ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

Triple Point Geometric Hashing based Audio Fingerprinting

Author:
Efriem Desalew Gebie

Advisor:
Dr. Surafel Lemma Abebe

A thesis submitted in partial fulfillment of the requirements for the Masters of Science in Computer Engineering

June 9, 2020
Addis Ababa, Ethiopia

Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Triple Point Geometric Hashing based Audio Fingerprinting

by Efriem Desalew Gebie

Approval by Board of Examiners

Dr. Yalemzewd Nagash Dean, SECE, AAiT	Signature
Dr. Surafel Lemma Advisor	Signature
Internal Examiner	Signature
External Examiner	Signature

Addis Ababa, Ethiopia

Declaration of Authorship

I, Efriem Desalew Gebie, declare that this thesis titled, "Triple Point Geometric Hashing based Audio Fingerprinting" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a masters degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.

Signed:

Date:

Abstract

Audio fingerprinting is a technique used for exact identification of an audio by extracting perceptually relevant audio features and transforming them into condensed reproducible formats. Different approaches are proposed to develop audio fingerprinting system. Based on their baseline assumption, these approaches can be grouped into three categories: Philips, Image Processing and Shazam approach. These audio fingerprinting systems, however, are not usually effective when the audio is distorted. Distortion in an audio might come from different modifications such as additive noise, speed change, pitch shifting, time stretching and others. Of these modifications, this thesis focuses on handling the problem of linear speed change in Shazam based audio fingerprinting system. Linear speed change is a common audio modification which occurs when the audio is played faster or slower with a constant rate. In this thesis, a Shazam based audio fingerprinting system which is robust to linear speed change is proposed. The proposed approach employs triple point geometric hashing to handle the effect of linear speed change on audio fingerprints.

The proposed approach is evaluated using 29,600 query audios, and compared with the baseline work, Shazam and recent Shazam based work, Panako. Evaluation results show that the proposed approach is robust to linear speed change in a range from -30% to 22%. This is a significant improvement compared to Panako, which is robust to linear speed change between -12% to 6%, and Shazam which failed to handle 2% linear speed change. In addition to speed change, the proposed approach is evaluated in terms of robustness to additive noise, time stretching and pitch shifting. The results show that the proposed approach is robust to: i) additive noise in a range from -5dB to 20dB, comparable robustness is also exhibited by Shazam and Panako; ii) time stretching in a range from -10% to 8%. This is also an improvement compared to Shazam and Pankao, which are robust to time stretching between -4% to 4%; and, iii) pitch shifting in a range from -4% to 4%, which is comparable robustness with Panako, where Shazam failed to handle 2% pitch shifting.

Keywords: *Audio Fingerprinting, Audio Identification, Geometric Hashing, Linear Speed Change*

Acknowledgements

I would like to express my gratitude to my advisor, Dr. Surafel Lemma , who guided me throughout this research.

Thank You!

Contents

List of Figures

List of Tables

List of Abbreviations

CQT	Constant Q Transform
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
FN	False Negative
FP	False Positive
MFCC	Mel Frequency Cepstral Coefficients
MIR	Music Information Retrieval
ORB	Orient Fast and Rotate Brief
PCM	Pulse Code Modulation
SIFT	Scale Invariant Feature Transform
STFT	Short Time Fourier Transform
TDHS	Time Domain Harmonic Scaling
TN	True Negative
TNR	True Negative Rate
TP	True Positive

Chapter 1

Introduction

Nowadays audio is almost everywhere, its there when we watch TV, listen to radio, celebrate party, workout in a gym, worship in a church, ride bus etc. As a human being, we have an ability to memorize audios we already heard and it is an impressive skill to memorize a song even in a presence of distortion. However, this memorization skill is a very limited skill considering the available millions of songs.

Imagine you are in a bus and heard a new song which you never heard before, and you asked those who are around you for the title, but you didn't get the answer. What can you do ? Of course, you can take out your phone and send short snippet of the song to apps like Shazam or SoundHound, and they will respond to you with required meta-data almost instantly. But, how could they do it? Well, they use a technique called audio fingerprinting. This technique, audio fingerprinting, is neither the only technique nor a new technology. Audio identification techniques, such as audio fingerprinting, audio watermarking and others have been studied since the time computers were available in research environments under an area called Music Information Retrieval (MIR) [1]. However, the study presented in this thesis is only limited to audio fingerprinting.

The importance of audio fingerprinting is not limited only to retrieving audio meta-data. It is also important in other application scenarios, such as broadcast monitoring, royalty tracking, copyright enforcement, audio file organization and more. Despite its importance, implementation of audio fingerprinting has not been easy due to the fact that audio is prone to modification. Of these audio modifications which impose difficulties while implementing audio fingerprinting systems, this thesis focuses on handling the problem of linear speed change. Linear speed change is a common modification which occurs when the audio is played faster or slower with constant rate [2].

1.1 Problem Statement

Linear speed change is a modification which affects an audio dynamics simultaneously along the pitch and tempo axis. This type of modification occurs when the audio is played at a faster or slower speed with a constant rate. Playing an audio at a faster speed increases its tempo (speed of the underlying beat) and raises up its pitch (position of a sound along frequency axis). Whereas, playing audio at slower speed introduces the exact opposite effects. Different researches are proposed to develop audio fingerprinting system which is robust to linear speed change. Based on their baseline assumption, these works can be grouped into three categories: Philips, Image Processing and Shazam approach.

Based on Phillips approach, Haitsma and Kalker [2] were able to achieve $\pm$10% robustness to linear speed change. However, according to the investigation conducted by Ouali et al. [3], -20% to 23% speed modification is common. Therefore, 10% robustness is not enough. Recent work based on image processing approach, Zhang et al. [4] achieved $\pm$30 % robustness to linear speed change using image processing tool called SIFT (Scale Invariant Feature Transform). This approach, however, is expensive both in storage and computation requirement due to the nature of underline image processing algorithms. In the last approach (Shazam), Six and Leman [5] reported $\pm$8 robustness to linear speed change using the association of triple spectral peaks to extract audio fingerprints; and, Sonnleitner and Widmer [6] enhanced this robustness to $\pm$30% using association of four spectral peaks to extract speed change invariant audio fingerprints. Increasing the number of spectral peaks, however, brings two problems: i) it increases computational and storage requirement of the system; and, ii) it will degrade the robustness of the system to other modifications as the increase in the number of spectral peaks leads to extraction of audio fingerprints with high entropy which are difficult to reproduce. Henceforth, new ways of approaching the problem are still required.

This thesis aimed to develop Shazam based audio fingerprinting system which is robust to linear speed change using triple point geometric hashing as a fingerprint extraction scheme. Strong emphasis is given to assess the effect of linear speed change in audio fingerprinting and evaluate the possibility of using triple point geometric hashing to extract speed change invariant audio fingerprints.

1.2 Objectives

1.2.1 General Objective

The general objective of this thesis is to develop an audio fingerprinting system which is robust to linear speed change.

1.2.2 Specific Objectives

- To assess the effect of linear speed change in Shazam based audio fingerprinting systems.
- To develop fingerprint extraction method which is robust to linear speed change using triple point geometric hashing.
- To collect reference audios and prepare test query audios which can be used for evaluation.
- To evaluate proposed audio fingerprinting system and compare it with previously developed systems.

1.3 Research Methodology

While conducting this research, steps outlined below are followed:

- **Literature Review:** Throughout the research literatures have been reviewed to formulate the problem, identify gaps in prior works and propose an approach.
- **Propose an Approach:** Once a clear problem is formulated, an approach which can be a solution for the stated problem is proposed.
- **Data Collection and Preparation:** Reference audios are collected and from the collected reference audios, query audios are generated by applying required audio effects.
- **Verify Proposed Approach:** Finally, the proposed approach is evaluated using prepared test audio corpus.

1.4 Scope

As it is mentioned in the Problem Statement 1.1, there are three possible ways to develop audio fingerprinting system. However, the scope of this research is only limited to handling the problem of linear speed change in Shazam approach.

1.5 Contributions

In this thesis, an open source audio fingerprinting system is developed, and it is made available on GitHub, https://github.com/Efode-r2d2/Efode, under GNU General Public License v3.0 and specific contributions of this work are summarized as follow:

- Developed triple point geometric hashing based audio fingerprint extraction technique which is robust to linear speed change.
- Developed open source audio processing toolbox which can be used for large scale audio manipulation. This project can be found here https://github.com/Efode-r2d2/eu-Audio under GNU General Public License v3.0.
- Reproduced the baseline work called Shazam which can be used for future related researches or practitioners interested in audio processing. Reproduced work is also available on GitHub, https://github.com/Efode-r2d2/Shazam , under GNU General Public License v3.0

1.6 Thesis Organization

The rest of this thesis is organized as follow. Chapter 2 discuss theoretical backgrounds including definition, requirements, challenges and applications of audio fingerprinting system. Chapter 3, presents summary of previous works proposed to handle problem of linear speed change. In Chapter 4 the proposed approach is presented. Experiments conducted to evaluate proposed approach as well as discussions on results are presented in Chapter 5. Finally, Chapter 6 summarized the research and identified gaps which can be researched in the future.

Chapter 2

Audio Fingerprinting

Audio fingerprinting is a technology used for exact identification of an audio. Its typical use is to precisely identify a piece of query audio from a large reference audio collection [7]. A query is an audio excerpt that is potentially modified. Typical modifications in audio includes additive noises, pitch shifting, time stretching, speed change and others.

In audio fingerprinting, identification is based on establishing perceptual equality between two audio objects: not by comparing the audio objects themselves, but by comparing their associated fingerprints. Haitsma and Kalker [8] stated advantages of using fingerprints for comparison instead of the audio objects themselves as follow:

- Reduced memory/storage requirements as fingerprints are relatively small.
- Efficient comparison as perceptual irrelevancies have already been removed from fingerprints.
- Efficient searching as the data-set to be searched is smaller.

Looking at a higher level, audio fingerprinting system need to answer two prominent questions: how to extract audio fingerprint which can uniquely represent a given audio and how to perform matching between extracted audio fingerprints?

To answer the above two questions, researchers in this area approached audio fingerprinting as a two part system: fingerprint extraction and fingerprint matching. Fingerprint extraction deals with issues related to extracting perceptually relevant audio features and transforming extracted features into condensed reproducible formats (audio fingerprints). Whereas, fingerprint matching perform a match between audio objects based on their associated fingerprints.

2.1 Fingerprint Extraction

Fingerprint extraction deals with extracting perceptually relevant audio features and transforming those extracted audio features into condensed reproducible formats (audio fingerprints). Two key issues which has to be addressed while extracting audio fingerprints are:

- **Feature Extraction:** deals with identifying perceptually relevant audio features which can be used to uniquely represent an audio.
- **Fingerprint Formation:** deals with transforming extracted audio features into condensed formats (audio fingerprints) which can be reproduced even in the presence of audio modification.

2.1.1 Feature Extraction

Feature extraction is a process of transforming a given high dimensional audio data into low dimensional data by avoiding perceptually irrelevant audio features. The very first thing while extracting audio features is to identify perceptually relevant audio features which are useful in capturing unique characteristics of an audio. Haitsma and Kalker [8] grouped set of relevant audio features in to two classes: semantic and non-semantic features.

Typical features in the semantic class are genre, beats-per-minute, and mood. These types of features usually have a direct interpretation, and they are commonly used to classify audios (classify music, generate play-lists and more). They are not common in exact music identification. Non-semantic class consists of features that have mathematical nature and are difficult for humans to read directly from music (eg. spectral peaks extracted form power spectrum of an audio). Non-semantic audio features can be extracted using mathematical tools such as: Mel Frequency Cepstral Coefficients (MFCC) [9], Discrete Cosine Transform (DCT) [10], Short-Time Fourier Transform (STFT) [11], Constant Q-Transform (CQT) [12], and Wavlet Transform [13].

Even-though, its difficult for human auditory system to directly interpret non-semantic audio features, they are preferable over semantic features for capturing unique characteristics of an audio for the following reasons [8]:

- Semantic features don't always have a clear and unambiguous meaning.
- Semantic features are in general more difficult to compute than non-semantic features.

- Semantic features are not universally applicable. For example, beats-per-minute does not typically apply to classical music.

Among the above listed tools used for non-semantic audio features extraction, STFT (Short Time Fourier Transform) and CQT (Constant Q Transform) are commonly used in audio fingerprinting systems reviewed in this thesis. Short-time fourier transform (STFT) is a sequence of fourier transforms of a windowed signal. STFT provides the time-localized frequency information for situations in which frequency components of a signal vary over time, whereas the standard fourier transform provides the frequency information averaged over the entire signal time interval [14]. Like STFT, a constant Q transform (CQT) is another commonly used music analysis tool. CQT, however, has geometrically spaced center frequencies and increasing time resolution towards higher frequencies [15]. MFCC is also another spectral analysis tool widely used as features extraction tool in speech recognition systems and trained music information retrieval techniques such as song similarity estimation, music recommendation and artist recognition, however, its application in non-trained music information retrieval techniques such as audio fingerprinting is not as such convenient [16].

2.1.2 Fingerprint Formation

Once relevant audio features are extracted, next step in fingerprint extraction is to transform extracted audio features into condensed formats (audio fingerprints) which are reproducible even in the presence of audio modification. While forming audio fingerprints, defining clear fingerprint format is considered as a key factor. According to Haitsma and Kalker, there are two options in defining audio fingerprint formats [8]. The first option is to represent fingerprints as a vector of real numbers, where each component expresses the weight of a certain basic perceptual feature. A second option is to use hash functions and transform given audio features as bit-strings. The choice of fingerprint format directly affects design issues in the fingerprint matching.

