A Review of Audio Fingerprinting and Comparison of Algorithms

H.B. Kekre, PhD.
Senior Professor, Computer Engineering,
MPSTME, SVKM's NMIMS University
Mumbai, 400056, India

Nikita Bhandari
Assistant Professor,
Information Technology,
MPSTME, SVKM's NMIMS University
Mumbai, 400056, India

Nisha Nair
Purnima Padmanabhan
Shravya Bhandari
B. Tech, I.T.,
MPSTME, NMIMS University

ABSTRACT

An audio finger print is a small set of features that uniquely identifies a song. An audio fingerprint can be used for broadcast monitoring, audience measurement, meta-data collection. The general framework for building an audio fingerprint includes a front- end and a finger print modeling block. This paper details various uses and properties of an audio fingerprint and also the various stages included in the front end. Two algorithms namely - PRH and MLH have been discussed.

General Terms

Audio Fingerprinting, Content Based Identification, Philips Robust Hashing Algorithm, Multiple Hashing Algorithm.

Keywords

CBID, Philips Robust Hashing Algorithm, Multiple Hashing Algorithm

1. INTRODUCTION

There have been noteworthy advancements in multimedia technologies in this decade, including software technologies for audio and video content. There has also been a considerable increase in the exchange of audio and visual content and information over the Internet. This has been augmented by the rapid growth of the internet. Personal computers have huge music libraries often containing thousands of songs downloaded from a plethora of sources, like the internet, a high-quality CD or a P2P network. Popularity of sharing portal sites such as Youtube, several BitTorrent networks has increased. Often the audio data is unlabeled or the filenames are misspelled, or only parts of information about the audio clip are available to the listener. Audio Fingerprinting is a technique which can be used to identify audio from the signal directly, instead of the tags or the file names or other metadata. This technology can be used for tagging unlabeled audio and also for more serious applications such as broadcast monitoring.

Rights holders often do not allow sharing of their content without permission and consider this as an illegitimate act. Often, legal action is taken against alleged violators who infringe on their rights, or the platforms and networks that encourage and facilitate the sharing of their content [1].

Platforms such as these, however, can only prevent unauthorized content sharing if they can automatically determine whether the content offered on the platform is authorized, i.e. there may not be any form of manual intervention. This requires identification of unlabeled content. Audio fingerprinting technology helps in achieving exactly this. Usually, the core aim of fingerprinting is to check whether the content originates from the same source material. Such fingerprints can be derived from audio, video and images. In such scenarios, right holders can extract a fingerprint from their content and "blacklist" it. Content sharing platforms can then derive a fingerprint from the incoming content that is to be shared on the platform and compare it with the fingerprints that have been blacklisted. If there is a match with one of the fingerprints on the blacklist, the associated content is not allowed to be shared on the content sharing platform.

Audio fingerprinting is best known for its ability to link unclassified or unlabeled audio to corresponding meta-data (e.g. artist and song name), regardless of the format the audio clip is stored as. [2] Essentially, they are Content Based Audio Identification Systems. They extract a perceptual digest of a piece of audio, of reasonable length depending on requirements, which is called the 'fingerprint' and store it in a database. Making use of the stored audio fingerprints and various matching algorithms, distorted versions of a recording can be identified as the same audio content.

There are two phases involved in the identification of an audio clip from its fingerprint:

Enrollment phase: A database or repository is filled with fingerprints and associated metadata of a large number of songs.

Identification phase: In this phase, the fingerprints of unknown songs are extracted and compared with the items in the database. If the fingerprint of the audio clip finds a match in the database, the song will be identified.

1.1 Definition of an audio fingerprint:

An audio fingerprint in the basic sense is a compact content-based signature that efficiently summarizes an entire or a part of an audio recording. [2].These fingerprints extract acoustically relevant characteristics of an audio piece. When an unidentified piece of audio is to be classified, required characteristics of that piece are calculated first and are then matched against the fingerprints stored in the database.

