A robust approach to Audio Fingerprinting for identification of songs

A D Mahit Nandan
Roll no. 211AI001
IT Dept.(AI)
NIT Surathkal
Surathkal, Karnataka

G V Ravi Ram
Roll no. 211AI018
IT Dept.(AI)
NIT Surathkal
Surathkal, Karnataka

Harshit Ravindra Gawade
Roll no. 211AI019
IT Dept.(AI)
NIT Surathkal
Surathkal, Karnataka

***Abstract*—Audio fingerprinting is the process of representing an audio signal compactly by extracting the relevant or key components of audio. The recorded fingerprints are matched with audio from other sources and stored in a database for future reference. A robust audio fingerprinting algorithm identifies the audio irrespective of background noise. Some applications of audio fingerprinting are content-based audio retrieval, watermarking etc.**

Keywords—fingerprinting, database, hashing, spectrogram, watermarking, retrieval

I. INTRODUCTION

A content-based signature that summarizes an audio is called an audio fingerprint. Its abilities to monitor sounds have gained interest. These content-based identification methods take an audio file and extract pertinent characteristics, storing them in a database (such as SQLite). When an unidentified audio is presented, its features are extracted and compared to those in the database.

The following are some benefits of employing audio fingerprinting rather than multimedia files:

- *Memory usage is efficient because fingerprints utilize less memory than the original audio.*
- *Effective searching thanks to the compact and small database.*
- *Effective comparison because extraneous background noise is eliminated.*

The ability to handle degradation brought on by sources like pitching (playing the audio at various speeds), background noise, D/A-A/D conversion (digital to analog/analog to digital), and audio coders is essential for an excellent audio fingerprinting device. As seen above, a fingerprint system typically consists of two parts: a mechanism to extract fingerprints and a method to quickly look for matching fingerprints in a fingerprint database for a variety of applications.

II. AUDIO FINGERPRINTING TERMINOLOGY

2.1 Definition

An audio object X with a large number of bits should be mapped by a fingerprint function F to a fingerprint with a small number of bits. We can compare this to so-called hash functions, which are well-known in the field of cryptography. An (often huge) object X is translated into a (typically small) hash value by a cryptographic hash algorithm H. Two huge items, X and Y, can be compared using a cryptographic hash function by simply comparing their respective hash values, H(X) and H. (Y). With only a very small chance of error, strict mathematical equality of the latter pair implies equality of the former. This chance is 2 - n for a properly constructed cryptographic hash function, where n is the total number of bits in the hash. There is an effective way to determine whether or not a specific data item X is included in a given and sizable data set Y=Yi using cryptographic hash functions. In the literature, fingerprinting is also occasionally referred to as robust or perceptual hashing. This is because it avoids storing and comparing with all of the data in Y.

A Very Stable Audio Fingerprinting Method In order to compare H(X) with this collection of hash values, it is sufficient to keep the set of hash values "hi = H(Yi)". At first glance, one could assume that fingerprint functions could benefit from cryptographic hash functions. But keep in mind from the introduction that we are more interested in perceptual resemblance than formal mathematical equivalence. The perception of two items X and Y as being similar to another pair of objects Y and Z does not necessarily imply that X and Z are also similar to each other.

In light of the aforementioned reasoning, we suggest designing a fingerprint function so that perceptually comparable audio objects generate fingerprints that are also similar. Additionally, there must be a very high likelihood that distinct audio objects produce different fingerprints in order to be able to distinguish between them. The threshold T for a properly constructed fingerprint function F should be such that ||F(X)-F(Y)||£T if objects X and Y are similar and ||F(X)-F(Y)||>T if they are dissimilar.

2.2 Audio Fingerprint System Parameters

Now that an audio fingerprint has been properly defined, our attention is on the system's many parameters. The main parameters are : ·

- **Robustness**: Can an audio clip still be recognised even with severely degraded signal? The fingerprint should be based on perceptual qualities that are (at least somewhat) invariant with regard to signal degradations in order to achieve high robustness. Preferably, audio that has been substantially damaged yet leaves behind highly comparable fingerprints. Typically, robustness is expressed using the false negative rate. When the fingerprints of perceptually comparable audio snippets are too dissimilar to result in a successful match, a false negative results.