In addition to defining clear fingerprint format, audio fingerprints are also expected to meet other key requirements [17]:

- **Temporally Localized:** each fingerprint is calculated using audio samples near a corresponding point in time, so that distant events do not affect the fingerprint.
- **Translation Invariant:** fingerprints derived from corresponding matching content are reproducible independent of position with in an audio file, as long-as portion of audio containing the data from which the fingerprint is computed contained in the file.

- **Robust to Noise:** fingerprints extracted from the original clean audio should be reproducible from the degraded copy of the audio.
- **Sufficiently Entropic:** insufficient entropy leads to excessive and spurious matches at non-corresponding locations, requiring more processing power to cull the results. Too much entropy usually leads to fragility and non-reproducibility of fingerprint tokens in the presence of noise and distortion.

2.2 Fingerprint Matching

Fingerprint matching is concerned with calculating similarity between two audio objects based on their associated fingerprints and this can be realized by addressing the following three key issues:

- **Reference Fingerprint Database:** deals with issues related to storing extracted audio fingerprints.
- **Search Strategy:** deals with issues related to searching for a match for a given query audio fingerprint.
- **Verification:** deals with issues related to verifying a match returned by the employed search strategy. Verification is a key issue which determine the reliability of fingerprint matching system.

2.2.1 Reference Fingerprint Database

Reference fingerprint database is responsible for storing fingerprints extracted from reference audio collections in a way which is suitable for matching. The nature of this database depends on the type of fingerprint format employed. According to revised works, for vector like fingerprint formats it is preferable to use tree based data structures, such as blacked B-Tree [5] and R-Tree [6]. Bit-string based fingerprint formats are more compatible with hash tables. Locality sensitive hashing is the most commonly used technique to construct hash tables [17].

2.2.2 Search Strategy

Calculating a match between audio fingerprints can be implemented in two possible ways: brute-force search and indexing. Brute-force search strategy is based on calculating similarity by iteratively comparing the given query audio fingerprint with the whole reference fingerprints. However, in indexing based search strategy fingerprint

of the query audio is used as an index to search for a matching audio. Compared to indexing, brute-force based search strategy is computationally expensive for an obvious reasons specially when it is implemented with large audio fingerprinting systems.

2.2.3 Verification

Here, the main goal is to verify the match returned by employed search strategy, so that the reliability of matching system can be enhanced. In this case, reliability is defined as minimizing false positive rate of the system.

2.3 Requirements for Audio Fingerprinting System

An ideal fingerprinting system should fulfill several requirements. The requirements depends on the application scenario. However, there are some requirements which are useful in order to evaluate and compare different audio fingerprinting systems. Those mandatory requirements are listed as follow [8], [18]:

- **Robustness:** ability to accurately identify an item, regardless of the level of audio modification. Ideally, audio fingerprinting systems are expected to detect severely degraded audio contents. However, in a practical scenarios there are still challenging audio modifications which are difficult to handle and those challenges are discussed in Section 2.4.

- **Reliability:** in a reliable audio fingerprinting system, if fingerprints for an audio has not been stored in a reference fingerprint database, an audio will not be identified as a match, while keeping the cost of missing actual matches as minimum as possible. The reliability of the system mainly depends on a fingerprint matching system, specifically on an employed match verification method.

- **Granularity:** ability to identify whole titles from excerpts of a few seconds long audio. It requires to deal with shifting, that is lack of synchronization between the extracted fingerprint and those stored in the database and it adds complexity to the search (it needs to compare audio in all possible alignments).

- **Scalability:** performance with very large databases of titles or a large number of concurrent identifications. This affects the accuracy and the complexity of the system.

- **Versatility:** both query and reference audios may come in different audio formats, but an audio fingerprinting system is expected to exhibit similar performance regardless of the audio format. Versatility can also be seen as ability to use the same database for different applications.

- **Complexity:** it refers to different computational costs. Examples of computational costs are cost of fingerprint extraction, complexity of the search, complexity of fingerprint comparison, and cost of adding new items to the database.

2.4 Challenges in Audio Fingerprinting

Developing audio fingerprinting system which can fulfill all the above requirements is a challenging task. Any kind of modification which can alter the dynamics of an audio can be considered as a challenge to a typical audio fingerprinting system. Common challenges in audio fingerprinting systems are:

- **Additive Noise:** noise in audio can be defined as any additive spectral information which has not been originally in the audio. The addition of new spectral informations to an audio will change the perceptual information of an audio which in-turn affects performance of audio fingerprinting systems.

- **Speed Change:** speed change is a common audio modification. This type of modification might happen in a DJ mix when the DJ tried to increase or decrease the tempo of the audio, in broadcast service when a specific program is played faster or slower to make it fit into an allocated time slot, and other similar scenarios. As it is stated in Section 1.1, speed change affects the dynamics of the audio along both tempo and pitch axis, and this impose inevitable difficulty in audio fingerprinting systems. Audio speed modification can be either linear or non-linear. In linear speed change, uniform speed change is applied throughout the whole audio either by uniform re-sampling on the whole audio or by removing and duplicating samples in a uniform fashion. Whereas, in non-linear speed change, the change is not uniform throughout the audio and this can be achieved either by applying non-uniform re-sampling or removing and duplicating audio samples in non-uniform fashion.

- **Time Stretching:** is a process of changing the tempo of the audio (speed of the underlying beat) without affecting its pitch (position of a sound along frequency axis) [19]. There are variety of methods used for applying time stretching. However, according to Bernsee [19], Phase Vocoder [20] and Time Domain Harmonic

Scaling (TDHS) [21] are the most common tools used for applying time stretching.

- **Pitch Shifting:** is a process of changing the pitch of an audio signal without affecting its tempo [19]. In practical applications this is usually achieved by first applying time-stretching and then re-sampling to change the pitch. Compared to speed change, both time stretching and pitch shifting are more technical and complex audio modifications.

- **Other Effects:** other common audio modification types which affect audio fingerprinting systems are:

  - Equalization - boosting or reducing (attenuating) the levels of various frequencies in a signal.
  - Reverb - short for reverberation, the effect of many sound reflections occurring in a very short space of time.
  - Chorus- the effect of making a signal sound like it was produced by multiple similar sources.
  - Filter Effects: they are used to emphasize or suppress frequencies in an audio signal, resulting in a change to the tonal color of the audio.

2.5 Applications of Audio Fingerprinting

The role of an audio fingerprinting system is to capture the signature of a piece of audio, such as a song, that allows it to be differentiated from other audios. This unique capability can be applied to numerous practical scenarios such as:

- **Content Based Audio Identification:** Companies such as: Shazam Entertainment Ltd. [1]and SoundHound Inc.[2] serve million of users around the world using audio fingerprinting to retrieve an associated meta-data from a given short query audio (potentially distorted). The application scenario in here is to store the hash of millions of songs along with their meta-data(title, artist name, lyrics, link to the audio file or link to videos related to the audio link) and respond meta-data of matching audio based on query audios sent from the users.

- **Broadcast Monitoring:** Another potential application of audio fingerprinting is in monitoring broadcast contents in a TV or Radio programs. The purpose might be to track advertisement or any specific broadcast content. Detail information

[1]Shazam: `https://www.shazam.com/`
[2]SoundHound: `https://www.soundhound.com`

about the application of audio fingerprinting in broadcast monitoring is available in [22]–[25].

- **Royalty Tracking and Copyright Enforcement:** Audio fingerprinting can also be helpful in tracking audio contents with royalty issue so that proper payments can be made in a transparent way. Recommendations on how to implement royalty tracking and copyright enforcement systems are presented by Cano et al. [18] and Gomes et a. [26]. A new Blockchain based audio streaming system called Audius [3] also used audio fingerprinting with the aim of providing decentralized royalty tracking and copyright enforcement system.

- **Other Applications:** In addition to the above three main application areas, audio fingerprinting is also applicable in multi-device self localization [27], indoor localization [28], enhancing speech recognition [29], and water leakage localization [30].

[3] Audius: https://audius.co/

Chapter 3

Related Works

Several methods are proposed to solve the problem of linear speed change. Based on their baseline assumption, these approaches can be grouped into three categories: Phillips approach, Shazam approach and Image Processing approach. Phillips approach is based on the assumption that sign of energy difference between consecutive bands (simultaneously along the tempo and frequency axes) is a property which is robust to distortion [8]. The main assumption in Shazam approach is time-frequency points with highest energy (spectral peaks) can survive severe signal distortion [17]. The last category, image processing approach is motivated by the idea of representing one dimensional audio signal varying over time by its two dimensional time-frequency representation called Spectrogram [31]. Detail review of representative works for each category are presented as follow.

3.1 Phillips Approach

The baseline work under this category, Haitsma and Kalker [8] proposed a novel audio fingerprinting system that segments a given audio into set of frames and applies FFT (Fast Fourier Transform) on each frame to extract audio fingerprints. In this approach, the audio is first segmented into frame of length 0.37 seconds weighted by a Hanning window with an overlap factor of $\frac{31}{32}$ which results in extraction of fingerprints every 11.6 milliseconds. Then, 33 non-overlapping bands in range form 300Hz to 2000Hz are selected for each frame. Finally, 32 bit sub-fingerprints are extracted from each frame using sign of energy difference between two consecutive bands (simultaneously along time and frequency axis). In this fingerprinting scheme, one sub-fingerprint usually does not contain sufficient data to identify an audio clip. The basic unit that contains sufficient data to identify an audio is referred to as a fingerprint- block and it contains 256 sub-fingerprints. As a baseline work, Haitsma and Kalker [8] were able to develop an audio fingerprinting system which is robust to noise. However, it is not robust against linear speed changes larger than $\pm 2\%$.

Considering susceptibility of the Phillips approach to linear-speed change as a potential problem, Haitsma and Kalker originally proposed to handle larger linear-speed changes by storing fingerprint at multiple speeds in the database or extracting query fingerprints at multiple speeds and then perform multiple queries on the database. The main disadvantage of these methods is that either the storage requirement increases or the search speed decreases by several factors. Later in their other work [2], they proposed modified approach to handle larger linear-speed change by increasing number of bands form 33 to 512 and applying auto-correlation function to extract shift invariant audio fingerprints. In addition to auto-correlation, they also introduced low pass filter and down sampling methods to further enhance robustness of the system and summarize extracted information into 32-bit sub-fingerprint. Those modifications enhanced robustness to linear-speed change further to ± 6 %. In another Phillips based work, Seo et al. [32] proposed to enhance a work by Haitsma and Kalker [8] using Fourier-Mellin transform to extract scale invariant audio fingerprints and achieved robustness to linear-speed change up to $\pm 10\%$.

Yao et al. [33] further improved the performance of Phillips based audio fingerprinting system using a new method called turning point alignment and achieved $\pm 30\%$ robustness to time-stretching. Compared to other Phillips based audio fingerprinting systems, the work presented by Yao et al. [33] can be considered as state-of-the art work built on top of an efficient fingerprint matching technique called sampling and counting [34]. However, robustness to pitch related modifications is still a problem.

In general, handling linear speed change is a major challenge for Phillips based audio fingerprinting systems. This difficulty comes from inability to handle modification along pitch axis. Phillips based approaches can handle significant modification along tempo axis by enforcing larger overlap. However, handling modification along pitch axis is difficult because small change in the pitch axis result in shift of energy from one band to another band and this energy shift will cause inevitable alteration on fingerprints.

3.2 Image Processing Approach

The idea of representing one dimensional audio signal varying over time by two dimensional time-frequency representation (audio spectrogram) motivated researchers to apply concepts of image processing as a solution for problems in audio fingerprinting. In image processing based audio fingerprinting, spectrogram of the audio is further transformed into convenient image format to extract fingerprints which are robust

to different kinds of audio distortions including linear-speed change.

Ke et al. [31] proposed an approach to learn global discriminative audio features from its spectrogram using a set of filters. Baluja and Covell [35] modified this approach by decomposing the spectrogram of the audio into small images, then extracted local audio features from top 200 wavelets for each image fragments. Eventough, they laid foundation to apply concepts of image processing to audio domain, both works did not report results on robustness of their systems to speed related audio modifications.

Zhang et al. [4] proposed an approach which is based on an image processing tool called Scale Invariant Feature Transform (SIFT) to extract 128 dimension set of features which are invariant to scaling and translation form spectrogram of the audio. In this approach, time stretching is treated as image stretching along tempo axis and pitch shifting is treated as image translation along pitch axis. Using SIFT, Zhang et al. [4] were able to achieve ± 30 % robustness to speed change. In a recent work under this category, Williams et al. [36] proposed an approach using an alternative feature extraction method called ORB (Orient Fast and Rotate Brief), which is more efficient (computation requirement) than SIFT and achieved ± 10 % robustness to speed change.

In general, using image processing technique to audio domain is more expensive both in computation and storage requirements. The high cost is due to computational complexity of the underline image processing algorithms.