1.2 Alternative content-based identification technology [1]

Cryptographic hash

It is also known as a Message Authentication Code (MAC) or message-digest. Well known examples are the MD5 and SHA family. A MAC is fixed-length (usually 128 or 160 bits). It is not dependent on the length of the message. For ensuring security, a key, which is not disclosed, is input for the computation of the MAC, along with the input message. A hash is bit-sensitive, i.e. changing even one bit in the message changes the entire hash. Two other important characteristics are: pre-image resistance, which is the inability to find a second message which results in the same hash value, and the collision resistance, that is the probability that two random, arbitrary messages result in the same hash value.

Audio Watermarking

Watermarking is a technology alternative to fingerprinting.

Watermarking can be defined as the 'imperceptible insertion of information into multimediadata by modifying the data slightly' [3].It can also be used for applications that make use of audio fingerprinting such as broadcast monitoring. But since the signal itself has been actively altered by introducing changes in it, it cannot be used for legacy content. Finally the embedded message is independent of the multimedia content. Hence, it can have any meaning beyond content identification, such as transaction tracking. In this manner, watermarking makes it possible to distinguish perceptually identical copies. Three combinations of fingerprinting and watermarking are generally used in Digital Rights Management (DRM) applications. First is a technique in which the audio fingerprint is embedded as a watermark for the sole purpose of authentication [1]. In the second method, the fingerprint can be used as an input to the procedure that embeds the watermark. [4, 5, 6, 7]

The watermark becomes largely content dependent and becomes robust to the so called copy attack. Third, in order to locate the start of a watermark message in an audio, the watermark contains "markers". These markers however may be a security risk, as they can be located easily and may be removed from the audio stream. [8]

1.3 Applications

Content ID verification and establishment is an extremely important component in Digital Rights Management (DRM) applications. DRM mainly refers to technologies that support the legal distribution of digital media while at the same time, ensuring the protection of appropriate associated property rights. So DRM can be seen as the conglomeration technology for legal, commercial and technical measures to enable trading of digital items on electronic infrastructures. [9]

The separation of content from rights is an important philosophy of design of a DRM system. [10].This enables the content to be distributed or downloaded freely. Fingerprinting is also used for a wide variety of other applications, some of which have been mentioned below [1, 2]:

- Broadcast monitoring

All advertisers spend money in order to have their commercials aired in accordance with a contract. However, it is a very labor intensive and time consuming task to manually check whether the commercials are actually aired as per the contractual agreement. They automatically monitor a number of radio and television channels looking for specific content, e.g., register when, where, how long has content been aired and so on.

- Audience measurement

Fingerprinting can be used for generating audience count per program or channel or show or music clip, in order to identify and establish which programs a certain panel is watching or listening to. Statistics can be generated on what content is available on the internet. Statistics can also be generated based on relations in the metadata collected using fingerprinting.

- Name that tune

Another example is the 'name that tune' service: if you are wondering what song you're listening to, e.g., on the radio, you can collect and send a few seconds of music using a cell phone. The service computes and matches the fingerprint, and returns a text message containing metadata like artist, song name, album etc. Examples include Shazam and SoundHound.

- Metadata collection

As mentioned earlier, a person collects enormous amounts of music, through different channels like CDs and downloads. Once stored on, say, a hard disk, the metadata often is scattered or unavailable, making organization of the existing content and associating it with related tags very hard.

- Find duplicates

A straightforward application of audio fingerprinting is to find duplicates in large multimedia archives, and to reduce the amount of storage needed.

- Added value services

Once the identity of a song or audio file has been established, service can be offered based on this information. Examples include an offer to buy the song you identified using 'name that tune', targeted advertisement in social networks based on musical interests, offering of related information like biographies, lyrics, news items, based on the customer's preference, etc.

