COMPARATIVE ANALYSIS BETWEEN AUDIO FINGERPRINTING ALGORITHMS

Milan Chikanbanjar
Department of Computer Engineering
Khwopa Engineering College
Libali-2, Bhaktapur, Nepal
milancb13.cool@gmail.com

Abstract — Fingerprinting or content based identification (CBID) technology work by extracting some unique piece of features/characteristics (often called fingerprint) from audio files and stores them in a database. When an unidentified piece of audio file is presented, fingerprint of that piece is calculated and matched against those which are stored in the database. Here, the research introduces factors that must be considered when performing a comparative evaluation of many fingerprinting algorithms, and presents a new evaluation framework that has been developed to address these factors. By using Chromaprint, Echoprint and Panako acoustic fingerprint algorithms, the research generates fingerprint of variable length audio section for each 300 audio songs and found that Panako is good choice than Echoprint and Chromaprint regarding audio matching queries. The queries were created with sections of audio with no modification as well as modifications like adding noise, recording, increase/decrease audio play speed, changing sampling rate and changing stereo sound to mono. This analysis gives the clear picture on which algorithm should be used to identify music queries recorded in different situations.

Keywords: Fingerprint, Audio Fingerprinting, Echoprint, AcoustID (Chromaprint), Panako, Fourier Transform, Spectrogram, Sampling.

I. INTRODUCTION

Personal music players can show the name of the recording currently playing. But when listening to music in public environments, it can be a challenge to recognize every song that you listen to in a day. [1]

Imagine the following situation. You're in your car, listening to the radio tape and suddenly you hear a song that catches your attention. It's the best new song you have heard for a long time, but you missed the announcement and don't recognize the artist. Still, you would like to know more about this music. What should you do? You could call the radio station or either your friends, but that's too cumbersome. Wouldn't it be nice if you could push a few buttons on your mobile phone and a few seconds later the phone would respond with the name of the artist, album and the title of the music you're listening to. [2]

One of the first audio fingerprinting systems was developed in the 1990s by E. Wold et al. named Muscle Fish. This audio fingerprinting system was used for the identification of a short sound like a doorbell or applause. The algorithm allows for the identification of audio, but cannot be used to recognize a musical track based on an excerpt of the track. From the year 2000, several audio fingerprinting systems were released which enabled the recognition of musical tracks based on their content. Examples are Relatable TRM, which was licensed by Napster for the filtering of copyrighted material [2, 3] and an audio fingerprinting system by Philips Research, which description initially was only released in scientific literature. In 2005, this technology of Philips Research was licensed by Gracenote which implemented it in their audio recognition products. The original Muscle Fish algorithm was in 2000 acquired by Audible Magic, which further developed the algorithm. They use it for their work in the media recognition and copyright management. The website, MusicBrainz.org, which enables users to manage their music libraries automatically, first used Relatable TRM for the identification of audio, but when they reached the boundaries of scalability of this algorithm, they switched to an audio fingerprinting system by MusicIP. [3]

The working principles is generally a two-step process i.e. **submission** and **lookup**.

a) Submission:

First, the fingerprint of audio file is made and then the fingerprint is submitted to server.

b) Lookup:

The analysis of audio fingerprint is done by server and compares it to other fingerprints i.e. millions of fingerprints listed in the database and decides whether it is sufficiently different from known fingerprints as to issue a new ID. Once this step is done, a fingerprint can be calculated for any file and this can be used to look up the corresponding ID.

This ID is associated with a given track or recording and metadata can be gathered from there [4].