3.3 Shazam Approach

Researches under this category are based on extracting spectral peaks and creating an association between them to uniquely represent an audio. The idea of using association between spectral peaks comes from the assumption that time-frequency points with highest energy can survive sever signal distortion. Wang [17] presented the first audio fingerprinting system which is based on association of two spectral peaks. In this work, first spectral peaks are extracted from spectrogram of the audio by applying rectangular filter and then the spectral peaks are associated into pairs. The association between spectral peaks is defined by storing the frequency component of each peak along with difference of time information between them. Using this approach, Wang [17] achieved significant robustness to additive noise. However, the approach is inherently susceptible to linear speed change.

Fenet et al. [37], developed similar fingerprinting scheme which is based on association

of two spectral peaks. Fenet et al. [37], however, used CQT (Constant Q Transform) instead of STFT (Short Time Fourier Transform). CQT is a well adapted audio signal analysis technique because of its geometrically spaced frequency bins. Moreover, pitch-shifting becomes a translation in the CQT domain. That is, a frequency which is located in bin b will have its pitch-shifted version located in bin $b + K$. To exploit this advantage of CQT, audio fingerprints are constructed using spectral extent and time extent between spectral peaks instead of exact frequency values. By using CQT to compute spectrogram of an audio, Fenet et al. [37] achieved a significant enhancement in identification ratio, 97.% (compared to 80% by Wang [17]). The experiment is conducted using 120 hour audio corpus from real radio broadcast. This approach, however, is also susceptible to audio modification along tempo axis and this made it still prone to linear-speed change.

Six and Leman [5] proposed fingerprint extraction technique based on the association of triple spectral peaks by adopting CQT from the work of Fenet et al. [37] and concept of triple spectral peaks from the work of Artz et al. [38]. In this new approach 24 to 60 spectral peaks are extracted per one second audio and those spectral peaks are grouped to triplets based on their temporal occurrence to generate speed change invariant audio fingerprints. Six and Leman [5] were able to achieve robustness upto $\pm$8% linear-speed change. However, according to the investigation conducted by Ouali et al. [3], -20% to 23% speed modification is common. Therefore, $\pm$8% robustness is not enough.

Sonnleitner and Widmer [6] adopted an approach from the field of blind astronomy [39] to develop robust audio fingerprinting system based on the association of four spectral peaks and achieved $\pm$30% robustness to linear speed change. Increasing the number of spectral peaks to four, however, has two main problems: i) it increases computational and storage requirement of the system; and, ii) it will degrade the robustness of the system to other modifications as the increase in the number of spectral peaks leads to extraction of audio fingerprints with high entropy which are difficult to reproduce.

Chapter 4

Proposed Approach

General architecture of proposed approach is depicted in Figure 4.1. This architecture consists of three core modules:

- Fingerprint Extraction
- Reference Fingerprint Database
- Fingerprint Matching

Reference fingerprints are extracted from audio collections and stored in reference fingerprint database along with their respective audio ID. Later, while recognizing unknown audio, fingerprint matching module return an audio ID based on fingerprints extracted form short query audio. Detail information about these three modules is given in the following three sub-sections.

4.1 Fingerprint Extraction

This module extract perceptually relevant characteristics of an audio and transform extracted information into audio fingerprints. As shown in Figure 4.2, proposed fingerprint extraction module consists of four main stages: Preprocessing, Spectral Estimation, Peak Extraction and Fingerprint Formation. The proposed fingerprint extraction technique is a recurrent technique among Shazam based audio fingerprinting systems with a modification on the process of fingerprint formation.

4.1.1 Preprocessing

In this stage a given audio file is re-sampled with sampling rate (sr) of 7 KHz and down-mixed to Mono PCM (Pulse Code Modulation). Sampling rate of 7 KHz is chosen for two reasons [8]: (i) it can cover perceptually relevant spectral ranges; and, (ii) to get sufficient time series data from the given audio file. Monaural Sound format is chosen for its advantage of being easy to manipulate. Five second long Monaural

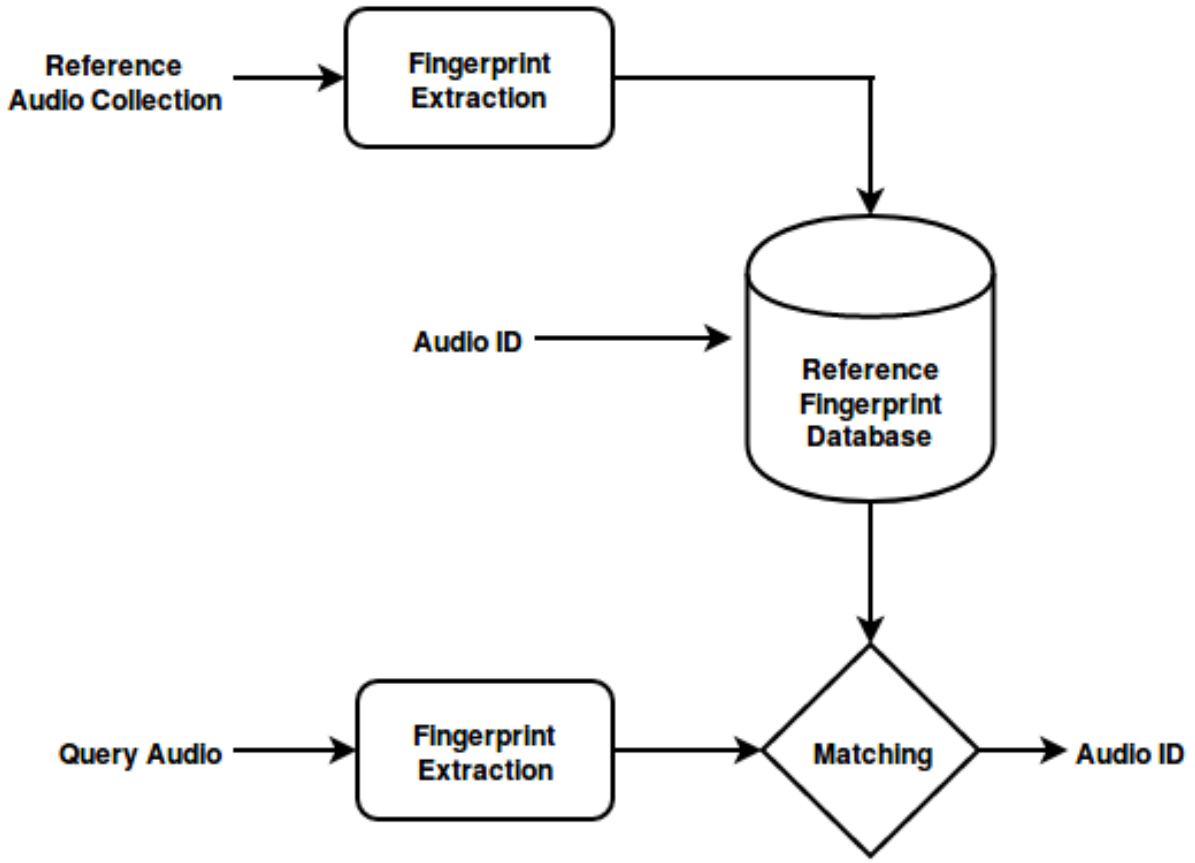


FIGURE 4.1: General Architecture for Audio Fingerprinting Systems adopted form [18]

waveform sampled at 7KHz from a song called "Aynih" by Aster Awoke is plotted in Figure 4.3.

4.1.2 Spectral Transformation

Time series audio data with Mono PCM format from preprocessing stage is transformed into its corresponding time-frequency representation by applying Short-Time Fourier Transform (STFT). The STFT represents a signal in the time-frequency domain by computing discrete fourier transforms (DFT) over short overlapping windows. STFT implementation provided by McFee et al. [40] is used with $n\_fft = 1024$ and $hop\_length = 32$. Where, n_fft defines number of frequency bins and hop_length defines number of audio samples between adjacent STFT columns. This function (STFT) returns a complex-valued matrix D, where:

- $(1 + n\_fft/2, n\_frames)$ defines shape of D. Where, n_frames is computed total number of frames which can be determined from duration of the audio ($duration$),

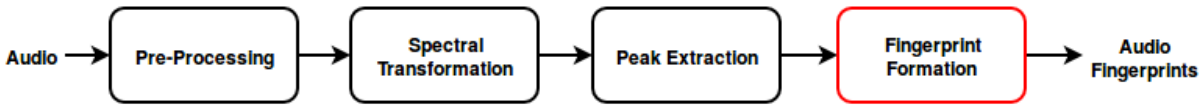


FIGURE 4.2: Proposed Fingerprint Extraction Method

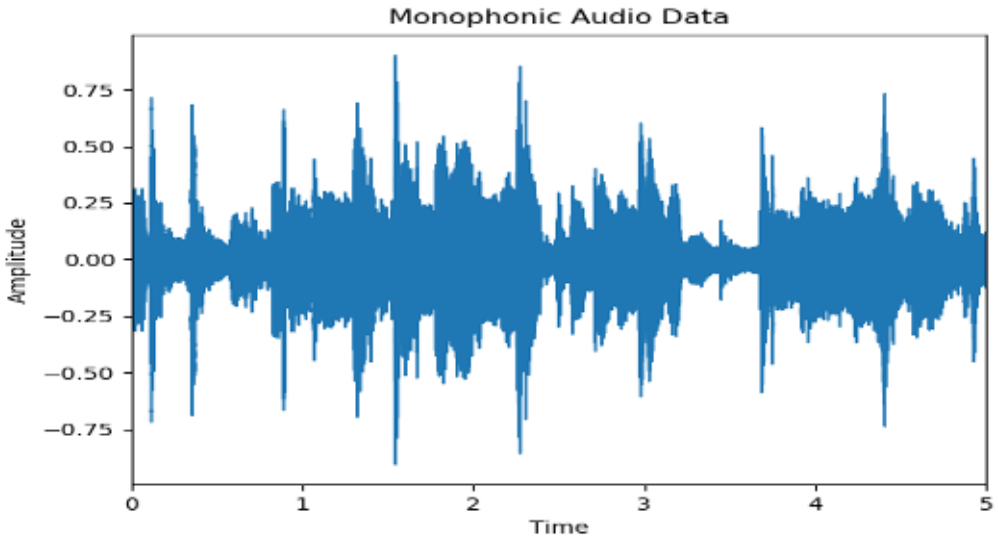


FIGURE 4.3: Monaural waveform of a song called "Aynih" by Aster Awoke (Five seconds duration)

sampling rate (sr) and hop_length using the equation $n\_frames = \left\lceil \frac{sr * duration}{hop\_length} \right\rceil$.

- $\left|D[f,t]\right|$ is the magnitude of frequency bin f at frame t, and
- $\angle(D[f,t])$ is the phase of frequency bin f at frame t.

Only the magnitude information ($|D|$) is used for this research. $|D|$, Power spectrogram of an audio, represents magnitude of each frame t at different frequencies. While computing $|D|$, choosing values for n_fft and hop_length is a key factor which affects overall computational and storage requirement of the system. Setting n_fft to a power of two is recommended while computing fast Fourier transform (FFT) and values ranging from 512 to 2048 are well adapted for music signals [40]. For this research $n\_fft = 1024$ and $hop\_length = 32$ are chosen to get sufficient spectral data for a given time series audio data based on a recommendation of McFee et al. [40]. Increasing n_fft beyond 1024 increases the frequency resolution, however, it will increase overall computational demand. Smaller hop_length values results in an increase in the total

number of frames, however, lowering it below 32 will have negative impact on computational cost.

With the above chosen parameter values, the frequency resolution becomes 6.82 Hz (513 frequency bins have equal width of 6.82Hz) which is equivalent to 146.62 millisecond time resolution and the shape of D for one second audio becomes (513, 219), interpreted as 219 audio frames each consists of 513 spectral information. Spectrogram representation of time series audio signal displayed in Figure 4.3 is depicted in Figure 4.4.