- **Reliability**: how often is a song incorrectly identified?. The frequency of this is commonly known as the false positive rate.
- **Fingerprint size**: How much space is required for a fingerprint in terms of size? In most cases, fingerprints are kept in RAM memory to facilitate quick searches. As a result, the fingerprint size, which is commonly expressed in bits per second or bits per song, largely determines the memory needs of a fingerprint database server.
- **Granularity**: how many seconds of audio is needed to identify an audio clip? The application might have an impact on granularity. For identifying purposes, the entire song can be used in some applications, but in others, a brief audio clip is preferred.
- **Search speed and scalability**: how long does it take to find a fingerprint in a fingerprint database? Search speed and scalability are important factors for the commercial adoption of audio fingerprint systems.. Speed of search should be approximately of order of milliseconds for a database containing numerous songs using only limited computing resources.

III. METHODS APPLIED

Each audio file is "fingerprinted," which is the extraction of repeatable hash tokens. The analysis is applied to both "database" and "sample" audio files. A sizable collection of fingerprints produced from the music database is compared to the fingerprints from the unidentified sample. After that, the candidate matches are assessed for accuracy of match.

The characteristics that should be used as fingerprints include being sufficiently entropic, resilient, translation-invariant, and temporally localised. According to the temporal locality rule, each fingerprint hash should be calculated using audio samples that are close to the appropriate point in time, preventing the hash from being impacted by distant occurrences. As long as the temporal locality containing the data from which the hash is computed is contained inside the file, fingerprint hashes formed from corresponding matched material are replicable regardless of position within an audio file. This makes sense because any part of the original audio file could contain an unknown sample.

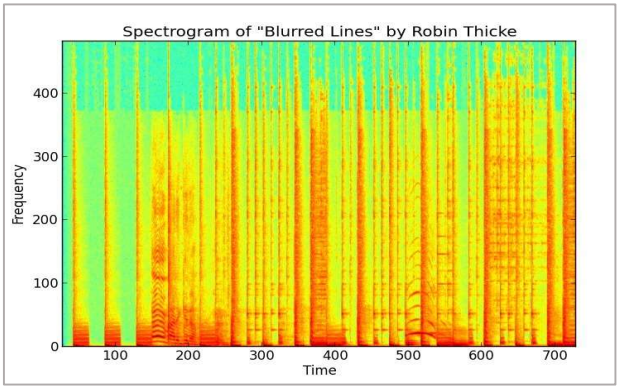


Figure 1

3.1 Extraction Algorithm

The audio signal is initially divided into frames based on chunk size. A set of features is computed for each frame. The features are chosen to be resistant to changes in signal quality. Well-known audio features like Fourier coefficients, Mel Frequency Cepstral Coefficients (MFFC), spectrum flatness, sharpness, Linear Predictive Coding (LPC) coefficients, and others have been presented as features. Additionally, derived values including the means, variances, and derivatives of audio characteristics can be used. In our case, we would be using amplitudes at various instances out of spectrogram to form a template in order to process it.

Due to the low entropy of the raw peak points, determining the exact registration offset directly from peak maps can be fairly time-consuming. A 1024-bin frequency axis, for instance, produces just 10 bits of frequency data maximum every peak. We've created a quick method for indexing peak maps. The peak map, in which pairs of time-frequency points are combinatorially connected, is used to create fingerprint hashes. Each anchor point that is selected has a target zone attached to it. Sequentially pairing each anchor point with points in its target zone results in two frequency components and the time difference between the points for each pair .Even with noise and speech codec compression present, these hashes are extremely repeatable. Additionally, each hash can fit inside a 32-bit unsigned integer. Although the absolute time is not included, each hash additionally carries a time offset from the start of the corresponding file to its anchor point.

The aforementioned method is applied to each track in a database to produce a corresponding list of hashes and their respective offset timings, which is used to form a database index. The small data structs can also have track IDs added to them, resulting in an overall 64-bit struct with 32 bits for the hash, 32 bits for the time offset, and 32 bits for the track ID. The 64-bit structs are arranged in order of hash token value to speed up processing.