II. AUDIO FINGERPRINTING METHODS

1) *Audio fingerprinting framework diagram*

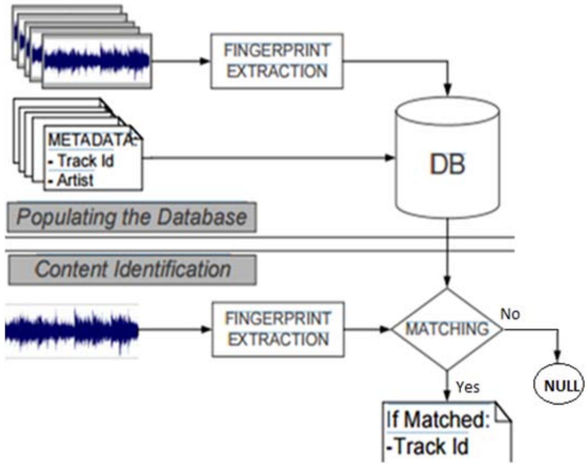

Figure 1: Audio fingerprinting framework [8]

As depict in figure 1, audio fingerprinting have two operating modes:

a) *Builds the database*

b) *Content identification*

By using Echoprint, Panako and Chromaprint algorithms i have generate fingerprint of about more than 300 audio files and compare/analyze to find out which one is better on the basis of number of true matching audio displayed.

2) *Audio fingerprinting extraction process*

Different fingerprinting algorithms extract different types of features from the audio signals. So, the fingerprinting algorithms convert features in to numerical codes, or hashes that represent the value of a feature. Most audio fingerprinting algorithms calculate features from the audio domain, usually using the Fourier transform [6].

3) *General structure of fingerprinting system*

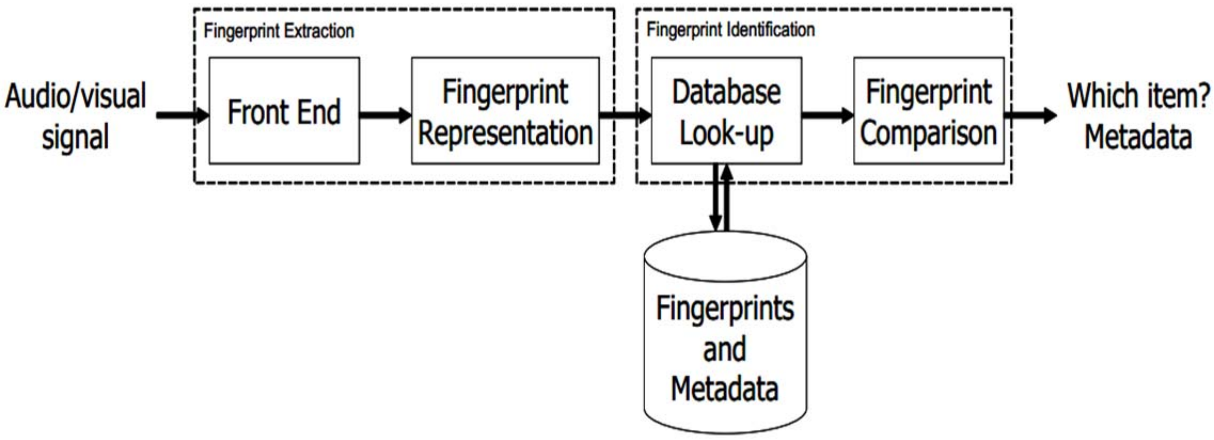

Figure 2: General structure of a fingerprinting system consisting of the fingerprint extraction module, fingerprint identification module and a database containing fingerprints and metadata [6].

Here the fingerprint extraction consists of a front-end and a fingerprint representation/modeling block. Here in this research, the concern is in the fingerprint extraction process as well as in the fingerprint matching/identification process. The fingerprint extraction process is shown in the figure 3.

The front-end computes a set of measurements from the signals. The fingerprint model/representation block defines the final fingerprint representation [3, 8]. When a fingerprint is derived from audio files, the matching algorithm searches a database to find the best match of fingerprints.

4) Common steps used in audio fingerprint algorithm to convert audios to fingerprints[7].