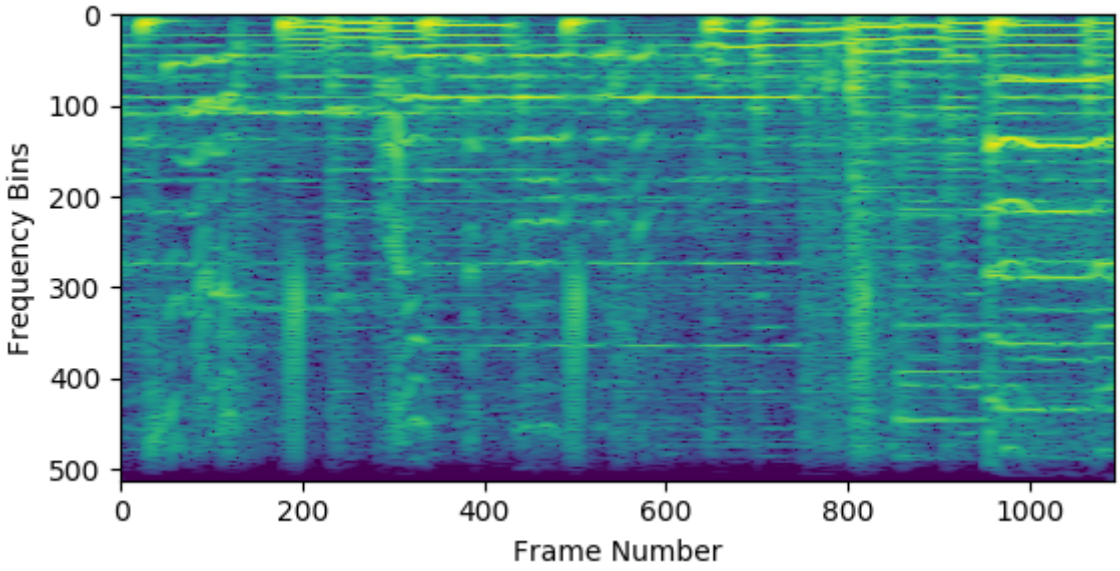


FIGURE 4.4: Spectrogram of a song called "Aynih" by Aster Awoke (Five seconds duration)

4.1.3 Peak Extraction

Once the spectrogram of the audio is computed, a rectangular maximum filter and square minimum filter are applied to extract temporally localized spectral peaks. As it is clearly stated by Wang [17], extracting temporally localized spectral peaks instead of global spectral peaks has an advantage that distant events will not affect fingerprints. Both the maximum and minimum filters are applied independently on the spectrogram of the audio. Peaks extracted by a maximum filter are considered as spectral peaks for further processing only if they are not detected by the minimum filter. Here, the minimum filter is used to reduce number of spectral peaks extracted from audio region with uniform energy, as a spectrogram might contain regions of uniform magnitude. Examples of such cases would be silence, clicks, or digitally created tones with

identical magnitudes [6]. Figure 4.5 and Figure 4.6 shows spectral peaks extracted from spectrogram shown in Figure 4.4. Here, a spectral peak P is defined as a point (Px, Py), where Px represent tempo information and Py represents pitch information of spectral peak P. Figure 4.5 shows spectral peaks extracted using a square minimum filter with dimension of 3x3 (spans 3 frames along tempo axis and 3 bins along pitch axis to pick one time-frequency point with a minimum energy) and maximum rectangular filter with dimension of 150x75 (spans 150 frames along tempo axis and 75 bins along pitch axis to pick one time-frequency point with a maximum energy). Whereas, Figure 4.6 shows spectral peaks extracted from the same spectrogram using a larger filter dimension, in this case a minimum filter size of 3x3 and maximum filter size of 300x150 is used. Looking at Figure 4.5 and Figure 4.6, using a maximum filter with smaller dimension results in an increase in total number of spectral peaks and this will have a negative effect on both storage and computation requirement. Whereas, using a maximum filter with larger dimension results in fewer number of spectral peaks and using a maximum filter with larger dimensions might have its own disadvantage while trying to uniquely represent a given audio. Therefore, it is necessary to choose for a convenient filter dimensions.

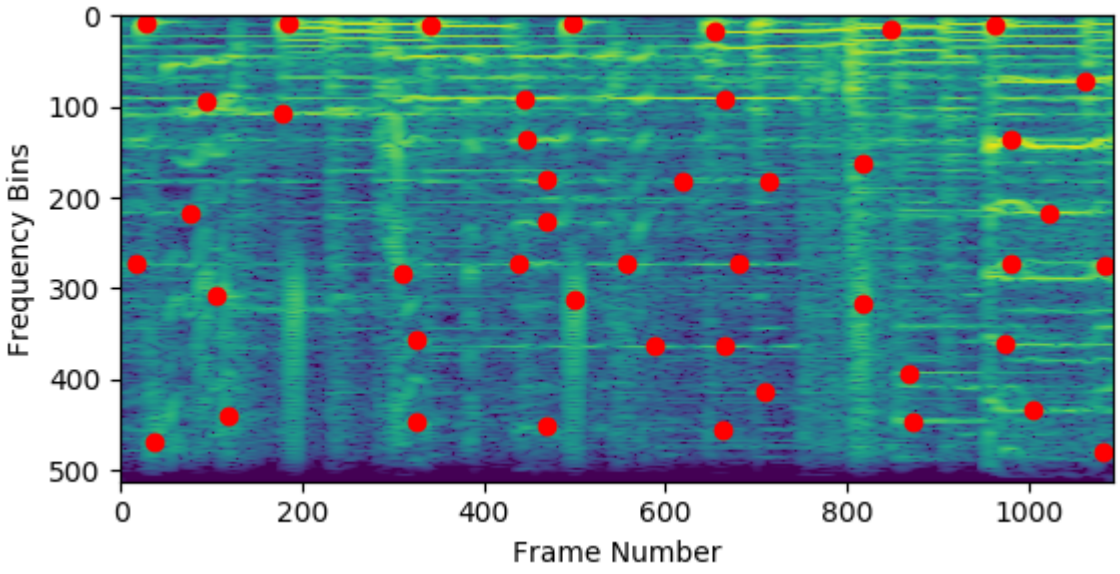


FIGURE 4.5: Spectral peaks extracted from a song called "Aynih" by Aster Awoke (Five seconds duration) using maximum filter dimension of 150x75 and minimum filter dimension of 3x3: the x- axis represents tempo and the y-axis represents pitch

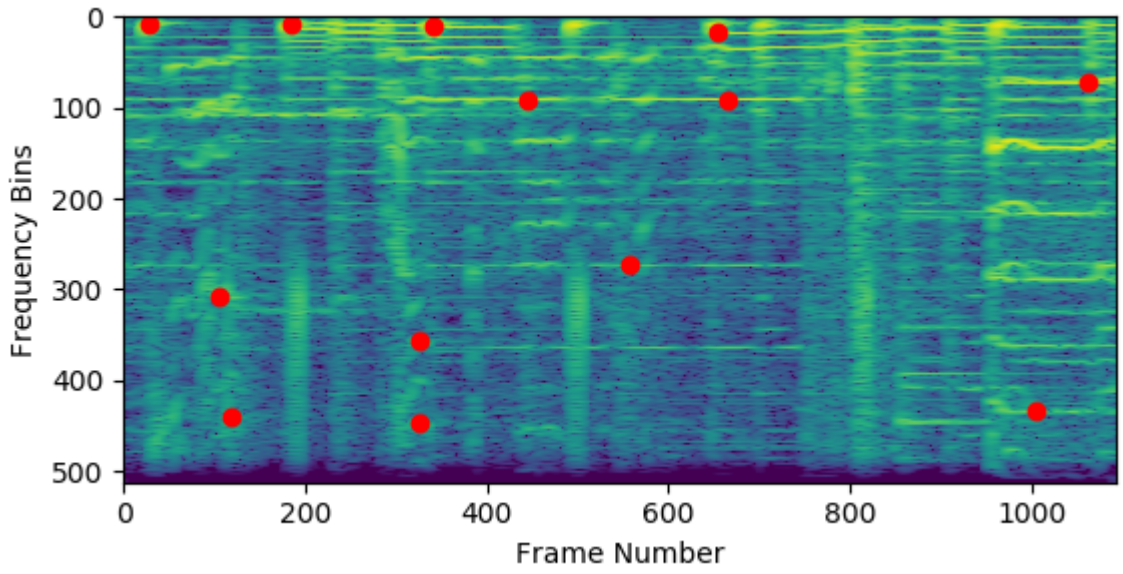


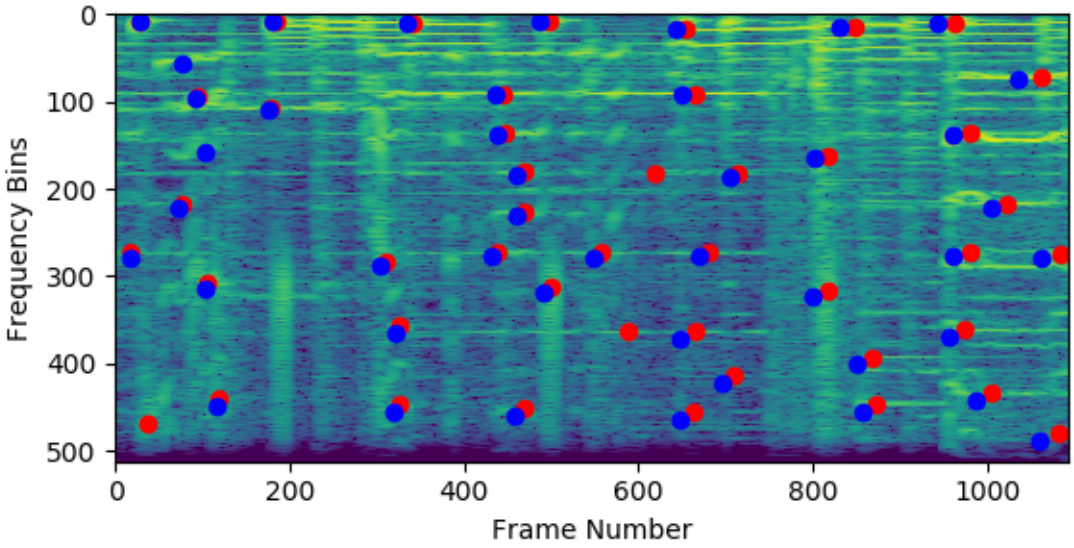
FIGURE 4.6: Spectral peaks extracted from a song called "Aynih" by Aster Awoke (Five seconds duration) using maximum filter dimension of 300x150 and minimum filter dimension of 3x3: the x- axis represents tempo and the y-axis represents pitch

4.1.4 Fingerprint Formation

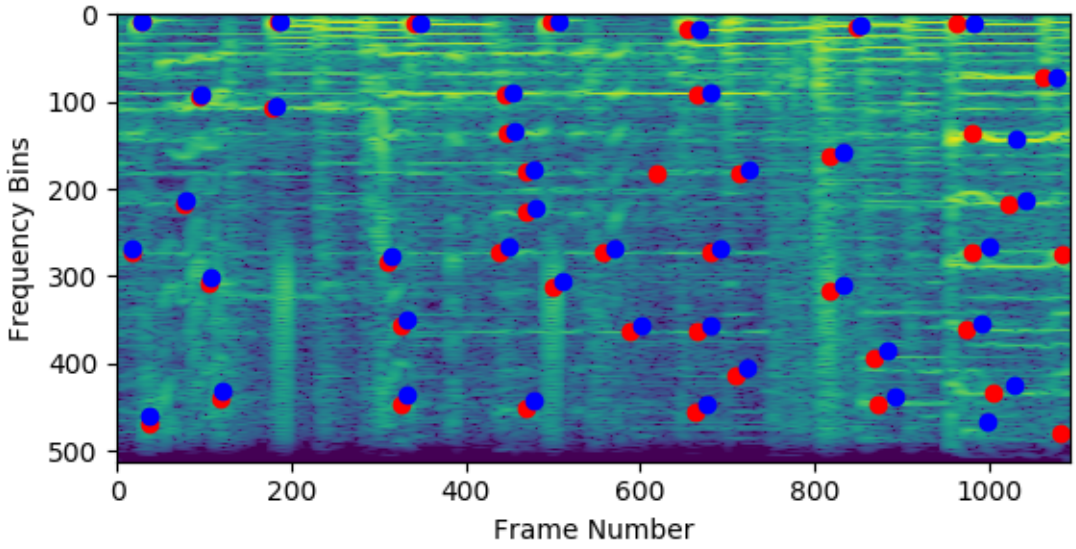
This is the final stage under fingerprint extraction module, and it is responsible for generating audio fingerprints which are invariant to linear speed change. However, before going to details of this stage, it is better first to analyze effects of linear speed change on spectral peaks. As shown in Figure 4.7, linear speed change affects the dynamics of the audio simultaneously along the pitch and tempo axis. Figure 4.7a depicted spectral peaks extracted from original audio (red dots) along with spectral peaks extracted from faster audio (blue dots). As it is shown in Figure 4.7a, playing an audio at a faster speed (2%) caused spectral peaks to move down along pitch axis (an increase in pitch) and to the left along tempo axis (an increase in tempo). Whereas, playing an audio with slower speed (-2%) caused the exact opposite effect and it is shown in Figure 4.7b. In both cases (playing an audio faster or slower), spectral peaks exhibited both translation and scaling simultaneous along tempo and pitch axis. This effect can be summarized as ***applying speed change result in translation and scaling between spectral peaks simultaneous along tempo and pitch axis. However, while looking at the pattern between spectral peaks for both modified and original audio it is more or less similar.***

Based on this analysis, an audio fingerprint extraction technique which is robust to both translation and scaling is proposed using geometric hashing, specifically triple

point geometric hashing. Geometric hashing is a model-based pattern recognition technique for detecting patterns which can be partially overlapping or partly occluded [41]. To look at the details of proposed fingerprint extraction method, this stage is further decomposed into three sub stages: Target Zone Assignment, Triplet Grouping and Hash Formation.



(A) Spectral peak extracted from original audio(red dots) vs spectral peaks extracted from 2% faster audio (blue dots).



(B) Spectral peak extracted from original audio(red dots) vs spectral peaks extracted from -2% slower audio (blue dots).

FIGURE 4.7: Effects of linear speed change on spectral peaks

Target Zone Assignment

The first step in fingerprint formation, is to assign a target zone for each spectral peak. As shown in Figure 4.8, a target zone is assigned to spectral peak $P1$. The height of the target zone covers the whole spectral region along pitch axis and its width along tempo axis is defined by $(max\_frame - min\_frame)$, where max_frame is maximum frame number included in the target zone and min_frame is minimum frame number included in the target zone. While assigning target zone, it is important to make sure that $P1$ is outside the target zone. Both distance of center of the target zone from $P1$ and width of target zone affects discriminatory power of fingerprints and computational requirement of the system. Increasing width of the target zone increases total number of audio fingerprints and this has a positive impact on ability of the system to uniquely identify a given audio but has cost in both computation and storage requirement. The distance between $P1$ and center of the target zone is also another determining factor on the discriminatory power of fingerprints. If small distance is used it will be difficult to reproduce audio fingerprints in the presence of linear speed modification. On the other hand, if large distance is used the temporal locality of fingerprints will not be kept.