1.4 Properties of Audio Fingerprinting

Following are the desired properties of a good, robust and efficient audio fingerprinting system [1, 2]:

- Accuracy: This measures the degree to which the identification results are correct. The total number of correct, missed, and wrong identifications (false positives).
- Complexity: It refers to the computational overhead and cost involved in extracting the various fingerprints, which include the size of the fingerprint, the complexity of the search algorithm used, the complexity of the fingerprint comparison or matching, the cost of adding new items to the database, etc. This is an extremely important and relevant parameter for systems which require real time operation and deployment and for systems with limited computing resources or bandwidth.
- Fingerprint rate (size): It is the amount of elements or bits extracted per second or song. To ensure the scalability of the system and the database, the

fingerprint size should be as small as possible. The fingerprint size is directly related to the number of fingerprints that can be represented, and to the granularity. Larger the fingerprint rate, finer the granularity.

- Granularity: The minimum length of the audio clip required for a dependable identification. Based on a small fragment, an audio track can be identified accurately. When a system is said to be fine granular, it means that the system is capable of reliable identification of audio clip by making use of small excerpts.
- Reliability: Various methods for assessing whether or not a query is present in the collection of items to identify is of major importance in play list generation for copyright enforcement organizations. In such situations, if a song hasn't been broadcast yet, it should not be identified as a match, even at the cost of missing actual true matches. Reliability is a measure of to what level a fingerprint can be depended upon for its accuracy in returning matches correctly.
- Robustness: Ability to accurately identify an item, regardless of the level of compression and distortion or interference in the transmission channel and withstand the effect of signal processing operations. Other sources of degradation include pitching, equalization, background noise, D/A-A/D conversion, audio coders (such as GSM and MP3), etc.
- Scalability: The audio fingerprinting system must be scalable to hold a large number of fingerprints. This is affected both by the database parameters (search speed, search efficiency, indexing structures), and also by the fingerprint parameters itself (how many fingerprints can be distinguished in a reliable way).
- Security: For certain applications it is highly important that the derivation of the fingerprint from the audio content is dependent on a key. One then should not be able to change the content without changing the fingerprint. Also, it should not be easy to find a different piece of content that generates the same fingerprint (collision), or to learn the key from one or more content items.

2. GENERAL FRAMEWORK [17]

Content-based audio identification (CBID) system is shown in Fig 1. A CBID system extracts a differentiating part of a piece of audio content, i.e. the fingerprint and store it in a database. Unlabeled audio, has its fingerprint calculated and matched against those stored in the database. Using fingerprints and matching algorithms, distorted versions of an audio file can be identified as the same audio content. However, a problem arises during matching of audio content, as many audio files may have similarity in their data or metadata. Thus a hash technique can be used, where a compact representation of a binary file is taken. This method is not robust as a single bit flip can alter the hash function. It also can't be considered as Content Based Audio Identification, as only the bits are taken into account, and not the whole content of the file.

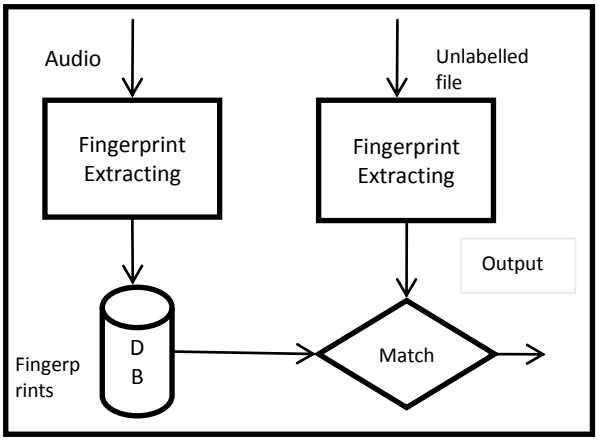

Fig 1: Content Based Audio Identification (CBID) framework

According to [10] good searching methods should be:

- Fast: Huge databases require more time to calculate distances.
- Correct: Should have low False Rejection Rate (FRR).
- Memory efficient: Space overhead required should be small.
- Easily updatable: Objects should be easily inserted, deleted and updated.

