Audio Retrieval using Hash-Index SearchTechnique

Mahesh B. Sonje[1], Prof. S.M. Rokade[2]

[1]*(Computer Engineering, S.V.I.T / SavitribaiPhuleUniversity of Pune, India)*
[2]*(Computer Engineering,S.V.I.T / SavitribaiPhuleUniversity of Pune, India)*

ABSTRACT ***:*** *An audio fingerprint is one of the generated content-based signature from original audio file that refers to an audio recording. Audio Fingerprinting technique is commonly used to monitor the audio independently without the need of meta-data. Different fingerprinting approaches are Pattern matching, Multimedia Information Retrieval and Cryptography. For browsing and identifying the audio content from large digital music audio data collection set is a major strand for research.The best solution for identifying the audio content is "Audio ID System" in which the specified audio is identified by passing a short query audio clip from a large set of audio data set. The popular audio fingerprinting schemes with a common audio fingerprinting framework with short query captured from microphone, are surveyed. The audio system originally developed by Haitsma and Kalker is most widely used and it proposes several modifications and improvements with to increase audio retrieval speed and storage requirements.*

KEYWORDS -*Audio identification, content-based retrieval, indexing, music.*

Introduction

Audio fingerprinting is best known for its ability to link unlabelled audio to corresponding metadata (e.g. artist and song name), regardless of the audio format. Although there are more applications to audio fingerprinting, such us: Content-based integrity verification or watermarking support, this review focuses primarily on identification. Audio fingerprinting or Content-based audio identification (CBID) systems extract a perceptual digest of a piece of audio content, i.e. the fingerprint and store it in a database. When presented with unlabelled audio, its fingerprint is calculated and matched against those stored in the database. Using fingerprints and matching algorithms, distorted versions of recording can be identified as the same audio content.A source of difficulty when automatically identifying audio content derives from its high dimensionality and the significant variance of the audio data for perceptually similar content. The simplest approach that one may think of – the direct comparison of the digitalized waveform – is neither efficient not effective. An efficient implementation of this approach could use 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 hash values instead of the whole files. However, hash values are fragile, a single bit flip is sufficient for the hash to completely change. Of course this setup is not robust to compression or minimal distortions of any kind and, in fact, it cannot be considered as content-basedidentification since it does not consider the content, understood as information, just the bits.

An ideal fingerprinting system should fulfil several requirements. It should be able to accurately identify an item, regardless of the level of compression and distortion or interference in the transmission channel. Depending on the application, it should be able to identify whole titles from excerpts a few seconds long (property known as granularity or robustness to cropping), which requires methods for dealing with shifting, that is lack of synchronization between the extracted fingerprint and those stored in the database. It should also be able to deal with other sources of degradation such as pitching (playing audio faster or slower), equalization, background noise, D/A-A/D conversion, speech and audio coders (such as GSM or MP3), etc. The fingerprinting system should also be computationally efficient. This is related to the size of the fingerprints, the complexity of the search algorithm and the complexity of the fingerprint extraction. The design principles behind audio fingerprinting are recurrent in several research areas. Compact signatures that represent complex multimedia objects are employed in Information Retrieval for fast indexing and retrieval. In order to index complex multimedia objects it is necessary to reduce their dimensionality (to avoid the "curse of dimensionality") and perform the indexing and searching in the reduced space. In analogy to the cryptographic hash value, content-based digital signatures can be seen as evolved versions of hash values that are robust to content-preserving transformations. Also from a pattern matching point of view, the idea of extracting the essence of a class of objects retaining the main its characteristics is at the heart of any classification system.

Related Work

J. Haitsma and T. Kalkerpresented a new approach to audio fingerprinting. The fingerprint extraction is based on extracting a 32 bit sub fingerprint every 11.8 milliseconds. The sub-fingerprints are generated by looking at energy differences along the frequency and the time axes. A fingerprint block, comprising 256 subsequent sub-fingerprints, is the basic unit that is used to identify a song. The fingerprint database contains a two-phase search algorithm that is based on only performing full fingerprint comparisons at candidate positions pre-selected by a sub-fingerprint search with reference to the parameters like Robustness, Reliability, Fingerprint size, Granularity, Search speed and scalability[1].

A.Wang processes the fingerprints from the unknown sample and matched with a large set of fingerprints derived from the music database. The candidate matches are subsequently evaluated for correctness of match. Some guiding principles for the attributes to use as fingerprints are that they should be temporally localized, translation-invariant, robust, and sufficiently entropic. The temporal locality guideline suggests that each fingerprint hash is calculated using audio samples near a corresponding point in time, so that distant events do not affect the hash[2].

H. Schreiber, P. Grosche, and M. Muller presented an optimization scheme of the Haitsma/Kalker audio fingerprinting search algorithm. The suggested approach exploits strong temporal correlations between sub-prints as an indicator for sub-print robustness. This can lead to significant savings in the number of required lookups leading to a significant overall speed-up for the identification task[3].