A "1" bit represents a white pixel, while a "0" bit represents a black pixel of spectogram and Maximum-Filter is used to reduce the noise in order to increase the robustness of the application.

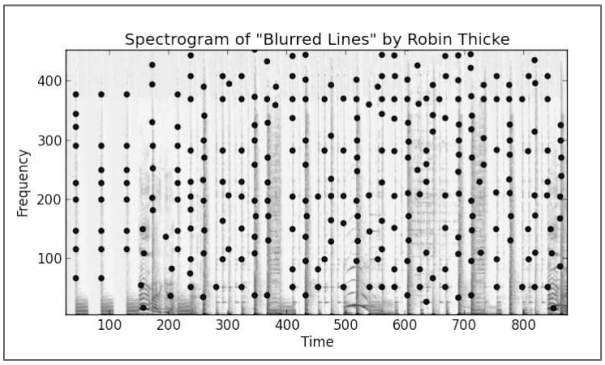
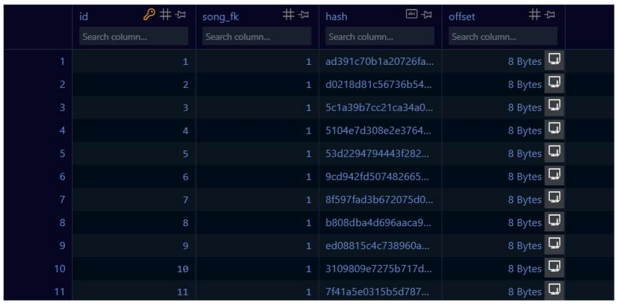


Figure 2

3.2 Searching Algorithm

A collection of hash:time offset records are created on a sample sound file that was captured in order to do a search using the fingerprinting procedure described above. Every hash from the sample is used to do a database search for

equivalent hashes. The appropriate offset timings from the start of the sample and database files are connected into time pairs for each matching hash found in the database. The track ID connected to the matching database hash determines how the time pairings are divided into bins.



	id	song_fk	hash	offset
	Search column...	Search column...	Search column...	Search column...
1	1	1	ad391c70b1a20726fa...	8 Bytes
2	2	1	d0218d81c56736b54...	8 Bytes
3	3	1	5c1a39b7cc21ca34a0...	8 Bytes
4	4	1	5104e7d308e2e3764...	8 Bytes
5	5	1	53d2294794443f282...	8 Bytes
6	6	1	9cd942fd507482665...	8 Bytes
7	7	1	8f597fad3b672075d0...	8 Bytes
8	8	1	b808dba4d696aaca9...	8 Bytes
9	9	1	ed08815c4c738960a...	8 Bytes
10	10	1	3109809e7275b717d...	8 Bytes
11	11	1	7f41a5e0315b5d787...	8 Bytes

Figure 3

The bins are checked for matches once all sample hashes have been used to search the database and create matching time pairs. The set of time pairs within each bin represents an association scatterplot between the sample and database sound files. A sequence of hashes in one file should also occur in the matching file with the same relative time sequence, indicating that the files are identical if matching characteristics appear at similar relative offsets from the beginning of the file. Determining whether a match has been found reduces down to spotting a sizeable group of points that form a diagonal line within the scatterplot. We will be using the offset values along with confidence of match to find which is the best match to given unknown song in the database.

IV. APPLICATIONS

4.1 Broadcast Monitoring

The most popular use of audio fingerprinting is likely broadcast monitoring. It refers to the automatic creation of playlists for radio, television, or web broadcasts for a variety of reasons, including the collection of royalties, the authentication of programmes and advertisements, and the measurement of audience size. Currently, broadcast monitoring is still done manually by "actual" people who are listening to broadcasts and completing out scorecards for organisations interested in playlists, such as performance rights organisations. Multiple monitoring stations and a central site with the fingerprint server make up a large-scale broadcast monitoring system based on fingerprinting. All of the (local) broadcast channels are fingerprinted at the monitoring sites. The monitoring locations send their fingerprints to the central site, which collects them. Then, the fingerprint server, which has a sizable fingerprint database, generates the playlists of all the broadcast channels.

