Framework for Audio Fingerprinting based on Discrete Wavelet Entropy

Nu War
University of Computer Studies, Yangon
nuwar81@gmail.com

Abstract

At the core of the presented system is a highly robust fingerprint extraction method which enables searching a large fingerprint database with only limited computing resources. Requirements for such systems include robustness to a wide range of signal distortions and availability of fast search methods, even for large fingerprint databases. In this paper an audio fingerprinting system is presented for song identification. For the high dimensional audio fingerprint data, audio fingerprint searching algorithm were proposed: an audio fingerprinting method based on DWE (Discrete wavelet entropy) with timbral features (MFCC and FFT) and an efficient indexing method for Audio fingerprint database using the filtering approach, known also as vector approximation approach which supports the nearest neighbor search efficiently. Spectral subband entropy is selected due to its resilience against equalization, compression, and noise addition. Region Approximation Blocks divides a high-dimensional feature vector space into compact and disjoined regions. Each region will be approximated by two bit-strings according to the RA-Blocks technique.

1. Introduction

Recent years have seen a growing scientific and industrial interest in computing fingerprints of multimedia objects. By using the fingerprint of an unknown audio clip as a query on a fingerprint database, which contains the fingerprints of a large library of songs, the audio clip can be identified. 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.

A fingerprinting system needs to have the following properties: robustness, reliability, compactness and scalability and search speed.

The robustness indicates that the fingerprinting system can resist various common audio distortions. The fingerprints of a degraded audio clip should be similar to the fingerprints of the original audio clip.

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.

The search speed indicates how fast a fingerprint can be found in a large fingerprint database.

The versatility indicates that ability to identify audio regardless of the audio format.

Fingerprinting can extract information from the audio signal at different abstraction levels, from low level descriptors to higher level descriptors. Especially, higher level abstractions for modeling audio hold the possibility to extend the fingerprinting usage modes to content-based navigation search by similarity, content-based processing and other applications of Music Information Retrieval. Adapting existing efficient fingerprinting systems from identification to similarity browsing can have a significant impact in the music distribution industry (e.g.: www.itunes.com, www.mp3.com). At the moment, on-line music providers offer searching by editorial data (artist, song name, and so on) or following links generated through collaborative filtering.

A way to efficiently implement the direct file comparison idea consists in using a hash method, such as MD5 (Message Digest 5) or CRC (Cyclic Redundancy Checking), to obtain a compact representation of the binary file. In this setup, one compares the compact signatures instead of the whole files. Of course, this approach is not robust to compression or distortions of any kind, and might not even be considered as content-based identification of audio, since it is not based on an analysis of the content (understood as perceptual information) but just on manipulations performed on binary data. This approach would not be appropriate for monitoring streaming audio or analog audio; however, it sets the basis for a class of fingerprinting methods: Robust or

Perceptual hashing [7] [10]. The idea behind robust hashing is the incorporation of acoustic features in the hash function, so that the final hash code is robust to audio manipulations as long as the content is preserved. Many features for audio characterization can be found in the literature, such as energy, loudness, spectral centroid, zero crossing rate, pitch, harmonicity, spectral flatness and Mel-Frequency Cepstral Coefficients (MFCC's). Several methods perform a filter bank analysis, apply a transformation to the feature vector and, in order to reduce the representation size, extract some statistics: means or variances over the whole recording, or a codebook for each song by means of unsupervised clustering.

2. Related work

The first thing an audio-fingerprinting system has to do is to extract features from the signal. The module in charge of extracting relevant perceptual features of the audio signal is known as the front end, once this module delivers the features the signal, the audio finger printing system models the songs in a way that best serve the purpose of the application for which it has been designed. Some Audio fingerprinting models are listed below:

• Sequences of Feature Vectors. This kind of audio fingerprints are also known as trajectories or traces. The features extracted at equally spaced periods of time are simply stored in a list of vectors or in a table, one row per frame. An example of this kind of audio fingerprinting is the binary vector sequence described in [7].

• Statistics Instead of storing every feature vector, only statistical data over the set of feature vectors are stored. The audio-fingerprint designed for MPEG-7 [9] computes the means, variances, minimum and maximum values every 32 frames. The minimum and maximum values are used for delimiting the search and the means and variances are used for the actual search using some measure like the Mahalanobis' distance.

• Codebooks The sequence of feature vectors extracted from a song is replaced by a small number of representative code vectors stored in a codebook, which from then on represents the song. This model disregards the temporal evolution of the audio signal.

• Strings. Trajectories can be converted into long strings of integers using vector quantization. This model allows the treatment of the songs as texts that can be compared using flexible string matching techniques [1].