Another important issue in target zone assignment is the use of different target zone size in reference and query fingerprint extraction. The size of target zone used for query fingerprint extraction must be larger than the size of target zone used for extracting reference fingerprints [6]. The relation between size of query target zone and reference target zone depends on tolerance of the system to modification along tempo axis, which is defined by $t\_tolerance$. For example, if the system is designed with tolerance of 0.3 along tempo axis, this means it can handle up 30% tempo modification in the query audio. Based on this, the size of the target zone used for extracting fingerprints from query audio is related with size of target zone used for extracting fingerprints from reference audios using $t\_width\_query = \frac{t\_width\_reference}{1 - t\_tolerance}$, where $t\_width\_reference$ defines width of target zone for reference audio t_width_query width of target zone used for query audios. However, in both cases the height of the target zone span over whole spectral range along the pitch axis.

Triplet Grouping

Once a target zone is assigned to spectral peak $P1$, triplets $(P1, P3, P2)$ are formed by picking $P2$ and $P3$ from a target zone based on a condition stated in Equation 4.1. The condition stated in Equation 4.1 is required for two reasons: i) as it is stated above, geometric hashing is model base pattern recognition technique so it requires pre-defined

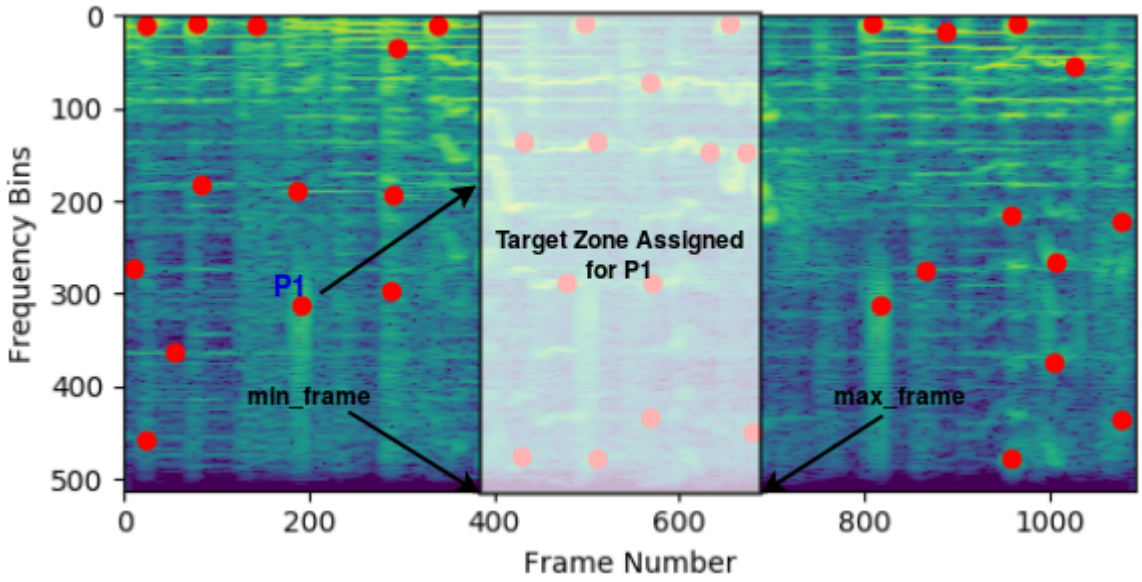


FIGURE 4.8: Target zone assignment

pattern; and, ii) this condition result in hash values which are normalized between 0 and 1, and this has its own advantage later while storing and comparing hashes. In general, the actual number of valid triplets depends on the nature of the audio data, but the maximum number of triplets which can be formed is $\binom{n}{2}$, where n is the total number of peaks in the target zone.

$$\begin{aligned} P1x < P3x < P2x \\ P1y < P3y < P2y \end{aligned} \tag{4.1}$$

Geometric Hashing

Geometric hashing is a model-based pattern recognition technique for detecting patterns which can be partially overlapping or partly occluded [41]. Given a two dimensional object, applying geometric hashing starts with identifying relevant feature points. In this work, an audio spectrogram is treated as a two dimensional object and spectral peaks as relevant feature points which are going to be used while generating geometric hashes. Once the relevant feature points are identified, the next step is to define a pattern between those relevant feature points and generate hashes based on the defined pattern. For this case a pattern between spectral peaks which can satisfy a condition stated in Equation 4.1 is chosen. According to Wolfson and Rigoutsos [42], geometric hashing has two main advantages when it comes to pattern recognition: i) Its ability in recognizing patterns even if the underling two dimensional object under go different transformations such as scaling , translation and rotation; and, ii) Geometric

hashing is inherently parallel. In fact, with minimal communication and maintenance costs, the underlying data structure can be easily decomposed and shared among a number of cooperating processors.

Coming back to generating audio fingerprints, triple point geometric hashing (geometric hashing based on a pattern defined by three geometric points) is applied to generate audio fingerprints which are invariant to linear speed change. As shown in Figure 4.9, spectral peaks formed a triplet $(P1, P3, P2)$ are transformed in to a new rectangular coordinate system. In this new coordinate system, $P1$ is placed at the origin $(0, 0)$ and $P2$ at $(1, 1)$. $P3x\_new$ and $P3y\_new$ are computed from original values $P1$, $P2$ and $P3$ by applying Equation 4.2. These new values of $P3$, $(P3x\_new, P3y\_new)$ are mathematically proven to be invariant to both scaling and translation as long as the applied scaling and translation between $P1$, $P2$ and $P3$ is uniform [42]. Therefore, by using $(P3x\_new, P3y\_new)$ values it is possible to represent a given audio file in a way which is invariant to linear speed change. Hereafter, $(P3x\_new, P3y\_new)$ are interchangeably referred to as audio hashes or audio fingerprints.

$$\begin{aligned} P3x\_new &= \frac{P3x - P1x}{P2x - P1x} \\ P3y\_new &= \frac{P3y - P1y}{P2y - P1y} \end{aligned} \tag{4.2}$$

In a nutshell, the applied fingerprint extraction technique is scanning predefined pat-

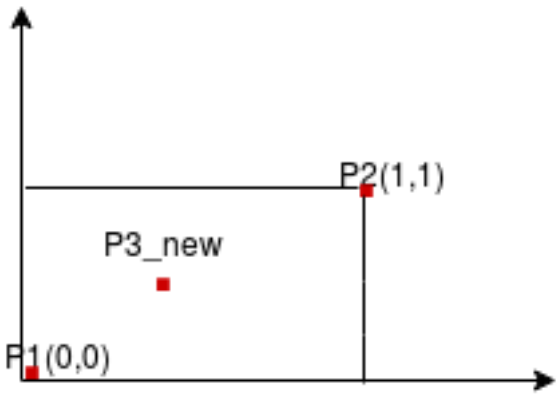


FIGURE 4.9: Geometric Hashing

tern between spectral peaks and representing the defined pattern uniquely using its geometric hash based on Equation 4.2.

To look at the details in applying triple point geometric hashing, an example is given

based on Figure 4.10. In this figure, there are 10 spectral peaks labeled from $P1 - P10$. The tempo and pitch value of each spectral peak is given in Table 4.1. To generate audio fingerprints by following the above procedures, let us follow the following steps:

Step 1): Sort extracted spectral peaks in an increasing order based on their frame number.

Step 2): Pick the first spectral peak as a root peak, assign a target zone to it and identify all spectral peaks to be included in the assigned target zone.

Step 3): Identify all triplets (association between three spectral peaks(the root peak and two other spectral peaks from the assigned target zone) which can satisfy a condition stated by Equation 4.1.

Step 4): Generate hashes for each valid triplet based on Equation 4.2. Finally, go back to Step 2 and repeat following Steps using the next spectral peak as root peak.

Applying this on spectral peaks depicted in Figure 4.10, the first step is to sort spectral peaks in an increasing order based on their frame number and pick the first peak which is $P1$. Moving to Step 2, a target zone is assigned to $P1$. Here, the target zone is set to be 0.2 seconds (using $n\_frames = \left\lceil \frac{sr*duration}{hop\_length} \right\rceil$, this value is computed to 44 frames) away from the root peak $P1$ and its width (t_width) is set to one second duration (219 frames). Those parameters of target zone (0.2 second away from the root peak and 1 second duration width) are selected to get sufficient audio fingerprints which can uniquely represent a given audio and they are chosen experimentally considering the overall computational demand, storage requirement as well as discriminative power of audio fingerprints. Using those value, the minimum frame number to be included in a target zone assigned to $P1$ will be 79 ($min\_frame = P1\_x + 44$) and the maximum frame number will be 298 ($max\_frame = min\_frame + t\_width$). This results in spectral peaks $P2$, $P3$, $P4$, $P5$, $P6$ and $P7$ to be inside a target zone assigned to $P1$. However, while moving forward to Step 3, which is triplet grouping there is no combination of spectral peaks which can satisfy the condition stated in Equation 4.1. Therefore, hash generation using $P1$ as a root peak will be halted here and the process will go back to Step 2 to pick the next root peak, which is $P2$. Following a similar approach, the minimum frame number to be included in the target zone assigned for $P2$ is computed to 99 ($min\_frame = P2\_x + 44$) and the maximum frame number computed to 318 ($max\_frame = min\_frame + t\_width$). This results in spectral peaks $P4$, $P5$, $P6$, $P7$ and $P8$ to be inside the target zone assigned to $P2$. Moving to Step 3, three triplets are found which can satisfy Equation 4.1, which are $(P2, P4, P6)$, $(P2, P4, P8)$ and $(P2, P7, P8)$.

The next step, Step 4 is to hash each triplets according to Equation 4.2 and similar process will be applied for the rest spectral peaks as root peak. In this case the next root peak is $P3$.

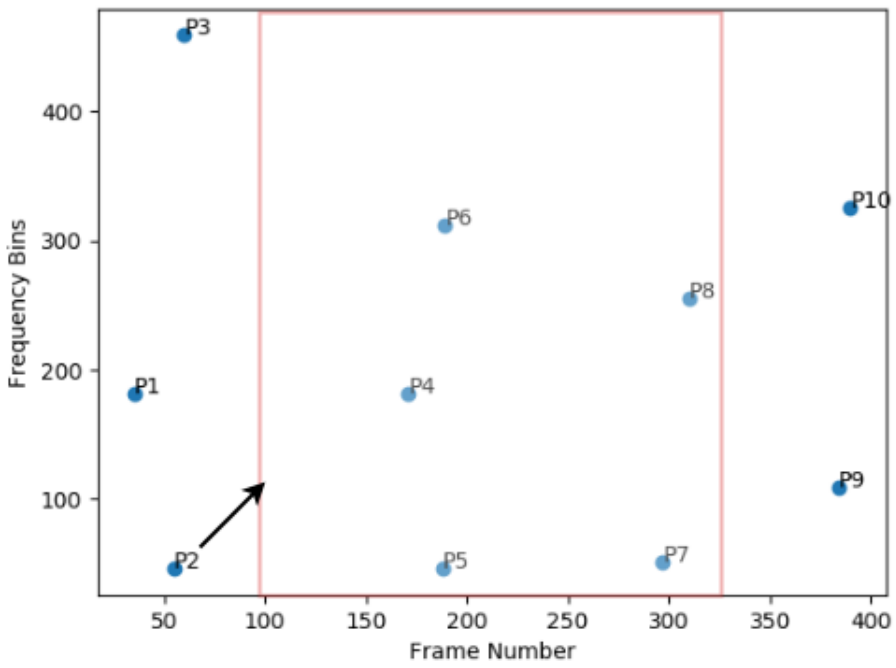


FIGURE 4.10: Example on triple point geometric hashing

Spectral Peak	Peak Value (frame number,frequency bin)	Spectral Peak	Peak Value (frame number,frequency bin)
$P1$	(35, 182)	$P6$	(189, 312)
$P2$	(55, 46)	$P7$	(297, 51)
$P3$	(60, 459)	$P8$	(310, 256)
$P4$	(171, 182)	$P9$	(384,109)
$P5$	(188, 46)	$P10$	(390, 326)

TABLE 4.1: Tempo and pitch information of spectral peaks shown in Figure 4.10: Here a peak P is represented as a point (P_x, P_y), where P_x represent tempo information(frame number) and P_y represent its pitch information(frequency bin number).

4.2 Reference Fingerprint Database

Audio hashes extracted from reference audio collections are stored in a reference fingerprint database. The reference fingerprint databased is developed using R-Tree[1] and persistent dictionary based python data structure called Shelf[2]. R-Tree is a dynamic indexing data structure widely used for spatial searching [43]. While constructing the reference fingerprint database, the R-Tree is used to store extracted audio hashes and their associated raw data (original values of $P1$ and $P2$) are stored in the shelf. R-tree is chosen because of its advantage in accelerating nearest neighbor search and Shelf is chosen for its advantage in indexing stored data (avoids brute force searching while retrieving raw data of audio hashes later in the matching phase).

4.3 Fingerprint Matching

In the fingerprint matching, hashes are extracted from potentially distorted query audio and matched with reference audio hashes. This module is implemented in two phases as follow.