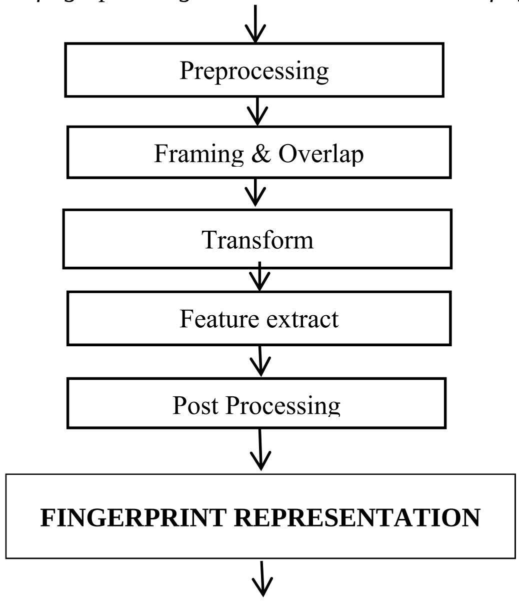

Figure 3: Steps used in audio fingerprinting

Given an input signal sample eventually corrupted, its fingerprint allows to quickly retrieve the original file among a database of known audio files when it exists. Since an input audio sample should be identified, audio fingerprints should be composed of elementary keys (called sub-fingerprints) computed continuously along the signal [17]. As stated by Cano et al. most audio fingerprinting systems extract their fingerprint in the same way. The front-end is the part where the feature extraction takes place and forms the basis for the fingerprint extraction. This front-end and fingerprint modeling block which is discussed by Cano et al., is presented in Figure 3 [3] and described below:

A. FRONT END

1) Preprocessing

Preprocessing involves the conversion of the audio data to a standard format [3]. The audio signal is digitized (if necessary) [7] and convert to mono stream. Furthermore, the audio is often resampled to a specific sample rate. This can be the standard CD-quality rate of 44.1 kHz, but the audio is often down-sampled to a rate between 4 kHz and 8 kHz. This will lead to the loss of the frequencies above 2 kHz to 4 kHz (Nyquist frequency), but these higher frequencies contain less important information for the recognition of the audio and it drastically improves the efficiency of the transformation algorithm. Finally, normalization can be applied, which will provide the amplitudes of the audio data in a standard range for the framing and overlap stage [3]. Here the down-sampling removes the high frequency information; the high frequency component usually contains less energy and therefore more sensitive to distortion and less stable [6].

2) Framing and overlap

The fingerprint needs to be computed for small excerpts of audio as well as complete songs, the input data is split up into small frames of several milliseconds. Due to this framing, only a short excerpt of the audio will be taken into account. This way the signal appears to remain equal over the length of the frame. Therefore, it is possible to determine describing features of the signal within the frame. To avoid discontinuities at the beginning and ending of the frame, a windowing function is applied. This frame is often extended to the surrounding frames (overlap), because this makes the fingerprint robust against small shifts in alignment of the audio data over the frames. Windowing functions that are used in the selected fingerprinting systems are the Hamming and von Hann windows. [3]

3) Transform

The third stage which is the same for most fingerprinting systems is the transformation stage [3]. The idea behind linear transforms is the transformation of the set of measurements to a new set of features. The redundancy is significantly reduced [7]. The most common transformation or the most systems apply the Cooley-Tukey implementation of the Fast Fourier Transform (FFT) to facilitate efficient compression, noise removal and subsequent processing. Some other transforms have been proposed - the Discrete Cosine Transform (DCT), the Haar Transform or the Walsh-Hadamard Transform, [3, 7] Richly et al. did a comparison of DFT and the Walsh-Hadamard Transform that revealed that the DFT is generally less sensitive to shifting.