P. Grosche, M. Müller, and J. Serra presented three representative retrieval strategies based on the query. (a)Traditional retrieval using textual metadata (e. g., artist, title) and a web search engine. (b)Retrieval based on rich and expressive metadata given by tags. **(c)** Content-based retrieval using audio, MIDI, or score information. Such content-based approaches provide mechanisms for discovering and accessing music even in cases where the user does not explicitly know what is actually looking for. Such approaches complement traditional approaches that are based on metadata and tags[4].

F. Kurth and M. Muller proposed a novel index-based audio matching algorithm, which allows for identifying and retrieving musically similar audio clips irrespective of a specific interpretation or arrangement—a task which previous audio identification algorithms cannot cope with. First absorb a high degree of the undesired variations at the feature level by using chroma-based audio features. To cope with global variations in tempo and pitch, multiple queries strategy is employed. Finally, introduced a further degree of robustness into the matching process by employing fuzzy and mismatch search. The combination of these various deformation- and fault-tolerance mechanisms allowed to employ standard indexing techniques to obtain an efficient as well as robust overall matching procedure, which can cope with significant, musically motivated variations that concern tempo, articulation, dynamics, and instrumentation[5].

D. P. Ellis and G. E. Poliner present a method for identification of cover tracks (when different musicians perform the same underlying song or piece, these are known as ‘cover’ versions) from music audio databases with two features i.e. Beat Tracker and Chroma feature. Beat tracker generate a beat-synchronous representation with one feature vector per beat. The representation of each beat is a normalized chroma vector. It provides the most efficient coverage of large music databases. These can then be used as ‘index terms’ to permit the use of more rapid indexing schemes, as well as potentially revealing interesting repeated and shared structure within music collections[6].

F. Kurth, A. Ribbrock, and M. Clausen present a new method for fast index-based robust identification of highly distorted audio w.r.t. large databases. It uses a very simple but yet robust preprocessing methods combined with simple error correcting codes and methods from multirate signal processing yields suitable feature extractors. The resulting features used with fast indexing and search algorithms which supports for large database application [7].

Proposed System Architecture

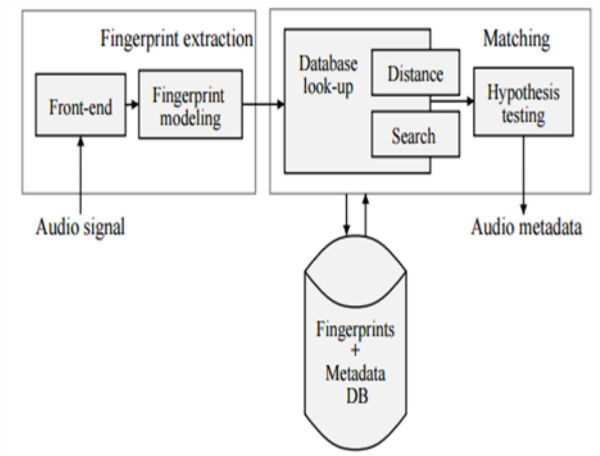

Figure 1. Framework for Audio Fingerprinting

Figure 1 shows the fingerprint extraction and a fingerprint modeling block. The fingerprint extraction consists of a front-end and a fingerprint modeling block. The front-end computes a set of measurements from the signal. The fingerprint model block defines the final fingerprint representation, e.g: a vector, a trace of vectors, a codebook. a sequence of indexes to HMM sound classes, a sequence of error correcting words or musically meaningful high-level attributes. Given a fingerprint derived from a recording, the matching algorithm searches a database of fingerprints to find the best match. A way of comparing fingerprints that is a distance is therefore needed. Since the

number of comparison is high and the distance can be expensive to compute, It is common to see methods that use a simpler distance to quickly discard candidates and the more correct but expensive distance for the reduced set of candidates. There are also methods that pre-compute some distances off-line and build a data structure that allows reducing the number of computations to do on-line.

FRONT-END of SYSTEM

The front-end converts an audio signal into a sequence of relevant features to feed the fingerprint model block.

Preprocessing

In this step, the audio is digitalized (if necessary) and converted to a general format Framing & Overlap signal is divided into frames of a size comparable to the variation velocity of the underlying acoustic events.

Framing and Overlap

A key assumption in the measurement of characteristics is that the signal can be regarded as stationary over an interval of a few milliseconds. Therefore, the signal is divided into frames of a size comparable to the variation velocity of the underlying acoustic events. The number of frames computed per second is called frame rate. A tapered window function is applied to each block to minimize the discontinuities at the beginning and end. Overlap must be applied to assure robustness to shifting (i.e. when the input data is not perfectly aligned to the recording that was used for generating the fingerprint). There is a trade-off between the robustness to shifting and the computational complexity of the system: the higher the frame rate, the more robust to shifting the system is but at a cost of a higher computational load.