• Single vectors. These are the smallest audio fingerprints, they are usually built with average features extracted from the whole song, for example, an audio fingerprinting can be a vector containing the beats per minute, the average zero crossing rate and the average spectrum .

• Hidden Markov Models (HMM). These finite state machines model non stationary stochastic processes (e.g. songs). For each song of the collection a HMM is built. The features extracted from the test's song are considered to be a sequence of acoustic events and then used as the input for the candidate's HMM. The candidate's HMM in turn reports the probability that the test song matches the candidate song, this probability is used as a proximity measure for choosing the right song [3].

Audio-fingerprinting systems extract features from the signal normally on a frame by frame basis. Most systems extract the signal features in the frequency domain using a variety of linear transforms such as the Discrete Cosine Transform, the Discrete Fourier Transform, the Modulation Frequency Transform [12] and some Discrete Wavelet Transforms like Haar's and Walsh-Hadamard's [13]. Early work on audio-fingerprinting inherited the benefits from decades of research in speech processing. When more relevant features of music are finding, a variety of perceptual variables have been used such as Loudness (PL) [4], the Joint Acoustic and Modulation Frequency (JAMF) [12], the Spectral Flatness Measure (SFM) [6], the Spectral Crest Factor (SCF) [6], Spectral Subband Centroids (SSC) [8]. In [8] it is shown how the Normalized SSC is more robust than MFCC and tonality for lossy compression and equalization. In [12] it is reported that the Normalized JAMF has superior robustness than a spectral estimate for compression and equalization.

3. Background

Daubechies Wavelet Coefficient Histograms (DWCH) is introduces for music feature extraction for music information retrieval. The histograms are computed from the coefficients of the db_8 Daubechies wavelet filter applied to 3 s of music. A comparative study of sound features and classification algorithms on a dataset compiled by Tzanetakis shows that combining DWCH with timbral features (MFCC and FFT), with the use of multiclass extensions of support vector machine, achieves approximately 80% of accuracy, which is a significant improvement [15]. And then entropy wavelet can give the reasonable result, the result of variety of different WT families (db4, ciof4, bior3.7 and sym4) for GEA types in wavelet entropy analysis of 10 GEA signals in normal and FGD is (87.5%, 86.1%, 89.7%, 89.2%)[16].

New techniques based on the approximation vector approach appeared to provide a solution to the dimensionality curse. VA-file (Vector Approximation File) [11, 2] is the first method based on this approach. The basic idea of the VA-File is to keep two files; one contains the exact representation (original vectors), the other, and relatively smaller,

has geometrical approximation for each vector. When searching vectors, the entire approximations file is scanned to select candidate vectors. Upper and lower bounds of the distance to the query are computed for each vector. These bounds frequently suffice to filter most of the vectors. Those candidates are then verified by visiting the original vectors file. The sequential scan is done on the smaller approximations file; it is very fast and allows reading just a few vectors from real database. This process decreases the number of I/O operations and the CPU cost compared with the sequential method that analyzes the totality of the database. Like the VA-File, the LPC-File (Local Polar Coordinate File) [5] is based on the approximate approach. Thus, the vector space is partitioned into rectangular cells which are used to generate bit-encoded approximations for each vector. Cha [5] noticed that the performances of the VA-File can be improved only by increasing the number of bits for approximations.

However, the performance of the VA-File decreases when the capacity of database becomes very large. Thereafter, the approximations file cannot be placed entirely in the memory. Thus, the RA-Blocks technique (Region Approximated Blocks) [5] is proposed. It divides the vectors space into regions containing each one a set of cells, in order to approximate each region by two string-bits. This reduces the computation time of the VA-File and also optimizes the page replacement to further reduce the number of I/O access as well. Moreover, the bounds (upper and lower) are calculated for each region and not for each vector in order to reduce the CPU cost.

4. Proposed audio-fingerprinting system

A musical knowledge database is needed to build. This database contains the fingerprints of all the songs the system is supposed to identify. The fingerprint of the input signal is calculated and a matching algorithm compares it to all fingerprints in the database during detection. The knowledge database must be updated as new songs come out.

A fingerprinting system used for identification is generally made up of two steps: fingerprint extraction and fingerprint matching. Query fingerprints are extracted from an audio clip that is to be identified. Then the candidates for the query fingerprints are obtained by the fast matching. As shown in figure.1, the overall system architecture of proposed system can be seen with the fingerprint extraction step and matching step.