Processes involved in Content Based Audio Identification (CBID)

This includes two fundamental processes [17]:

- Fingerprint extraction
- Matching

These fundamental processes are explained below.

Fingerprint Extraction

This includes set of unique characteristics of an audio file, in a concise form. The fingerprint extraction consists of:

- A front-end
- A fingerprint modeling block

2.1Front End [17]:

The front end as shown in Fig 2 is described in detail below:

- Front End: The front end computes a set of measurements from the audio file. The front end converts an audio signal into a string of features that uniquely identify that audio file. These relevant features are then fed to the fingerprint model block.
- Preprocessing: In this step, the audio is digitized and then converted to a general format. It can be a raw format (16bit PCM), or a specific sampling rate (ranging from 8 to 44.1 KHz). If the audio signal is stereo, it is converted to mono by taking an average of the channels. The signal is down sampled.
- Framing and Overlap: Audio signals are highly non-stationary. But the assumption that a signal can be taken as stationary for a time span of a few milliseconds is

important. Hence, the signal must be divided into a number of frames. The rate at which frames are computed per second is called frame rate. A 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.

- Linear Transforms: The use of a linear transform is to map a set of characteristics to a new set of features. An appropriate transform, when used, reduces redundancy. According to [11]**,** there are optimal transforms such as Karhunen- Lo`eve (KL) or Singular Value Decomposition (SVD). These are however complex to compute. Hence, lower complexity transforms are used. The most common transforms used are Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), the Haar Transform or the Walsh-Hadamard Transform. The DFT transform, on comparison to other transforms has been found to be less sensitive to shifting**.** [12]
- Feature Extraction: Most fingerprint extraction algorithms are based on the following approach. First the audio signal is segmented into frames. For every frame a set of features is computed. In this step, there are a variety of algorithms. Additional transformations are applied in order to generate the final acoustic vectors. The objective is again to reduce the dimensionality, and at the same time, to reduce the distortions.
- Post Processing: In this step higher time derivatives are added to the signal model. Some systems only use the derivative of the features, not the absolute features (measured features).[13, 14]. Using the derivative of the signal measurements tends to amplify noise, filters the distortions produced. Sometimes quantization can be applied to the features. The purpose of quantization is to gain robustness against distortions. It also helps reduce hardware and memory requirements.

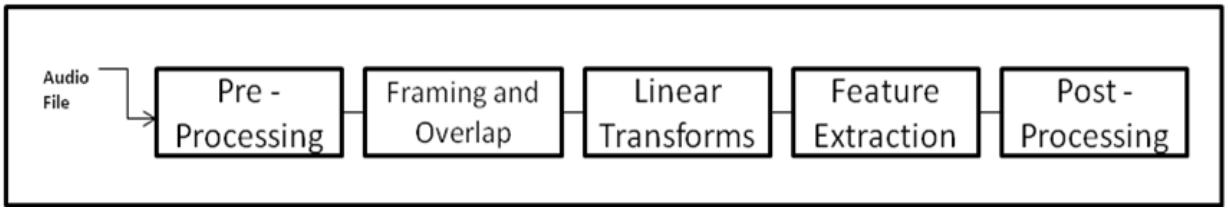

Fig 2: General framework for fingerprint extraction.

2.2 Fingerprint Modeling Block

It is next process of the fingerprint extraction and is used to define the final fingerprint representation.

The fingerprint modeling block usually receives a sequence of feature vectors. These feature vectors are calculated on a frame by frame basis. Depending on the model chosen, distance metric and the design of indexing algorithms is altered for fast retrieval. A very condensed form of fingerprint can be achieved by summarizing the multidimensional vector sequences of a whole song (or a fragment of it) in a single vector.