4) Feature extract

Feature extraction mainly aims at dimensionality reduction in the form of efficient and effective descriptions of the underlying signal. Furthermore, by using features that are based on the most robust signal elements it can increase robustness to distortions. Popular features include Mel Frequency Cepstral Coefficients (MFCC), Spectral Flatness Measure (SFM) and Haar features on spectral energies. [7]

5) Post processing

This step can be used to normalize the features, to emphasize the temporal evolution of the feature sequence (derivatives) or to represent the data in an efficient form. The order of these steps may be different, repeated, or applied on different time or frequency scales. In conclusion, we can say that each of the before mentioned building blocks aims at one or more of the following goals:

• Dimensionality reduction and compact representation
• Increase robustness to distortion
• Emphasize unique characteristics of the signal
• Match perceptual characteristics

B. FINGERPRINT MODELING

The three categories of fingerprint representations are:

1) Fixed size fingerprints

The size of the fingerprint is independent of the song length. Music is non-stationary; parts with different signal and statistical characteristics are mixed in the final representation. There are three drawbacks of such a system. First, when different parts are mixed together in one model, the discriminating characteristics of such fragments are lost in the modeling procedure; when identifying shorter fragments there is only a partial match with the model derived for the entire system. Second, the timing information and the temporal order of the features is a distinguishing feature of a signal. Third, the fingerprint differences cannot be used to locate the differences between the signals. One of the advantages of losing the temporal information is that the model potentially becomes independent of time scaling distortions [6].

2) Constant rate fingerprints

Most fingerprinting systems extract features on regular time intervals (frames). Therefore, the fingerprint size is proportional to the song length. The main advantage is that signal characteristics that are changing over time are not mixed in the final fingerprint. Furthermore, the amount of information extracted in a certain time window can be guaranteed. Finally, when comparing the fingerprint of a distorted version to the fingerprint of the original undistorted recording, the fingerprint difference can be used to localize the changes in the distorted version [6].

3) Variable rate fingerprints

For efficient representation, the rate of fingerprint varies with acoustical events. In this way, the fingerprint only represents those salient characteristics of the underlying acoustic signal. In the Shazam fingerprint, for instance, the spectral peak locations that are most significant in both the frequency and in the temporal dimension represent the fingerprint. This may result in very compact fingerprints. However, one cannot guarantee the amount of information extracted in a certain time window. The part of the fingerprint that corresponds to a particular time instant is called a sub-fingerprint. The fingerprint of a song is thus given by a time-series of sub-fingerprints. A number of sub-fingerprints used for identification are called a fingerprint block [6].

III. ILLUSTRATION OF PROPOSED ALGORITHM

A. Echoprint

"Fingerprinting" of audio files is becoming a necessary feature for any largescale music identification service or system. Echoprint is an open source and open data music identification service available to anyone.

Echoprint is efficient and speedy and works by generating dozens of hashes a second from input audio (microphone or files) and then matching those hashes in a large scale inverted index for queries [12].

The Echoprint algorithm works by finding onsets (points in time where musical notes occur). Features are created by calculating the difference in time between subsequent onsets and creating a hash of these time values. Matching recordings are found by looking for identical hashes in the database [16].

Algorithm of Echoprint

1. Audio signal is converted to mono and their sample rate is reduced to 11025Hz.
2. From the input signal, a 40-pole linear predicator filter is generated to perform the whitening.
3. Once the audio has been down sampled and whitened it is transformed into the frequency domain. A 128 band cosine filter bank is used to perform the transform.
4. The filter bank is moved over the signal with a hope size of 32 samples.
5. The resulting frequency bands are grouped into 8 equally spaced bins by summing the absolute difference of adjacent bands. The eight bins are equally spread out from 0Hz to 5512.5Hz.
6. Echoprint hashes are calculated based on the time difference between musical onsets in each band. The first step of the hashing process is to detect the onsets in the audio signal.
7. In each band, an envelope follower is used to measure the amplitude of the band. When the amplitude reaches a threshold, an onset is registered.
8. To encode the onsets to numerical values, the algorithm considers the time of each onset (o) and the time of its four successors (s1-s4). A hash value is created by taking the time delta between pairs of the five onsets.
9. The two hash values and band index are stored in 40 bits (5 bytes) number (two bytes for each delta and 1 byte for the band index). The number is reduced to 32 bits integer with the Murmur Hash algorithm (Appleby 2009).
10. Each onset and set of successors generates six hashes.
11. Hashes are paired with the time that the onset occurs in the audio query. And are stored in a database.