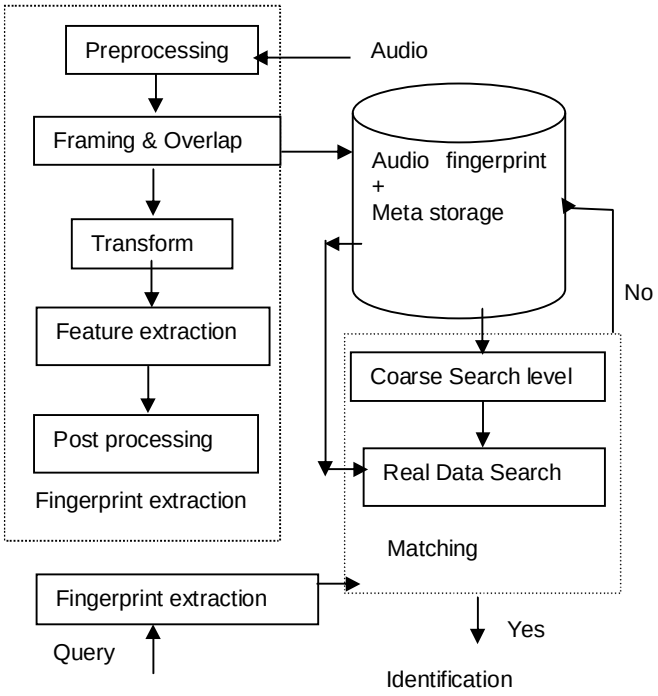

Figure.1: General architecture of proposed system

4.1 Feature extraction

Searching for features in audio signals that would still be present if those audio signals are severely degraded, the use of entropy is explored for audio fingerprinting purposes. In search for a technique that wouldn't be restricted to stationary or quasi stationary signals but would work even if the signal has a fractal structure an Information theoretic based audio fingerprinting is designed, this audio fingerprinting was also motivated by the intuition that is information what the human brain really perceives.

4.1.1 Calculation of spectral entropy

Wavelets are designed to give good time resolution at high frequencies and good frequency resolution at low frequencies. They have several favorable properties, including compact support, vanishing moments and decorrelated coefficients and have been successfully applied in signal representation and transformation. The entropy-based approach is more reliable than pure energy-based methods in some cases, particularly when noise-level varies with time. Since the frequency energy of different types of noise focus on different frequency subbands, the effect of corrupted noise on each frequency subband is different. The human ear perceives better the lower frequencies than the higher ones. Critical subband is widely used in perceptual auditory modeling. A Bark-scale wavelet

decomposition is used to decompose the signal into 24 critical wavelet subband signals, and it is implemented with an efficient five-level tree structure that implemented with the Daubechies family wavelet, with downsampling by 2.

Spectral energy : For the m^{th} frame, the spectral energy of the ξ^{th} subband is evaluated by the sum of squares:

$$E(\xi, m) = \sum_{\omega_{\xi,l}}^{\omega_{\xi,h}} |X(\omega, m)|^2$$

Where X (ω, m) means the ω^{th} wavelet coeffience. $\omega_{\xi, l}$ and $\omega_{\xi, h}$ denote the lower boundaries and the upper boundaries of the ξ^{th} subband, respectively. Sort the frequency subband according to their E (ξ, m) value. That is,

E (I_1, m) ≥ E (I_2, m) ≥ E (I_3, m) ≥ … E (I_N, m),

where I_i is the index of the frequency subband with the i^{th} max energy.

Spectral entropy: To calculate the spectral entropy, the probability density function (pdf) and the entropy calculation are both necessary steps. The pdf for the spectrum can be estimated by normalized the frequency components:

$$P(\xi, m) = \frac{E(\xi, m)}{\sum_{\omega=1}^{N_{ub}} E(\omega, m)}$$

where N_{ub} is the number of useful frequency subbands. Having finishing applying the above constraints, the spectral entropy H (m) of frame m can be defined below.

$$H(m) = -\sum_{\xi=1}^{N_{ub}} P(\xi, m).log[P(\xi, m)]$$

4.1.2 Timbral textual features

Timbral textual features are those used to differentiate mixture of sounds based on their instrumental compositions when the melody and the pitch components are similar. The use of timbral textural features originates from speech recognition.

Extracting timbral features require preprocessing of the sound signals. The signals are divided into statistically stationary frames, usually by applying a window function at fixed intervals. The application of a window function removes the so-called "edge effects." Popular window functions including the Hamming window function and the Blackman window function.

a. **MFCC** -> it is obtained as follows: each frame is computed firstly, the logarithm of the amplitude spectrum based on short-term Fourier transform, where the frequencies are divided into thirteen bins using the Mel-frequency scaling. (The "cepstrum" is the FFT of this logarithm.). Then discrete cosine transform is applied to de-correlate the Mel-spectral vectors. In this study, the first five bins are used, and compute the mean and variance of each over the frames.

b. **Short-Term Fourier Transform Features** -> this is a set of features related to timbral textures and is not captured using MFCC. It consists of Spectral Centroid, Spectral Rolloff, Spectral Flux and Low Energy, Zero Crossings and then computes the mean for all five and the variance for all but zero crossings. So, there are a total of nine features.