3. ALGORITHM

3.1 Types of different Audio Fingerprinting Techniques [18]:

Group 1: Systems that use features based on multiple sub-bands, namely Philips' Robust Hash algorithm, which is reported to be very robust against distortions. Phillips Robust Hash Algorithm uses Haitsma and Kalker's algorithm. [15]

Group 2: Systems that use features based on a single band such as the spectral domain, namely Avery Wang's Shazam and Fraunhofer's Audio ID algorithms.

Group 3: Systems that a use a combination of sub-bands or frames, which is optimized through training, namely Microsoft's Robust Audio Recognition Engine (RARE). [16]

3.2 Philips Robust Hashing Algorithm

The Philips Robust Hash (PRH) algorithm has been proved to be a robust content-based audio identification technique. The robustness of the PRH algorithm has been verified mathematically via analyzing the overall bit error probability [19] or bit error rate (BER) [20]. It is important for any fingerprinting algorithm that it not only results in few bit errors, but also allows for efficient searching.

There are two steps in the PRH algorithm. The steps as shown in Fig 3 are explained below:

- Fingerprint Extraction:

  The input, i.e. the audio signal is first divided into overlapping frames. Length of each frame is about 370ms, and frame shift is kept as 1/32 of the frame length. FFT function is then applied to successfully obtain the power spectrum. The next step is the computation of the energies for 33 non-overlapping logarithmically spaced sub bands.[21] From each frame, subfingerprints, or hash strings are then calculated in [21].

- Database Searching:

  The subfingerprints for all the audio files stored in a database are treated as keys and registered in a hash table.Every entry of the hash table has a list of pointers to the audio files. They point to the positions where the sub-fingerprint occurs. From the audio provided by the user as input or query, 256 sub-fingerprints are secured. To find the candidate or audio file that the query belongs to, each sub-fingerprint is compared with the contents of the hash table. -A fingerprint block with the same size as the query block from the candidate position is obtained; Bit error rate (BER) between the fingerprint block of audio file and query audio is calculated and compared with a threshold which is set in advance to 0.35 in [15].The candidate or audio file is accepted as

the result if the two blocks are similar, i.e. BER is less than the threshold.

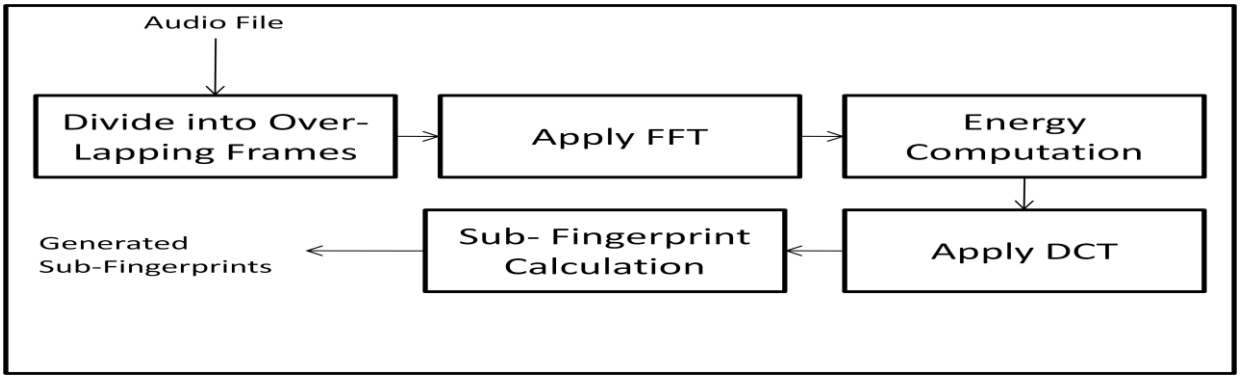

Fig 3: Fingerprint Extraction Stage of PRH algorithm

3.3 Multiple Hashing Algorithm

In [21], a new audio fingerprinting technique called the multiple hashing (MLH) method is proposed.