4.3.1 Matching Hashes

The matching phase consists of two stages: retrieving candidate matches followed by filtering. In the first stage, candidate matches are retrieved by applying nearest neighbor search on R-Tree and for each candidate match their corresponding raw data (original $P1$ and $P2$ values of each matching reference hash) along with their Audio ID are retrieved from Shelf. This results in set of candidate matches and their associated raw data with potentially large number of incorrect matches. The second stage reduce number of incorrect matches by applying set of two filters, similar to filtering technique proposed by Sonnleitner and Widmer [6] with minor modification.

The first filter defined in Equation 4.3, checks as to whether the modification of query hashes relative to their matching reference hashes in both tempo and pitch axis is with in the allowed ranges or not.

$$\begin{aligned} min\_t\_mod < t\_mod < max\_t\_mod \\ min\_f\_mod < f\_mod < max\_f\_mod \end{aligned} \tag{4.3}$$

[1] R-Tree:https://pypi.org/project/Rtree/
[2] Shelf: https://docs.python.org/3/library/shelve.html

Where, t_mod (Equation 4.4) is estimated modification of query hash along tempo axis relative to its matching reference hash, f_mod (Equation 4.5) is estimated modification of query hash along pitch axis relative to its matching reference hash, min_t_mod and max_t_mod (Equation 4.6) defines minimum and maximum tempo modification that the system can handle and min_f_mod and max_f_mod (Equation 4.7) defines minimum and maximum pitch modification that the system can handle.

$$t\_mod = \frac{P2x\_query - P1x\_query}{P2x\_ref - P1x\_ref} \tag{4.4}$$

$$f\_mod = \frac{P2y\_query - P1y\_query}{P2y\_ref - P1y\_ref} \tag{4.5}$$

$$\begin{aligned} min\_t\_mod &= \tfrac{1}{1+t\_tolerance} \\ max\_t\_mod &= \tfrac{1}{1-t\_tolerance} \end{aligned} \tag{4.6}$$

$$\begin{aligned} min\_f\_mod &= \tfrac{1}{1+f\_tolerance} \\ max\_f\_mod &= \tfrac{1}{1-f\_tolerance} \end{aligned} \tag{4.7}$$

The second filter defined in Equation 4.8, checks as to whether positional mismatch between query hashes and their matching reference hashes is within allowed range or not.

$$\left|P1y\_query - P1y\_ref * f\_mod\right| < p\_max \tag{4.8}$$

Where, p_max is a constant value which defines maximum acceptable positional mismatch between query hash and matching reference hash.

The above two filters remove a significant number of incorrect matches and only set of potential candidate matches are left with different Audio ID. To decide the final correct Audio ID, these potential candidate matches are further processed by the verification stage.

4.3.2 Verification

In this phase potential candidate matches are grouped based on their Audio ID and matches in each group are sorted in an increasing order based on their temporal occurrence. For a matching Audio ID hashes are expected to exhibit sequential pattern along tempo axis. Therefore, if a histogram of $(P1x\_query - P1x\_ref * t\_mod)$ is calculated for each group, the histogram for matching audio will have a maximum value around c, where c is the offset of query audio relative to its reference. The ID of a histogram

(Audio ID) which contain a bin with relatively maximum value relative to other histograms is returned as a matching Audio ID. This verification technique is adopted from the baseline Shazam based audio fingerprinting system [17].

Chapter 5

Experiments

In this Chapter datasets, tools and experimental scenarios used to evaluate the proposed approach along with the obtained results are discussed. Experiments are designed to evaluate the proposed approach based on the requirements discussed in Section 2.3.

5.1 Test Data Preparation

An open source audio processing tool box called eu-Audio, is developed to prepare test audio corpus with required audio effects. The developed tool is freely available on GitHub, https://github.com/Efode-r2d2/eu-Audio, under GNU General Public License v3.0. Using this tool, 29,600 query audios with an average duration of 30 seconds are extracted from 200 reference audio files. Meta-data of reference audio files is also available on https://github.com/Efode-r2d2/Efode/blob/master/Test_Data/Reference_Audios.txt. Summary on extracted query audios is presented as follow:

- **Query Audios with Additive White Noise:** 3,600 audio excerpts with additive noise, 2 query audios per reference audio, are extracted with signal to noise ratio (SNR) between -20dB to 20dB in a 5dB step.
- **Linear Speed Modified Query Audios:** 12,400 linear speed modified audio excerpts, 2 query audios per reference audio, are extracted with linear speed modification between -30% to 30% with 2% step. Here, the speed modification is done by applying re-sampling on the original audio using the re-sampling technique provided by McFee et al. [40]. In the employed modification scheme, positive speed changes results in an increase in tempo as well as pitch values of the audio and a decrease in the total duration of the audio. Whereas, negative speed changes results in the exact opposite effects. For example, both 2% and -2% speed change results in 2% difference between the duration of original and modified audio. However, for a 2% speed modification the duration of the modified audio is shorter in length by 2% compared to length of the original audio and for -2%

speed modification the duration of the modified audio is longer in length by 2% compared to length of the original audio.

- **Time Stretched Query Audios:** 6,800 time stretched audio excerpts, 2 query audios per reference audio, are extracted with time stretching between -16% to 16% with 2% step. Time stretching is done by applying Phase Vocoder [44] implementation provided by McFee et al. [40]. In this modification scheme, positive time stretching results in an increase in tempo without affecting the pitch of the audio and a decrease in the total duration of the audio. Whereas, negative time stretching results in the exact opposite effects. For example, both 2% and -2% time stretching results in 2% difference between the duration of original and modified audio. However, for a 2% time stretching the duration of the modified audio is shorter in length by 2% compared to length of the original audio and for -2% time stretching the duration of the modified audio is longer in length by 2% compared to length of the original audio.

- **Pitch Shifted Query Audios:** 6,800 pitch shifted audio excerpts, 2 query audios per reference audio, are extracted with pitch shifting between -16% to 16% with 2% step. Pitch shifting is also achieved by employing Phase Vocoder [44] implementation provided by McFee et al. [40]. Positive pitch shifting raises up pitches of the audio with out affecting its tempo and negative pitch shifting lowers down pitches of the audio. Unlike speed change and time stretching pitch shifting doesn't cause any change on the duration of the audio.

5.2 Experimental Setup

All modules in the developed audio fingerprinting system are implemented in Python using version 0.4.0 of Librosa[1]. Librosa is a Python package for audio and music signal processing and it provides implementations of a variety of common functions used throughout the field of music information retrieval. The hardware and software specification of the machine used to run all experiments is given in Table 5.1.

5.3 Experimental Scenarios

Several experiments are conducted to evaluate the proposed audio fingerprinting system. Based on their intended goal, those experiments are grouped in to three major categories:

[1]Librosa: https://librosa.github.io/librosa/

Specification of Machine Used For Experiments	
Manufacturer	HP
Model	HP Pavilion 15-au018wm
Processor	Intel® Core™ i7-6500U CPU @ 2.50GHz x 4
Memory	8 GB DDR4-2133 SDRAM (1 x 8 GB)
Operating System	64 bit Ubuntu 16.04 LTS

TABLE 5.1: Machine Specification

- **Experiment I:** consists of experiments conducted to evaluate robustness of proposed audio fingerprinting system.
- **Experiment II:** aimed at evaluating the reliability of proposed audio fingerprinting system.
- **Experiment III:** consists of experiments to evaluate the granularity and time-complexity of proposed audio fingerprinting system.

In all of the above three experimental scenarios, the proposed audio fingerprinting system is compared with two other previous works, Shazam and Panako. For Shazam our own implementation based on a work presented by Wang [17] is used and it is available on GitHub, `https://github.com/Efode-r2d2/Shazam`, under GNU General Public License v3.0. Whereas, for Panako an open source implementation by Six and Leman [5] is used. The source code for Panako is also available on GitHub, `https://github.com/JorenSix/Panako`, under GNU Affero General Public License.

5.3.1 Experiment I: Robustness

Experiments under this category aimed at evaluating ability of the proposed approach to accurately identify fingerprinted audio regardless of the level of audio modification applied on a query audio. Prior to conducting robustness experiment, the reference fingerprint database is populated with audio fingerprints extracted from 150 reference audios (of the 200 reference audios 150 audios are fingerprinted and the rest 50 are reserved for reliability test). The proposed approach emphasized on robustness to linear speed change, however, the robustness of the system to other audio modifications is also evaluated. For the ease of discussion, conducted experiments under this category are further classified into four groups:

- **Experiment I-A: Robustness to Additive Noise**
  In this experiment, 2,700 query audios are processed with noise level between -20dB to 20dB with 5dB step. Experimental scenario for this experiment is shown in Table 5.2.

- **Experiment I-B: Robustness to Linear Speed Change**
  In this experiment, 9,300 linear speed modified query audio with linear speed modification between -30% to +30% with 2% step are processed. Experimental scenario for this experiment is shown in Table 5.3.

- **Experiment I-C: Robustness to Time Stretching**
  In this experiment, 5,100 time-stretched query audios with time-stretching between -16% to +16% with 2% step are processed. Experimental scenario for this experiment is shown in Table 5.4.

- **Experiment I-D: Robustness to Pitch Shifting)**
  In this experiment, 5,100 pitch-shifted query audios with pitch shifting between -16% to +16% with 2% step are processed. Experimental scenario for this experiment is shown in Table 5.4.

	Additive White Noise						
	-20dB	-15dB	..	0dB	...	15dB	20dB
Shazam	300 queries	300 queries	300 queries	300 queries	300 queries	300 queries	300 queries
Panako	300 queries	300 queries	300 queries	300 queries	300 queries	300 queries	300 queries
Proposed	300 queries	300 queries	300 queries	300 queries	300 queries	300 queries	300 queries

TABLE 5.2: Experimental Scenarios for Experiment I-A (Robustness to Additive Noise): all query audios are 30 second long and total of 2700 query audios are processed by each system (Proposed, Shazam and Panako)

	Linear Speed Change						
	-30%	-28%	..	0%	...	28%	30%
Shazam	300 queries	300 queries	300 queries	300 queries	300 queries	300 queries	300 queries
Panako	300 queries	300 queries	300 queries	300 queries	300 queries	300 queries	300 queries
Proposed	300 queries	300 queries	300 queries	300 queries	300 queries	300 queries	300 queries

TABLE 5.3: Summary of experimental scenario for Experiment I-B (Robustness to Linear Speed Change): all query audios are 30 second long and total of 9300 query audios are processed by each fingerprinting system (Proposed, Shazam and Panako)

	Pitch Shifting/Time Stretching						
	-16%	-14%	..	0%	...	14%	16%
Shazam	300 queries	300 queries	300 queries	300 queries	300 queries	300 queries	300 queries
Panako	300 queries	300 queries	300 queries	300 queries	300 queries	300 queries	300 queries
Proposed	300 queries	300 queries	300 queries	300 queries	300 queries	300 queries	300 queries

TABLE 5.4: Summary of experimental scenario for Experiment I-C and I-D (Robustness to Time-Stretching and Pitch-Shifting): all query audios are 30 second long and total of 5100 query audios are processed by each fingerprinting system (Proposed, Shazam and Panako)

5.3.2 Experiment II: Reliability

This experiment aimed at evaluating the reliability of of the proposed approach in a way that, if fingerprints for an audio has not been stored in a reference fingerprint database, the audio doesn't have to be identified as a match. To evaluate the reliability of the approach, 7400 query audios (3100 linear speed modified, 900 with additive noise, 1700 with time stretching and 1700 with pitch shifting) extracted form 50 reference audios which have not been fingerprinted, are processed.

5.3.3 Experiment III: Granularity and Time-Complexity

This final experiment aimed at evaluating the effect of length of query audio (Granularity) and analyzing response time of the system (Time-Complexity). In this experiment, 1,800 query audios with additive noise are processed based on experimental scenario presented in Table 5.5. Only query audios with additive noise in a range from -5dB to 2dB are used for a reason that all three audio fingerprinting systems (Proposed System, Shazam and Panako) exhibit comparable recognition rate only to noise added query audios as shown in Figure 5.1 (result for Experiment I-A: Robustness to Additive Noise).

5.4 Parameter Selection

In all the above experimental scenarios the parameters used while setting up the proposed approach are the same and detail information on the values of each parameter is given in this section. Parameter selection starts from the preprocessing stage, in this

Query Audio Length	Shazam	Panako	Proposed
5 seconds	1,800 queries	1,800 queries	1,800 queries
10 seconds	1,800 queries	1,800 queries	1,800 queries
15 seconds	1,800 queries	1,800 queries	1,800 queries
20 seconds	1,800 queries	1,800 queries	1,800 queries
25 seconds	1,800 queries	1,800 queries	1,800 queries
30 seconds	1,800 queries	1,800 queries	1,800 queries

TABLE 5.5: Experimental Scenarios for Experiment III (Granularity and Time Complexity): 1,800 query audios with additive noise in a range from -5dB to 20dB are processed by proposed system as well as Shazam and Panako.

stage a sampling rate (sr) of 7KHz is chosen for a reason discussed in Section 4.1.1. Audios re-sampled at 7KHz are transformed into their corresponding time-frequency representation (spectrogram) by applying STFT with $NFFT = 1024$ and $hop\_length = 32$. Values of $NFFT$ and hop_length are chosen to get optimal spectral information from time series audio data. Once the spectrogram is computed a peak picking algorithm is applied using 150x75 maximum filter (spans 150 frames along tempo axis and 75 bins along pitch axis to extract one time-frequency point with maximum energy) and 3x3 minimum filter (spans 3 frames along tempo axis and 3 bins along pitch axis to extract one time-frequency point with minimum energy) to extract temporally localized spectral peaks. Here, reducing the dimension of maximum filter increases total number of spectral peaks which in-turn affects overall storage and computation requirement of the system. Whereas, increasing the dimension of the maximum filter will reduce total number of spectral peaks and this will have a negative effect on the discriminative power of audio fingerprints to be formed.