B. AcoustID (Chromaprint)

It is open source and most recently used audio fingerprinting system. This was created by Lukas Lalinsky and made public around January 2011. Chromaprint is the core component of the AcoustID project. It's a client-side library that implements a custom algorithm for extracting fingerprints from any audio source.

Using the AdaBoost technique described by Jang et al. (2009), the algorithm generates 16 filters that are composed of different sizes of six filters. These 16 filters are pre-calculated as part of the Chromaprint algorithm and do not change. A 12-by-16 sliding window is moved over the chromagram one sample at a time. For each frame, the 16 generated filters are applied to the window. To apply a filter, the filter sums the amount of energy in the white area and subtracts the amount of energy in the black area, resulting in a single value. Each of the filters quantizes the energy value to a 2-bit number. The 2-bit value is encoded using Gray coding, resulting in a binary sequence where each value differs from the previous and next value by only one bit. The 2-bit hash values from each of the 16 filters are converted to a single 32-bit integer representing the sub fingerprint of the 12-by-16 window. The Window is advanced one sample to calculate the next sub fingerprint. The sub fingerprints are stored in an inverted index pointing to the recording in which they occur, and the full fingerprint is stored in a database [13].

Algorithm of Chromaprint

1. Input audio is converted to mono and down sampled to 11025HZ.
2. The audio signal is converted into the frequency domain by performing Short-Time Fourier Transform (STFT) with a frame size of 4096 samples and a 2/3 overlap (2731 samples).
3. The resulting spectrum is converted to 12 bins representing the chroma of the signal.
4. Each bin in the chromagram represents the energy that is present in a musical note. The 12 bins represent the 12 notes.
5. Six filters are used to calculate the hash values for Chromaprint.
6. Using the AdaBoost tech. described by Jang et al. (2009), the algorithm generates 16 filters that are composed of different sizes of the six filters.
7. These 16 filters are pre-calculated as part of the Chromaprint algorithm and do not change.
8. A 12x16 sliding window is moved over the chromagram one sample at a time.
9. For each frame, the 16 generated filters are applied to the window.
10. To apply a filter, the filter sums the amount of energy in the white area and subtracts the amount of energy in the black area resulting in a single value.
11. Each of the filters quantizes the energy value to a 2-bit numbers (from 0 to 3).

12. The two-bit value is encoded using Gray Code resulting in a binary sequence where each value differs from the previous and next value by only one bit.
13. If you do this for every sub-image generated by the sliding window, you get the full audio fingerprint.

C. *Panako*

Panako is an acoustic fingerprinting system. Some parts of Panako were inspired by the Robust Landmark-Based Audio Fingerprinting Matlab implementation by Dan Ellis [17]. The Landmark algorithm uses peaks in the amplitude of the spectrum in each frame to find features to encode as the fingerprint [1]. The system is able to extract fingerprints from an audio stream, and either store those fingerprints in a database, or find a match between the extracted fingerprints and stored fingerprints. Several acoustic fingerprinting algorithms are implemented within Panako. The main algorithm, the Panako algorithm, has the feature that audio queries can be identified reliably and quickly even if they have been speed up, time stretched or pitch shifted with respect to the reference audio.

Analogue physical media such as wax cylinders, wire recordings, magnetic tapes and gramophone records can be digitized at an incorrect or varying playback speed. Even when calibrated mechanical devices are used in a digitization process, the media could already have been recorded at an undesirable speed. To identify duplicates in a digitized archive, a music search algorithm should compensate for changes in replay speed. Next to accidental speed changes, deliberate speed manipulations are sometimes introduced during radio broadcasts: occasionally songs are played a bit faster to fit into a timeslot. During a DJ-set speed changes are almost always present. To correctly identify audio in these cases as well, a music search algorithm robust against pitch shifting, time stretching and speed changes is desired.