First three steps are identical to Philips Robust Hashing (PRH). The steps followed in the MLH algorithm are shown in Fig 4. They are:

1. Dividing audio into overlapping frames.
2. Fast Fourier Transform (FFT) function is applied.
3. Energy computation takes place.

The next step however, is performing DCT before determining the hash strings.

4. In each subband, DCT is applied to temporal sequence of energies and for each DCT coefficient stream, a subfingerprint is generated.
5. Only the lower-ordered K values from the DCT coefficients are kept for further computation of subfingerprints.
6. Subfingerprints are derived from the audio files in the database and then recorded and stored in hash tables. K hash tables are constructed because K subfingerprints are computed for each frame. Database searching comprises of three steps:

- Input or query audio is divided into 256 frames, and for each frame, K subfingerprints are obtained using the fingerprint extraction phase.
- The candidate positions are generated in each hash table, and a candidate list is created by compiling all the search results in all included hash tables.
- Calculate BERs by comparing the query fingerprint block with those stored at the candidate positions in the database. The final result is the most hit candidate with BER less than the specified threshold.

Reasons for using DCT in the MLH method:

The decorrelation performance of DCT compared to all other orthogonal transforms is closest to the Karhunen–Loéve

transform [23]. This makes it possible for each subfingerprint to be treated separately and improves overall efficiency by generating more subfingerprints. DCT has a strong energy compaction property [22] This means that most of the signal energy is concentrated in a few low-frequency components.