The next step in parameter selection is to select the size of the target zone. Based on Figure 4.8, the width of target zone ($t\_width\_referece$) assigned to the root peak ($P1$) while extracting reference audio fingerprints is set to one second duration along tempo axis and its height covers the whole spectral region along pitch axis. Using $n\_frames = \left\lceil \frac{sr*duration}{hop\_length} \right\rceil$, $t\_width\_referece$ is computed to 219 frames. The root peak ($P1$) is expected to be 0.2 seconds (44 frames) away from the assigned target zone and this results in the minimum frame number to be included in the target zone (min_frame) to be 44 frames away from the frame number of the root peak (min_frame=$P1\_x + 44$) and the maximum frame number to be included in the target zone (max_frame) to be $min\_frame + 219$ frames. However, while extracting fingerprints from query audio, the width of the target zone (t_width_query) is set to larger value (318 frames) based on $t\_width\_query = \frac{t\_width\_reference}{1-t\_tolerance}$, where $t\_tolerance$ is set to 0.31 (handle up-to 30%

modification along tempo axis). Similar to reference audio fingerprint extraction, the assigned target zone is set to be 0.2 seconds (44 frames) away from the root peak while extracting query fingerprints.

While matching audio fingerprints extracted from query audios, parameters for the first filter defined by Equation 4.3 are initialized as follow. Using Equation 4.6, minimum allowed modification along tempo axis (*min_t_mod*) is computed to 0.76 and the maximum allowed modification along tempo axis (*max_t_mod*) is computed to 1.449. While computing the maximum and minimum allowed modifications along pitch axis, *f_tolerance* is set to 0.31 to handle 30% modification along pitch axis. Using Equation 4.7, *min_f_mod* is computed to 0.76 and *max_f_mod* is computed to 1.449. Finally, in the second filter defined by Equation 4.8, the maximum acceptable positional mismatch is set to 2.

5.5 Evaluation Metrics

Different statistical tools are used to analyze results collected from conducted experiments. To evaluate robustness of the system both Accuracy defined in Equation 5.1 and Precision defined in Equation 5.2 are used as evaluation metrics. The accuracy is interpreted as Recognition Rate where as the precision tells about the ability of the system in avoiding false positives.

$$Accuracy = \frac{TP}{TP + FP + FN} \tag{5.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{5.2}$$

True Positive (TP) is a correct Audio ID that is returned by by the system for a given query audio where its reference audio fingerprints are stored in the reference fingerprint database. **False Positive (FP)** is an incorrect Audio ID that is returned by the system for a given query audio where its reference audio fingerprints are stored in the reference fingerprint database. **False Negative (FN)** is where the system fails to return an Audio ID for a given query audio whose reference audio fingerprints are stored in the reference fingerprint database.

To evaluate the reliability of the system True Negative Rate (TNR), defined in Equation 5.3, is used as an evaluation metrics. Here, the TNR tells about the ability of the

system to specify audios which are not known by the system.

$$TNR = \frac{TN}{TN + FP} \tag{5.3}$$

True Negative (TN) is a correct "no match found" response by the system for a given query audio whose reference audio fingerprints are not stored in the reference fingerprint database. **False Positive (FP)** is when incorrect Audio ID is returned by the system for a given query audio whose reference audio fingerprints are not stored in the reference fingerprint database.

5.6 Results

5.6.1 Experiment I: Robustness

Results collected from conducted experiments to test robustness of the system are presented as follow.

Experiment I-A: Additive Noise

In this experiment, 2,700 query audios with additive noise in a range from -20dB to 20dB are processed. As shown in Figure 5.1, all three systems (Proposed, Shazam and Panako) exhibited an increase in recognition rate as a noise level decrease from -20dB to 20dB. For the ease of analysis it is better to look at the result presented in Figure 5.1 in three parts:

- For a noise level in a range from -20dB to -10dB, Shazam shows an increase in recognition rate from 0.0 to 0.27, Panako shows almost no change in recognition rate (from 0.0 to 0.05) and proposed approach shows an increase in recognition rate from 0.013 to 0.127.
- For a noise level in a range from -10dB to 5dB, Shazam shows an increase in recognition rate from 0.27 to 0.987, Panako shows an increase in recognition rate from 0.05 to 0.907, and proposed approach shows an increase in recognition rate from 0.127 to 0.973.
- As a noise level decreases beyond 5dB, the recognition rate of Shazam and proposed approach becomes 1.0 where as for Panako the maximum recognition rate attained is 0.95 at 20dB noise level.

In general, from results collected in this experiment it is possible to infer the following two key points: i) all three system shows robustness to additive noise in a range from

-5db to 2dB with slight difference in their recognition rate and precision (Shazam an average recognition rate of 0.96 with precision of 99%, Panako an average recognition rate of 0.82 with precision of 97% and proposed approach an average recognition rate of 0.9 with precision of 97%). This supports the assumption that spectral peaks can survive significant audio distortion; and, i) compared to Shazam and Proposed approach recognition rate of Panako is relatively lower and this shows the idea of extracting 24 to 60 spectral peaks and transforming them into 8 to 20 triplets for each second audio might lead to high entropic audio fingerprints which are difficult to reproduce in the presence of noise.

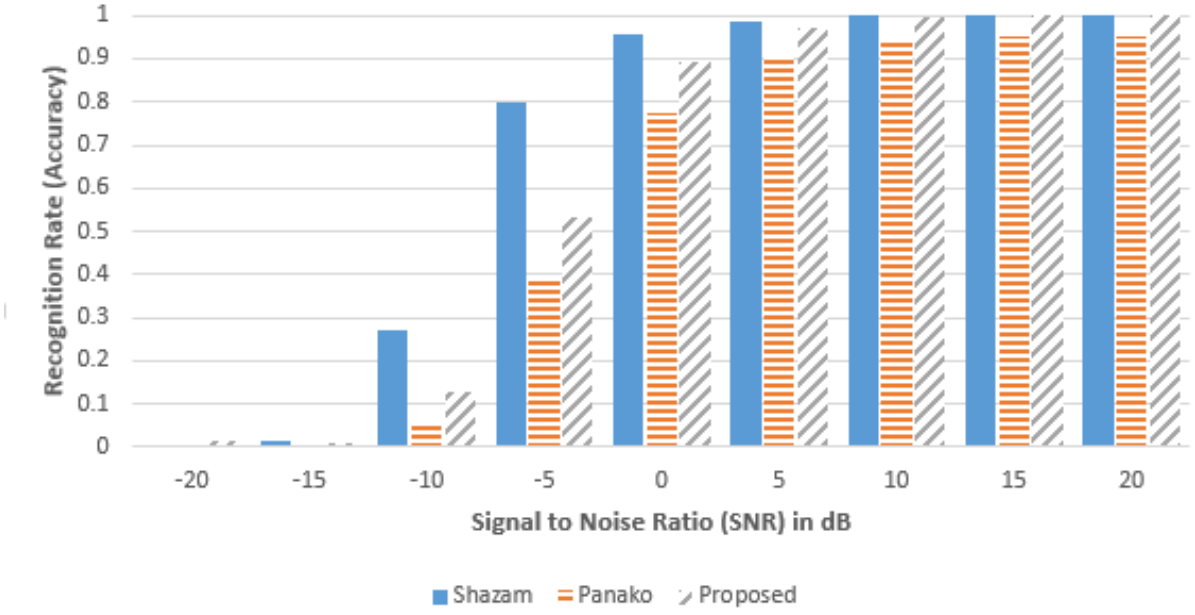


FIGURE 5.1: Results for Experiment I-A: Robustness to additive noise

Experiment I-B: Linear Speed Change

In this experiment, 9,300 query audio with linear speed modification between -30% to 30% are processed. As shown in Figure 5.2, the proposed system is robust to linear speed change in a range of -30% to 22% with an average recognition rate of 0.845 and precision of 97%, which is a significant improvement compared to Panako (robust to speed change between -12% to 6% with an average recognition rate of 0.89 and precision of 99%) and Shazam (failed to handle 2% linear speed change). However, as shown in Figure 5.2, the recognition rate degrade faster for positive speed changes than for negative speed changes. The recognition rate becomes close to 0.0 beyond 22% speed change, whereas it is still above 0.4 until -30% speed change. To better understand this result, we conducted further analysis on effects of linear speed change. The analysis indicates that the loss in recognition rate when speed change is above

22% is due to reduction of spectral peak density. The reduction in spectral peak is attributed to the following two factors: i) positive speed change means playing an audio at a faster speed and this can be done either by reducing the sampling rate or removing samples at a constant rate. In both cases there is a reduction in perceptual information of the audio which in turn affects the overall spectral density in the audio; and ii) positive speed change brought spectral peaks closer to each other along tempo axis, i.e peaks which were far apart originally become closer to each other in a faster audio. However, the size of employed rectangular filter is fixed and this results in selection of only one strong peak and the others will be discarded.

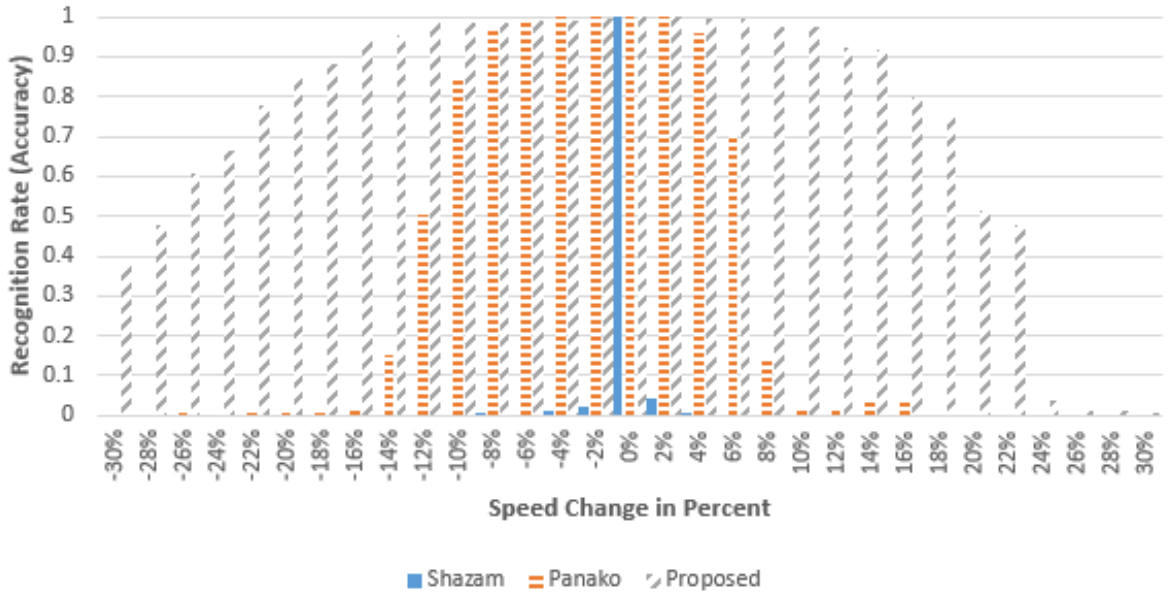


FIGURE 5.2: Results for Experiment I-B: Robustness to linear speed change

Experiment I-C: Time Stretching

In this experiment, 5,100 query audios are processed with time stretching in range from -16% to 16%. Based on the result shown in Figure 5.3, the proposed approach is robust to time stretching in a range from -10% to 8% with an average recognition rate of 0.57 and 89% precision, which is relatively better compared to Panako (robust to time stretching between -4% to 4% with an average recognition rate of 0.3 and 99% precision) and Shazam (robust to time stretching between -4% to 4% with an average recognition rate of 0.58 and 99% precision). However, compared to its robustness to linear speed change, recorded robustness for time stretching is relatively lower and two basic factors are attributed to this: i) ideally time stretching is expected to introduce constant scaling along tempo axis. However, this is impossible in a practical scenario [45]. ii) time stretching is known for introducing audible artifacts (spectral information

which has not been in the original audio) [19], [45], [46]. These two factors imposed negative effect on the recognition rate of the system.