4.2 Connected Audio

The phrase "connected audio" refers generally to consumer applications where music is somehow linked to supplemental and supporting data. One of these examples is the one provided in the abstract, which involves using a mobile device to identify a music. Due to radio station processing, FM/AM transmission, the acoustical path between the mobile phone's loudspeaker and microphone, speech coding, and finally the transmission through the mobile network, the audio signal in this application has been drastically damaged. As a result, this application is exceedingly difficult from a technical standpoint. Other instances of connected audio include (vehicle) radios with identifying buttons and fingerprint software that "listens" to audio streams coming into or going out of a soundcard on a computer. The user of the fingerprint program may be taken to a web page with information on the artist by clicking a "info" button. A Highly Robust Audio Fingerprinting System users could also purchase the CD online by clicking a "buy" button. In other words, audio fingerprinting can provide audio material a universal linking system.

4.3 Filtering Technology for File Sharing

Filtering is the term for deliberate content distribution interference. One may use a more sophisticated strategy in a lawful file-sharing business than just removing copyrighted content. One may see a system that included free music, several premium music genres (available to those with the right subscription), and prohibited music. Although audio filtering might be considered a bad technique from the user's point of view, there are also a number of potential advantages for the consumer. First, it may use the trustworthy meta-data from the fingerprint database to consistently group music song names in search results. Second, fingerprinting can ensure that the downloaded file is exactly what it claims to be.

V. CONCLUSION

A method for audio fingerprinting in this research is implemented in which for every few milliseconds, a subfingerprint is extracted as the basis for the fingerprint extraction.

- **Robustness:** The extracted fingerprints are quite robust.Due to application of Maximum-Filter, the noise got drastically reduced and hence even music recorded and transmitted by a mobile phone can be recognized with significantly less error using them.
- **Reliability:** After several random tests, it has been observed that out of every ten songs, seven were guessed correctly.
- **Fingerprint size:** a 160-bit fingerprint is extracted every few milliseconds, resulting in an average 10kbps fingerprint size. At the cost of more data, we are increasing accuracy of result as less chunksize implies more number of hashes thereby increasing positive expectancy.
- **Granularity:** The granularity as said earlier can be varied according to user. It is a general observation that more granularity gives higher chances of getting correct result.
- **Scalability and search speed:** a fingerprint database with 20,000 songs and the ability to handle thousands of queries per second can be handled on a current PC but initially obtaining a database of fingerprints take time as creation of fingerprint for each song of average duration 300s takes about 15 seconds on an average.

VI. FUTURE WORKS

Other methods of feature extraction and search algorithm optimization will be the main topics of future research. Along with it, research and comparative study on various techniques of noise reduction and preprocessing in order to find most efficient way to tackle the issue of real-time noise to get more accurate results is a future goal.

VII. REFERENCES

[1] Cheng Y., "Music Database Retrieval Based on Spectral Similarity", International Symposium on Music Information Retrieval (ISMIR) 2001, Bloomington, USA, October 2001. [2] Allamanche E., Herre J., Hellmuth O., Bernhard Fröbach B. and Cremer M., "AudioID: Towards Content-Based Identification of Audio Material", 100th AES Convention, Amsterdam, The Netherlands, May 2001.

[3] Neuschmied H., Mayer H. and Battle E., "Identification of Audio Titles on the Internet", Proceedings of International Conference on Web Delivering of Music 2001, Florence, Italy, November 2001.

[4] Fragoulis D., Rousopoulos G., Panagopoulos T., Alexiou C. and Papaodysseus C., "On the Automated Recognition of Seriously Distorted Musical Recordings", IEEE Transactions on Signal Processing, vol.49, no.4, p.898-908, April 2001.

[5] Haitsma J., Kalker T. and Oostveen J., "Robust Audio Hashing for Content Identification, Content Based Multimedia Indexing 2001, Brescia, Italy, September 2001. [6] Oostveen J., Kalker T. and Haitsma J., "Feature Extraction and a Database Strategy for Video Fingerprinting", 5th International Conference on Visual Information Systems, Taipei, Taiwan, March 2002. Published in: "Recent advances in Visual Information Systems", LNCS 2314, Springer, Berlin. pp. 117-128.