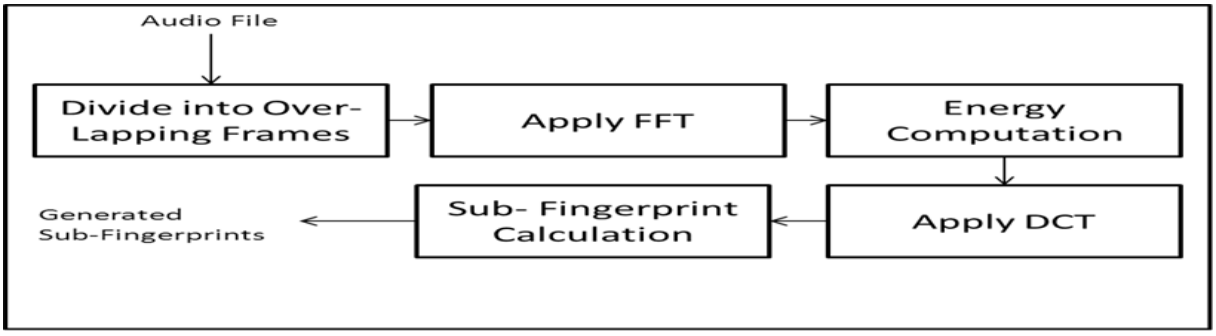

Fig 4: Fingerprint Extraction Stage of MLH method

3.4 Performance comparison between PRH and MLH:

RECOGNITION RATES (%) OF THE PRH AND MLH ALGORITHMS WITH DIFFERENT COMBINATIONS OF HASH TABLES

Overall recognition rates					
Query Set	PRH	MLH			
	0	1	1, 2	1, 2, 3	1, 2, 3, 4
Set 1	97.75	98.58	99.08	99.33	99.75
Set 2	92.25	94.00	96.75	97.50	98.33
Set 3	96.83	97.17	98.17	98.50	99.33
Set 4	34.75	37.75	52.33	60.67	69.58
Recognition rates for rock music					
Query Set	PRH	MLH			
	0	1	1, 2	1, 2, 3	1, 2, 3, 4
Set 1	99.49	100.00	100.00	100.00	100.00
Set 2	95.93	96.18	98.73	99.49	99.49
Set 3	98.98	98.47	99.24	99.49	99.75
Set 4	33.08	33.59	49.87	60.81	70.74
Recognition rates for classical music					
Query Set	PRH	MLH			
	0	1	1, 2	1, 2, 3	1, 2, 3, 4
Set 1	96.95	96.95	97.89	98.36	99.30
Set 2	89.67	92.25	94.13	95.07	97.18
Set 3	93.66	94.84	96.71	97.18	99.30
Set 4	35.92	42.02	54.93	60.80	68.31
Recognition rates for pop music					
Query Set	PRH	MLH			
	0	1	1, 2	1, 2, 3	1, 2, 3, 4
Set 1	96.85	98.95	99.48	99.74	100.00
Set 2	91.34	93.70	97.64	98.16	98.43
Set 3	98.16	98.43	98.69	98.95	98.95
Set 4	35.17	37.27	51.97	60.37	69.82

Table 1: Recognition rates of PRH and MLH algorithms [21]

In Table 1, on comparing Philips Robust Hashing (PRH) with Multiple Hashing Algorithm (MLH), it can be seen that:

- In every query set, taken across different genres of music- such as rock, classical and pop- the MLH algorithm, even with only one hash table, has a higher accuracy than PRH; once, giving an accuracy value of 100%, while PRH accuracy percentage for the same set is 99.49 (Rock music, set 1).
- In some cases, both the algorithms fare equally, such as in the case of Classical music, set 1, with a value of 96.95%. In the case of Rock, set 3, PRH has been found to have a higher accuracy of 98.98% whereas MLH stands at 98.47% for the same set. But, one observes that when another hash table is employed along with the first hash table that was used to compute the MLH accuracy factor, the performance of the MLH algorithm improves significantly (to 97.89% in Classical, set 1 and to 99.24% in Rock, set 3). As can be observed, when more hash tables (3 and 4) were added to hash tables, 1 and 2, the recognition rates of MLH showed a drastic improvement. In some cases, the results after the execution of the 4th hash table are nearly or even greater than twice as those, which were computed after the 1st hash table of MLH or after executing the PRH algorithm. Hence, most of the time, performance improves dramatically when more hash tables are employed, such as in Pop, set 1, where MLH accuracy hit 100% after employing four hash tables to identify the audio clip.

However, it must not be overlooked that although the performance increases considerably and directly with the number of hash tables, hash tables have a high memory and computation tradeoff associated with them. Increase in the number of tables employed increases the memory usage and the time taken to retrieve results.

Therefore, a greater number of hash tables must be employed only if the audio clips have undergone serious distortion by noise. Again, this comes with the cost of higher usage of memory and complexity, which increases computational cost for the user. MLH with one hash table achieves good results, too, and hence, should be used for clips that are corrupted very mildly by noise, and do not justify the burden of employing multiple hash tables.

4. CONCLUSION

Two steps are involved in extracting an audio fingerprint, enrollment- where fingerprints are extracted and added to a database- and identification phase. The general framework for the extraction of a fingerprint consists of two stages: Front-End and the Fingerprint Modeling Block. Two algorithms, PRH and MLH have been compared. MLH, as shown in Table 1, yields better efficiency results as it employs a greater number of hash tables. Increase in the number of tables employed increases the memory usage and the time taken to retrieve results. Further extension of research can be carried out using various alternative transforms such as Walsh, Haar, DST, and so on to observe the differences in various parameters which include (but not limited to) computation speed, efficiency of overall algorithm and error rate.

5. REFERENCES

[1] Modeling Audio Fingerprints : Structure, Distortion, Capacity by P. J. O Doets

[2] A Review of Audio Fingerprinting by Pedro Cano AndEloi Battle.

[3] E. Batlle, J.Masip, and E. Guaus.Automatic song identification in noisy broadcast audio. In IASTED International Conference on Signal and Image Processing, August 2002.