The Panako algorithm allows such changes while maintaining other desired features as scalability, robustness and reliability.

Matching Algorithm for Panako

1. Local maxima are extracted from a constant-Q spectrogram from the query. The local maxima are combined by three to form f ngerprints.
2. For each f ngerprint a corresponding hash value is calculated.
3. The set of hashes is matched with the hashes stored in the reference database, and each exact match is returned.
4. The matches are iterated while counting how many times each individual audio identif er occurs in the result set.
5. Matches with an audio identif er count lower than a certain threshold is removed, effectively dismissing random chance hits. In practice, there is almost always only one item with a lot of matches, the rest being random chance hits. A threshold of three or four suff ces.
6. The residual matches are checked for alignment, both in frequency and time, with the reference f ngerprints using the information that is stored along with the hash.
7. A list of audios identif ers is returned ordered by the amount of f ngerprints that align both in pitch and frequency.

IV. RESULTS

A. *Factors for the good audio fingerprinting system*

1) *Robustness:*

This is generally expressed by observing the false negative rate (when the algorithm returns no result, given an available result). In order to achieve high robustness, the fingerprint should be extracted based on perceptual features that do not vary with signal degradation.

2) *Reliability:*

This is expressed through the false positive rate (when the algorithm returns a result but it is incorrect). This is an important consideration as false positives can cause problems with copyright, royalty distribution, users downloading incorrect songs and so on. Echoprint, Chromaprint and Panako are reliable as all the algorithms' incorrect result display is negligible.

3) *Fingerprint size:*

This property represents how much storage space is needed for each fingerprint. This can be an important consideration for consumer tagging systems where fingerprints are sent for identification using cellular data networks. Fingerprint length is calculated so as to find the size. Panako stores the hash values but also displays the fingerprint in graphical method so it requires higher space to store fingerprint.

4) Granularity:

This property indicates how many seconds of audio is needed for a positive match. This will vary depending on the application, with most applications requiring a relatively small granularity such as a consumer music identification service where users will normally capture only a small section from the middle of the audio they wish to identify. Echoprint and Panako could display the correct result even in 5 seconds of audio from middle of audio while Chromaprint could not. It could display the result only with longer audio sections from beginning of audio.

5) Search speed and scalability:

These are related more to the fingerprint database than to the feature extraction algorithm. They represent how fast fingerprint matching is and its computational cost.

The query modifications for the comparative study of algorithms are as follows. Each of the modifications is tested to the three algorithms Echoprint, Chromaprint and Panako.

TABLE 1: QUERY MODIFICATIONS USED IN THE EXPERIMENT
(SOURCE: SIMULATED BY AUTHOR)

Query start time	Query length	Query modification
0	5, 12, 15, 30 seconds	No modification
0	15, 30 seconds	Recorded
0	15, 30 seconds	Mix 1dB white noise
0	15, 30 seconds	Mix 3dB pink noise
0	15, 30 seconds	Increase speed by 1.0%
0	15, 30 seconds	Increase speed by 2.0%
0	15, 30 seconds	Decrease speed by 1.0%
0	15, 30 seconds	Decrease speed by 2.0%
0	15, 30 seconds	Change sample rate to 11k
0	15, 30 seconds	Change sample rate to 8k
0	15, 30 seconds	Convert to mono
30	5, 12, 15, 30 seconds	No modification
30	15, 30 seconds	Recorded
30	15, 30 seconds	Mix 1dB white noise
30	15, 30 seconds	Mix 3dB pink noise

B. Comparison results for the algorithm precision

TABLE 2: TEST RESULTS FOR 0 SECOND START QUERY WITH NO MODIFICATION
(SOURCE: SIMULATED BY AUTHOR)

	0-5 sec songs		0-12 sec songs		0-15 sec songs		0-30 sec songs		Average precision of songs detected
Total songs	300		300		300		300		
Echoprint	135	45%	281	93.67%	285	95%	287	95.67%	82.335%
Chromaprint	1	0.33%	58	19.33%	290	96.67%	299	99.67%	54%
Panako	182	60.67%	280	93.33%	285	95%	300	100%	87.25%

According to the above data gained the average true result given by Echoprint is 82.335%, Chromaprint is 54% and Panako is 87.25%. Chromaprint is not good enough for the 5 seconds and 12 seconds of the audio matching. Whereas Panako and Echoprint displays good result from 5 seconds audio query to 30 seconds audio query.