4.2 Audio fingerprint matching algorithm

A K-nearest neighbor queries algorithm of proposed system is inspired of the VA-NOA search algorithm in the VA-File [11]. The research algorithm for K-NN is divided into two phases.

Algorithm

Input: query and database vectors
Output: NNQ containing nearest neighbors // NNQ is a list containing the nearest neighbors found so far, sorted according to ascending order of exact distance from a query point.

1. Calculate or load (if pre-computed) the approximation of database vectors : ap_v and query point : ap_q;

2. Initialize NNQ;

3. Max_db = maximum possible value;

4. For each approximation ap_v of database vectors {
  Current_bd = block distance between ap_v and ap_q;
  If (current_bd < max_db) {
  Calculate the corresponding actual distance;
  Insert this vector into NNQ, if it is closer to query point than the last element in NNQ;
  Update max_db to be the distance from query point to the last element of NN found so far;
}
}

1) Coarse search level: the approximations file is scanned sequentially, and the candidates regions are selected.
2) Real data search level: calculating the real distance from vectors of candidates regions to the query vector.

5. Conclusion

The fingerprint and the computation thereof both need to satisfy several conditions. This system hopes

to allow for both reliable recognition of real-world audio signals and real-time operation on today's standard PC computing platforms. An important feature is the robustness of the system: an audio clip must be identifiable even after severe signal degradation. Reliability is another issue that should be addressed. Also, the size of the fingerprint should be kept to a minimum. This will be especially important when we want to search the database later on to find a match. The algorithm is based on bark-scale wavelet decomposition to decompose the input speech signal into critical sub-band signals for audio recognition. This entropy based audio fingerprinting can be used when dealing with large databases since an even more detailed representation of the songs may slow down the matching process. For the robustness experiment, it will be tested with the signal degradations such as MP3 compression, Real Media (RM) compression, echo addition, noise addition, equalization, low-pass filtering, and resampling.

6. References

[1] A. Y. Guo and S. Hava, "Time-warped longest common subsequence algorithm for music retrieval," in 5th International Conference on Music Information Retrieval (ISMIR), 2004.

[2] D. R. Heisterkamp and J. Peng, "Kernel Vector Approximation Files for Relevance Feedback Retrieval in Large Image Databases," Multimedia Tools and Applications, vol 25., N° 2, pp. 175-189, June, 2005.

[3] E. Batlle, J. Masip, and E. Guaus, "Amadeus: a scalable hmm-based audio information retrieval system," in First International Symposium on Control, ommunications and Signal Processing, March 2004, pp. 731– 734.

[4] E. Zwicker and H. Fastl, Psycho-Acoustics. Facts and Models. Springer, 1990.

[5] G.–H. Cha, X. Zhu, D. Petrovic, "An Efficient Indexing Method for Nearest neighbor searches in High-Dimensional Image Databases," IEEE transactions On Mutimedia, Vol 4, N°1, pp 76-87, March 2002.

[6] J. Herre, E. Allamanche, and O. Hellmuth, "Robust matching of audio signals using spectral flatness features," IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 127–130, 2001.

[7] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in International Symposium on Music Information Retrieval ISMIR, 2002.

[8] J. S. Seo, M. Jin, S. Lee, D. Jang, S. Lee, and C. D. Yoo, "Audio fingerprinting based on normalized spectral subband centroids," in International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2005.

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

[10] R. Venkatesan (2001) A perceptual audio hashing algorithm: a tool for robust audio identification and information hiding. 4th Int. Information Hiding Workshop, Pittsburg, PA

[11] R. Weber, H.-J. Schek, S. Blott, A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Space. Proceedings of the 24th VLDB Conference, USA, 1998.

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

[13] S. Subramanya, R. Simha, B. Narahari, and A. Youssef, "Transform-based indexing of audio data for multimedia databases," in International Conference on Multimedia Applications, 1999.

[14] T. Chen, M. Nakazato, T. S. Huang, Speeding up the Similarity Search in multimedia Database. In Proceedings of IEEE ICME, 2002.

[15] T. L. Ogihara, et .al *Toward intelligent music information retrieval*. 15 May 2006 Multimedia, IEEE Transactions on page: 564-574, volume 8, issue: 3.

[16] Y. Z. Mehran1, A. M. Nasrabadi2 "Wavelet Entropy Analysis of GEA Signal to Abnormal Pattern Recognition" http://www.chaos2008.net/