Linear Transforms: Spectral Estimates

The idea behind linear transforms is the transformation of the set of measurements to a new set of features.

Feature Extraction

In this step, propose system find a great diversity of algorithms. The objective is again to reduce the dimensionality and, at the same time, to increase the invariance to distortions.

Post-processing

Propose system uses low resolution quantization to the feature extraction.

Mathematical Model

Mathematical Model using Set Theory

Let S be the system of

S= { I,O,P,F,A,Pa,T,Pp }

Where,

I = set of Input

O = set of output

P = is a set of preprocessing function
P={P1,P2,……,Pn}

F = is a set of frames of audio
F={F1,F2,…….,Fn}

A = set of Audio File

Function(f1) : function f1 gets the audio input and starts preprocessing algorithm.

F(f1)={A1,A2,……,An} → {P1,P2,…..,Pn}tp

Function(f2) :function f2 apply A/D conversion, sampling, normalization on audio file and gives Pa as output.

Pa is the audio file after preprocessing

F2(A1)={a1,a2,…..,an}→Pa1,Pa2,,Pan}→ Pa

Function(f3) : function f3 get input and preprocess audio file and apply HarrHandmard transformation and produced T as output.

T = Tranform audio

F(f3)←{Pa1,Pa2,.., Pan} → {t1,t2,….,tn}→ T

Function(f4) : this function accepts the feature extract and post processing.

F4(T) = { t1,t2,…,tn} → Pp

Pp is the post procen audio

Function(f5) : this function calculates Robust hash of given Pp input and stored in location by applying histograms, error correction, quantization , binary sequence generation, frequency modulation

F5(Pp) ← {Pp → Fing }

Fing = is the audio fingerprint

Function(f6) : this function is used to search the audio fingerprint

Input = audio signal

Output = its location in Hash Table

F6(A) → {P,Pa,T,Pp,Fing} → Hash table location

Conclusion

We perform the survey and evaluation of popular audio fingerprinting schemes such as searching, retrieval and its applications in a common framework. Also studied the results for Audio retrieval with respect to the size of fingerprints generated compared to size of the compressed audio sample.

References

[1] J. Haitsma and T. Kalker, “A highly robust audio fingerprinting system,” in *Proc. Int. Conf. Music Inf. Retrieval (ISMIR)*, Paris, France, 2002, pp. 107–115.

[2] A.Wang, “An industrial strength audio search algorithm,” in *Proc. Int.Conf. Music Inf. Retrieval (ISMIR)*, Baltimore, USA, 2003, pp. 7–13.

[3] H. Schreiber, P. Grosche, and M. Müller, “A re-ordering strategy for accelerating index-based audio fingerprinting,” in *Proc. 12th Int. Conf. Music Inf. Retrieval (ISMIR)*, Miami, FL, USA, 2011, pp. 127–132.

[4] P. Grosche, M. Müller, and J. Serrà, “Audio content-based music retrieval,”in *Multimodal Music Processing*, ser. Dagstuhl Follow-Ups,M. Müller, M. Goto, and M. Schedl, Eds. Dagstuhl, Germany:SchlossDagstuhl–Leibniz-ZentrumfürInformatik, 2012, vol. 3, pp.157–174.

[5] F. Kurth and M. Müller, “Efficientindex-based audio matching,” *IEEETrans. Audio, Speech, Language Process.*, vol. 16, no. 2, pp. 382–395,Feb. 2008.

[6] D. P. Ellis and G. E. Poliner, “Identifying ‘cover songs’ with chromafeatures and dynamic programming beat tracking,” in *Proc. IEEE Int.Conf. Acoustics, Speech Signal Process. (ICASSP)*, Honolulu, HI,USA, Apr. 2007, vol. 4, pp. IV-1429–IV-1432.

[7] F. Kurth, A. Ribbrock, and M. Clausen, “Identification of highly distortedaudio material for querying large scale data bases,” in *Proc.112th AES Conv.*, 2002.

[8] C.Bellettini and G. Mazzini, “A Framework for Robust Audio Fingerprinting,” J. Comm.,vol.5,no.5, pp. 409-424, 2010.

[9] E. D. Scheirer, “Tempo and beat analysis of acousticalmusical signals,” *J. Acoust. Soc. Amer.*, vol. 103, no. 1, pp. 588–601, 1998.

[11] Juan Pablo Bello. “Audio-based cover song retrieval using approximate chord sequences: testing shifts, gaps, swaps and beats. In Proceedings of the International Conference on Music Information Retrieval,” (ISMIR), pages 239–244, Vienna, Austria, 2007.

[12] Simfy AG, Accessed: Dec. 30, 2012 http://www. simfy.de/start?locale=en.

[13] Shazam Entertainment Ltd, Accessed: Dec.30,2012 http://www.shazam.com/.

[14] TuneUp Media, “Tuneup,” Accessed: Dec. 30, 2012 http://tuneupmedia.com/.