TABLE 2: TEST RESULTS FOR 0 SECOND START RECORDED QUERY
(SOURCE: SIMULATED BY AUTHOR)

[redacted]					
	0-15 sec songs		0-30 sec songs		Average precision of songs detected
Total songs	300		300		
Echoprint	133	44.33%	174	58%	51.165%
Chromaprint	0	0%	0	0%	0%
Panako	244	81.33%	292	97.33%	89.33%

According to the above data gained the average true result given by Echoprint is 51.165%, Chromaprint is 0% and Panako is 89.33%. Chromaprint failed to find the matching audio when the audio is recorded. Whereas Echoprint find the matching audio only half the test data and Panako is good enough to display 89.33% of the total test recorded audio files.

TABLE 3:TEST RESULT FOR 0 SECOND START 1DB WHITE NOISE MIXED QUERY
(SOURCE: SIMULATED BY AUTHOR)

[redacted]					
	0-15 sec songs		0-30 sec songs		Average precision of songs detected
Total songs	300		300		
Echoprint	203	66.69%	243	80.19%	73.44%
Chromaprint	223	73.59%	235	77.55%	75.57%
Panako	282	93.06%	296	97.68%	95.37%

According to the above data gained the average true result given by Echoprint is 73.44%, Chromaprint is 75.57% and Panako is 95.37%.All three algorithm are good to display the matching audio when 1dB noise was added to the query of 15 and 30 seconds. Among the algorithms Panako gave the best result.

TABLE 4: TEST RESULT FOR 0 SECOND START 3DB PINK NOISE MIXED QUERY
(SOURCE: SIMULATED BY AUTHOR)

[redacted]					
	0-15 sec songs		0-30 sec songs		Average precision of songs detected
Total songs	300		300		
Echoprint	176	58.67%	221	73.67%	66.17%
Chromaprint	120	40%	242	80.67%	60.335%
Panako	231	77%	277	92.33%	84.665%

When the query with 3dB pink noise mixed was tested, Echoprint displayed average true result of 66.17%, Chromaprint displayed average true result of 60.335% and Panako displayed average true result of 84.665%. Echoprint and Chromaprint results were poor for 15 seconds of 3dB mixed audio query.

C. Results

This analysis tested three fingerprinting algorithms on a test set of 300 recordings. Each test file was modified in a different way to represent the kind of query. Each of the fingerprinting systems was tested with 34 different types of query.

From the results,

- The algorithms precision decreases when the audio query is modified.
- Very few audio are detected when audio query is change with respect to the play speed.
- The Chromaprint performed well with almost all modified queries, especially when the query was long.
- One major failing of the Chromaprint is that it requires queries to be taken from the beginning of the recording.
- Chromaprint could not detect audio when query is given from middle section of audio.
- Hence, Echoprint and Panako stand good choice than the Chromaprint.
- For short signals, both Echoprint and the Panako perform well.
- For 30 seconds start query, Panako gave more accurate results than Echoprint.
- So, again comparing between the Panako and Echoprint, Panako seems good choice than Echoprint.

V. CONCLUSION

This analysis presented a review of fingerprinting algorithms and developed a new evaluation framework which was used to compare the accuracy of three different fingerprinting algorithms when presented with different audio queries. This analysis gives the clear picture on which algorithm should be used to identify music queries recorded in different situations. The results of the evaluation are useful to people wanting to choose an audio fingerprinting algorithm for their own use.

From the study of audio fingerprinting algorithms, it was known that audio fingerprints are required for the copyright and finding duplicate audio files.

