pass filtering, and mp3 compression) have been made. In dealing with different degradation issues, this audio fingerprinting system is most resistant to resampling and conversion into mp3 compression standards. According to the experiments on low-pass filtering distortion, satisfactory correct identification rates are obtained. Highest incorrect recognition rates are found at noise adding attack. To have the best match song, bit error rates (BER) is used as the tolerance. Higher bit error rates are needed for retrieving songs with noise attack/resampling attack whereas lower bit error rates can be set in searching of lossy compressed/low-passed audio file. In determining the finger code extraction time, it takes about 20 percent of the whole duration of the song. When queries of 5s long have been made at random offsets, the reply to this query can be made within 1.3 s in average. Therefore, this system has low complexity in query extraction.

7. Conclusion

A highly robust audio-fingerprint (AFP) is developed based on entropy named as the *spectral entropy sequence*. To extract the finger codes of a song, Shannon's entropy is determined from the spectral coefficients of each one of the first 24 critical bands according to the Bark scale. The spectral entropy sequence developed in this study has proved to be highly robust to heavy degradations of the audio signals. The time it takes to determine existence of a song is approximately 25 percent of the duration of the query when it is ran on Matlab environment. This parameter is important for real time applications where fast search speed is essential. The results proved that the algorithm is resistant to noise and other distortions, computationally efficient and can quickly identify a query out of a data.

8. References

[1] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *International Symposium on Music Information Retrieval ISMIR*, 2002, pp. 144-148.

[2] P. Shrestha and T. Kalker, "Audio fingerprinting in peer-to-peer networks," *5th International Retrieval(ISMIR)*, 2004.

[3] Musicbrainz, [Online]. Available: ftp://ftp.musicbrainz.org/pub/musicbrainz/ 2002.

[4] O. Hellmuth, E. Allamanche, M. Cremer, T. Kastner, C. Neubauer, S. Schmidy, and F. Siebenhaar, "Content-based broadcast monitoring using mpeg-7 audio fingerprints," in *International Symposium on Music Information Retrieval (ISMIR)*, 2001.

[5] J.S. Seo, M. Jin, S. Lee, D. Jang, S. Lee, and C. D. Yoo, "Audio fingerprinting based on normalized spectral sub-band moments," in *International Conference on Acoustics, Speech, and Signal Processing* (*ICASSP*), April 2006.

[6] A. C. Ibarrola and E. Chavez, "A robust entropy-based audio-fingerprinting," *IEEE International Conference on Multimedia and Expo* (*ICME2006*), July 2006, pp. 1729-1732.

[7] S. Sukittanon and E. Atlas, "Modulation frequency features for audio fingerprinting," *IEEE International Conference on Acoustics, Speech and Digital Processing* (ICASSP), University of Washington USA, 2002, pp. II 1773-1776.

[8] P. Cano, E. Batlle, H. Mayer, and H. Neuschmied, "Robust sound modeling for song detection in broadcast audio," In *Proc. AES 112th Int. Conf*, Munich, Germany, May 2002.

[9] J. Herre, E. Allamanche, and O. Hellumth, "Robust matching of audio signals using spectral flatness features," In *Proc. IEEE Workshop Applications Signal Processing Audio Acoustics*, 2001, pp. 127-130.

[10] A Wang, "An industrial strength audio search algorithm," In *Proc. 4th Int. Conf. Music Information Retrieval (ISMIR)*, 2003.

[11] E. Weinstein and P. Moreno. "Music identification with weighted finite-state transducers," In *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Hawaii, 2007.