[4] J. Dittmann. Content-fragile watermarking for image authentication in Security, steganography, and watermarking of multimedia contents III, volume 4314 of Proceedings of the SPIE, pages 175 – 184, January 2001

[5] J. Dittmann, A. Steinmetz, and R. Steinmetz. Content-based digital signature for motion pictures authentication and content-fragile watermarking. In International Conference on Multimedia Computing and Systems (ICMCS), volume 2, pages 209 – 213, 1999.

[6] E. G´omez, P. Cano, L. Gomes, E. Batlle, and M.Bonnet.Mixed watermarking fingerprinting approach for integrity verification of audio recordings. In IEEE International Telecommunications Symposium, September 2002.

[7] C.-P.Wu and C.-C. J. Kuo. Speech content integrity verification integrated with. itu g.723.1 speech coding. In IEEE International Conference on Information Technology: Coding and Computing, pages 680 – 684, April 2001

[8] D. Delannay and B. Macq. Watermarking relying on cover signal content to hide synchronization marks. IEEE Transactions on Information Forensics and Security, 1(1):87 – 101, March 2006. W. Jonker and J.-P.Linnartz.Digital rights management in consumer electronics products. IEEE Signal Processing Magazine, 21(2):82 – 91, March 2004

[9] S. R. Subramanya and B. K. Yi.Digital rights management. IEEE Potentials, 25(2):31 – 34, March / April 2006

[10] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval. Addison Wesley, 1999

[11] S. Theodoris and K. Koutroumbas, Pattern Recognition. Academic Press, 1999

[12] G. Richly, L. Varga, F. Kov´acs, and G. Hossz´u, "Short-term sound stream characterisation for reliable, real-time occurrence monitoring of given sound-prints," in Proc. 10th Mediterranean Electrotechnical Conference, MEleCon, 2000

[13] F. Kurth, A. Ribbrock, and M. Clausen, "Identification of highly distorted audio material for querying large scale databases," in Proc.AES 112th Int. Conv., Munich, Germany, May 2002.

[14] E. Allamanche, J. Herre, O. Helmuth, B. Fr¨oba, T. Kasten, and M. Cremer, "Content-based identification of audio material using mpeg-7 low level description," in Proc. of the Int. Symp. of Music Information Retrieval, Indiana, USA, Oct. 2002

[15] J. Haitsma and A. Kalker, "A Highly Robust Audio Fingerprinting System," International Symposium on Music Information Retrieval (ISMIR), pp. 107-115, 2002.

[16] D. P. W. Ellis. (2009) Robust Landmark-Based Audio Fingerprinting. http://labrosa.ee.columbia.edu/matlab/fingerprint

[17] A Review of Algorithms for Audio Fingerprinting by Pedro Cano and EloiBatlle and TonKalker and JaapHaitsma

[18] Comparison of Algorithms for Audio Fingerprinting by Heinrich A. van Nieuwenhuizen, Willie C. Venter and Leenta M.J. Grobler

[19] F. Balado, N. Hurley, E. McCarthy, and G. Silvestre, "Performance analysis of robust audio hashing," IEEE Trans. Inform. Forensics Security, vol. 2, no. 2, pp. 254–266, June 2007.

[20] P. Doets and R. Lagendijk, "Distortion estimation in compressed music using only audio fingerprints," IEEE Trans. Audio, Speech, Lang. Process., vol. 16, no. 2, pp. 302–317, Feb. 2008.

[21] Audio Fingerprinting Based on Multiple Hashing in DCT Domain Yu Liu, Hwan Sik Yun, and Nam Soo Kim, Member, IEEE

[22] K. Rao and P. Yip, Discrete Cosine Transform: Algorithms,Advantages, Applications. New York: Academic, 1990.

[23] N. Ahmed, T. Natarajan, and K. Rao, "Discrete cosine transform," IEEE Trans. Comput., pp. 90–93, Jan. 1974.