From the observations, it can be concluded that the algorithms precision decreases when the audio query is modified. Very few audio are detected when audio query is change with respect to the play speed. The Chromaprint algorithm performed well with almost all modified queries, especially when the query was long. One major failing of the Chromaprint algorithm is that it requires queries to be taken from the beginning of the recording. Chromaprint could not detect audio when query is given from middle section of audio.

Hence, Echoprint and Panako stand good choice than the Chromaprint. For short signals, both Echoprint and the Panako algorithm perform well. For 30 seconds start query Panako gave more accurate results then Echoprint. So, again comparing between the Panako and Echoprint, Panako seems good choice than Echoprint.

Also from observation and analysis, following facts can be concluded:

- The good choice among three is Panako.
- As from the literature review, it is known that the Panako algorithm has the feature that audio queries can be identified reliably and quickly even if they have been speed up, time stretched or pitch shifted with respect to the reference audio.

REFERENCES

[1] Alastair Porter, "Evaluating musical fingerprinting systems," PhD thesis, Montreal, Canda, 2012.

[2] Philips Haitsma Jaap and Philips Kalker Ton, "A Highly Robust Audio Fingerprinting System," in proc. International Symposium on Music Information Retrieval (ISMIR), Paris, France, 2002.

[3] Marco Kruijswijk, "Matching Audio Fingerprinting Systems," June 25 2007.

[4] musicbrainz.org. [Online]. https://musicbrainz.org/doc/Fingerprinting, (2015, January)

[5] www.iponz.govt.nz [Online], (2013, Agust 27)

[6] Peter Jan Otto DOETS, Modeling Audio Fingerprints:Structure, Distortion, Capacity. Amsterdam: TNO, October 2010.

[7] MarcLeman SixJoren, "PANAKO – A SCALABLE ACOUSTIC FINGERPRINTING SYSTEM HANDLING TIME-SCALE AND PITCH MODIFICATION," in 15th International Society for Music Information Retrieval Conference (ISMIR 2014), Department of Musicology, Ghent University , 2014.

[8] E Batle, T Kalker, J Haitsma P Cano, "A Review of Algorithms for Audio Fingerprinting," Multimedia Signal Processing, 2002 IEEE Workshop, 2002

[9] Solon P. Pissis,Jyh-Shing R. Jang,Costas S. Iliopoulos Kaichun K. Chang, *Sub-Nyquist Audio Fingerprinting for Music*. London, UK, 2010.

[10] G Richard, Y Grenier S Fenet, "A SCALABLE AUDIO FINGERPRINT METHOD WITH ROBUSTNESS TO PITCH-SHIFTING," in *12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, Paris, France, 2011.

[11] Ms. Komal V. Goenka Ms. Pallavi K. Patil, "Overview of Audio Watermarking Techniques," *International Journal of Emerging Technology and Advanced Engineering*, vol. ISSN 2250-2459, Volume 2, no. Issue 2, February 2012.

[12] Daniel P. W. Whitman, Brian Porter, Alastair Ellis, "Echoprint: An Open Music Identification Service," ISMIR 2011, October 2011.

[13] Lukas Lalinsky. (2015, March) oxygene.sk. [Online]. https://oxygene.sk/2011/01/how-does-chromaprint-work/

[14] Jerome Lebosse, "AUDIO FINGERPRINT IDENTIFICATION BY APPROXIMATE STRING," in Proceedings of ISMIR 2007, Sep 2007, Vienna, Austria, 2007.

[15] Tim Clark. A Review of Audio Fingerprinting and its Applications. [Online]. https://www.researchgate.net/publication/259557182_A_Review_of_Audio_Fingerprinting_and_its_Applications

[16] The echonest. (2015, March) echprint.me. [Online]. http://www.echoprint.me/how

[17] Dan Ellis. /www.ee.columbia.edu. [Online]. http://www.ee.columbia.edu/~dpwe/resources/matlab/fingerprint/