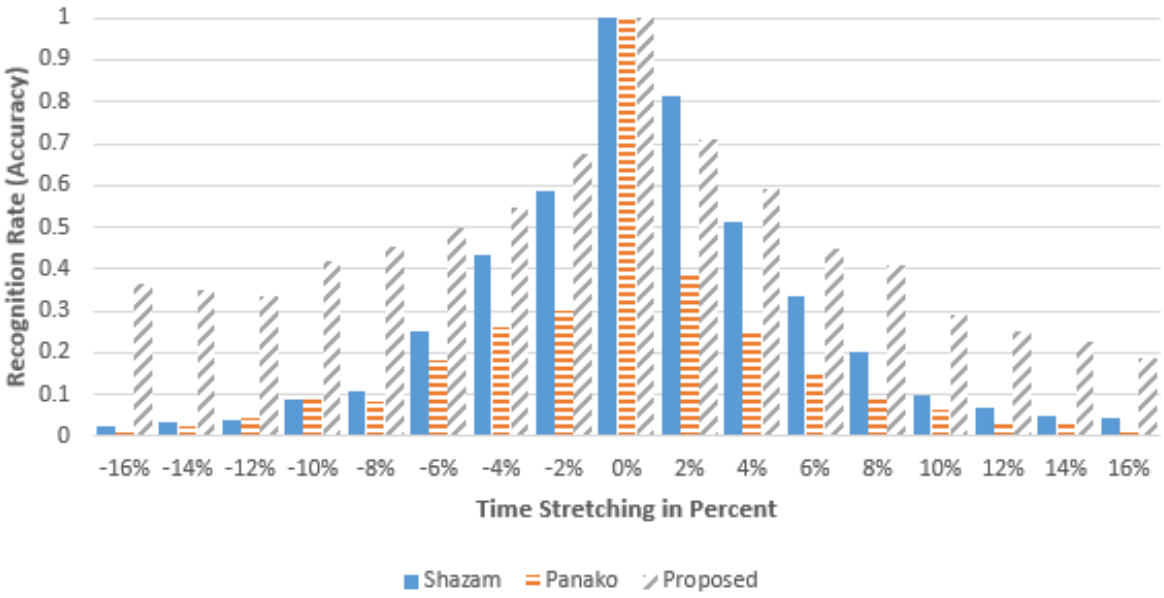


FIGURE 5.3: Results for Experiment I-C: Robustness to time stretching

Experiment I-D: Pitch Shifting

This experiment aimed at evaluating robustness of the proposed approach to pitch shifted query audios. As shown in Figure 5.4, the proposed approach shows a comparable robustness to pitch shifted query audios (average recognition rate of 0.46 with 85% precision is recorded for a pitch shifting in a range from -4% to 4%) with Panako (average recognition rate of 0.37 with 99 % precision is recorded for a pitch shifting in a range from -4% to 4%) where Shazam failed to handle 2% pitch shifting. However, the recognition rate of the proposed system has degraded after 4% pitch shifting. Similar to time stretching, lack of constant scaling as well as introduction of audible artifacts imposed notable challenges. In addition to these two factors, another factor which affects the robustness of the system to pitch shifted query audios is frequency resolution effect of STFT (Short Time Fourier Transform). This can be explained using Figure 5.5 , which shows spectral peaks extracted form an audio with 4% pitch shift (blue dots) along with spectral peaks extracted form original audio (red dots). As shown in Figure 5.5, when a uniform pitch shifting of 4% is applied on the whole spectrum, the positional shift (along pitch axis) exhibited by spectral peaks with higher frequency is not similar to positional shift exhibited by spectral peaks with lower frequency. This irregular positional shift affected shift invariant characteristics of geometric hashing,

which in-turn imposed negative effect on the recognition rate of the system for larger pitch shifts.

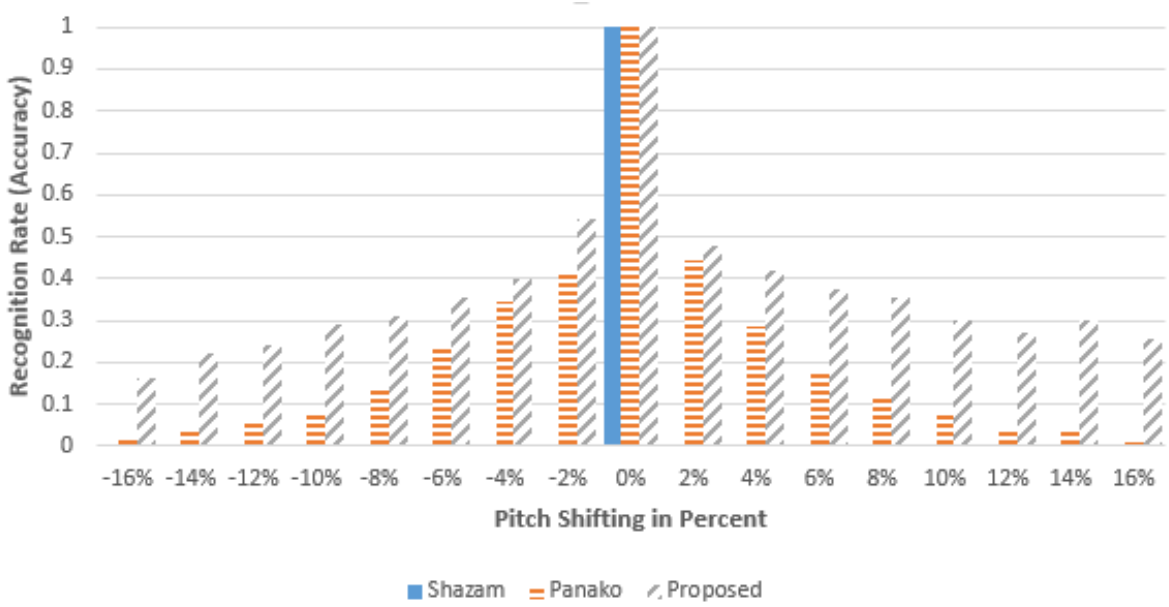


FIGURE 5.4: Results for Experiment I-D: Robustness to pitch shifting

5.6.2 Experiment II: Reliability

To evaluate the reliability of the proposed system, 7,400 query audios extracted from 50 reference audios are processed. The results showed that the proposed approach gives 89% true negatives for query audios extracted from unknown reference audios. However, compared to Shazam (96% detected as true negatives) and Panako (95% detected as true negatives), true negative rate of the proposed system is relatively lower. As its stated in Section 2.3, the reliability of the system depends on the employed matching technique, specifically on the verification part. Ideally, true negative rate of audio fingerprinting system is expected to be 100%, therefore there is still a room for improvement in the employed matching system.

5.6.3 Experiment III: Granularity and Time Complexity

In this experiment granularity (effect of query audio length on recognition rate) and time complexity of proposed system are evaluated in comparison with Shazam and Panako. As shown in Figure 5.6, the proposed approach as well as Shazam and Panako exhibited an increase in recognition rate as the length of query audio increases from 5 seconds to 30 seconds. The recognition rate of Shazam increased from 0.53 for 5 seconds query audio length to 0.96 for 30 seconds query audio length, the recognition

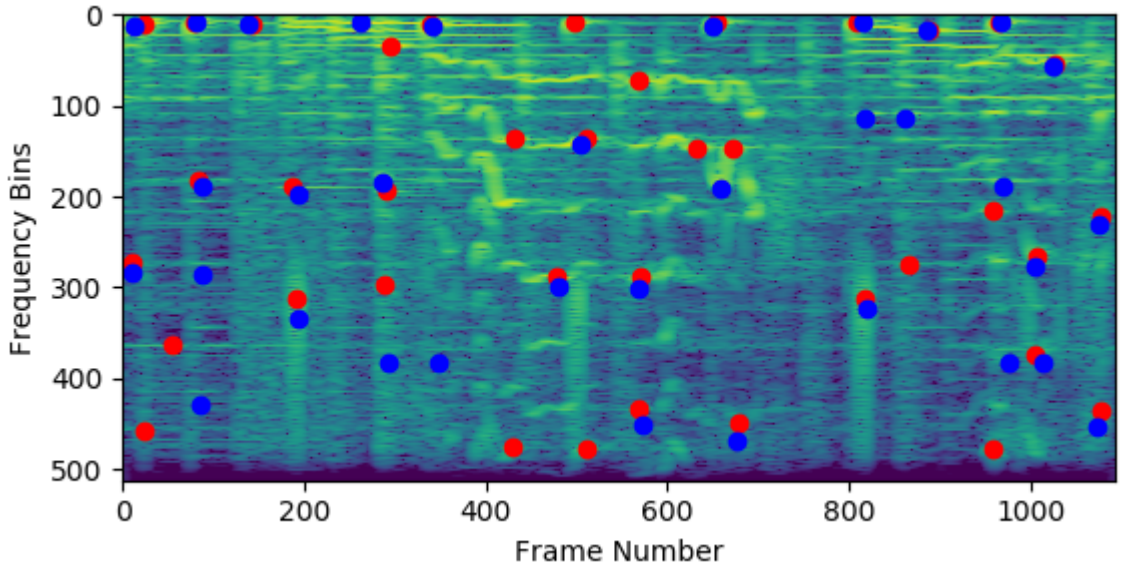


FIGURE 5.5: Effect of 4% Pitch Shifting(red dots represent spectral peaks extracted form original audio and blue dots represent spectral peaks extracted from modified audio with 4% pitch shifting)

rate of Panako increased from 0.2 for 5 seconds query audio length to 0.82 for 30 seconds query audio length and the recognition rate of proposed approached increased from 0.4 for 5 seconds query audio length to 0.9 for 30 seconds query audio length. However, this increase in recognition rate comes along with an increase in average response time. As shown in Figure 5.7, the response time of all three systems (Shazam, Panako and Propsosed) increases as the length of query audio increases from 5 seconds to 30 seconds. Compared to Shazam (an average response time of 0.38 seconds for a query audio length in a range from 5 seconds to 30 seconds) and Panako (an average response time of 1.55 seconds for a query audio length in a range from 5 seconds to 30 seconds) the average response time of proposed approach (an average response time of 2.19 seconds for a query audio length in a range from 5 seconds to 30 seconds) is relatively higher. The increase in average response time of proposed approach compared to Shazam and Panako comes from an increase in computation due to the employed fingerprinting scheme. That is, when the query audio length increases, it will increase the perceptually relevant audio information which in-turn increases the number of spectral peaks. As the number of spectral peak increases, the amount of time required to generate audio fingerprints (group triplets, select valid triplets which can satisfy the condition stated in Equation 4.1 and generate geometric hashes) as well as the time required to match extracted fingerprints (match generated query audio fingerprints with reference audio fingerprints, filter incorrect matches and verify matches) increase and all these computations will have their own contribution on the increase in the average

response time of the system.

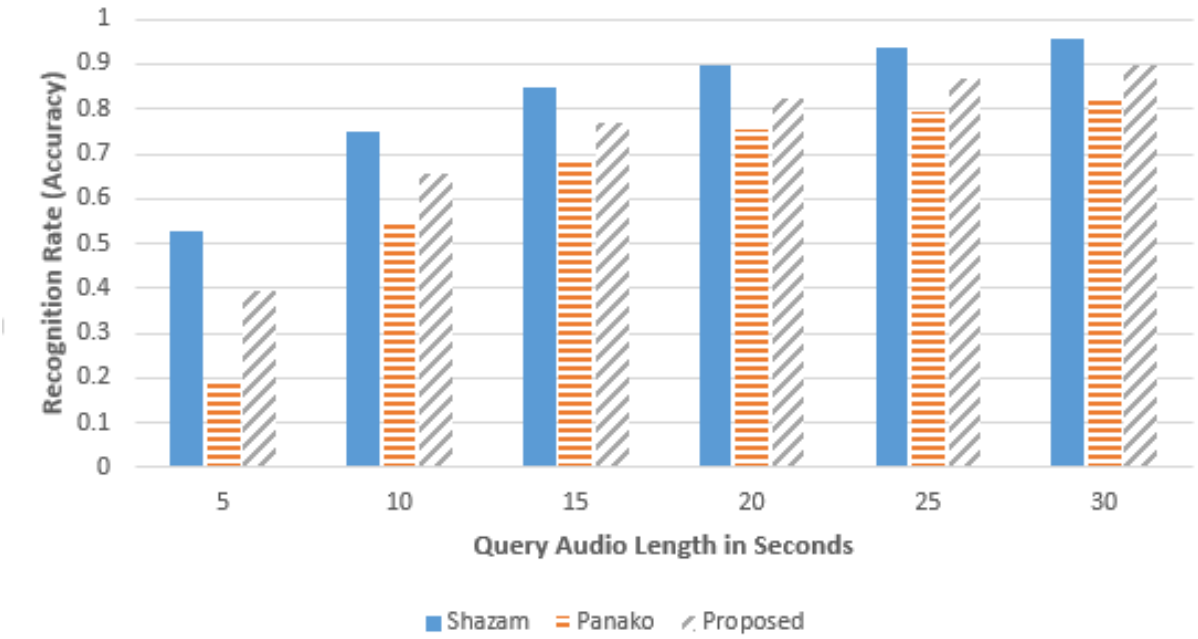


FIGURE 5.6: Results for Experiment III: Granularity

5.6.4 Summary of Results

Collected results from all the experiments are summarized as follow:

- The proposed approach is robust to linear speed change in range -30% to 22% with an average recognition rate of 0.845 and precision of 97%. In addition to robustness to linear speed change, the proposed approach also showed robustness to additive noise in a range from -5dB to 20dB with an average recognition rate of 0.9 and 97% precision, robustness to time stretching between -10% and 8% with average recognition rate of 0.57 and 89% precision, and robustness to pitch shifting between -4% to 4% with an average recognition rate of 0.46 and 85% of precision.

- Reliability of the proposed system is evaluated with 7,400 unknown query audios and of these query audios 89% were classified as correct True Negatives.

- The effect of query audio length on recognition rate (granularity) of proposed approach is also assessed and the proposed approach shows an increase in recognition rate from 0.4 for 5 seconds query audio length to 0.9 for 30 seconds query audio length. However, the increase in recognition rate comes along with an increase in average response time of the proposed system.

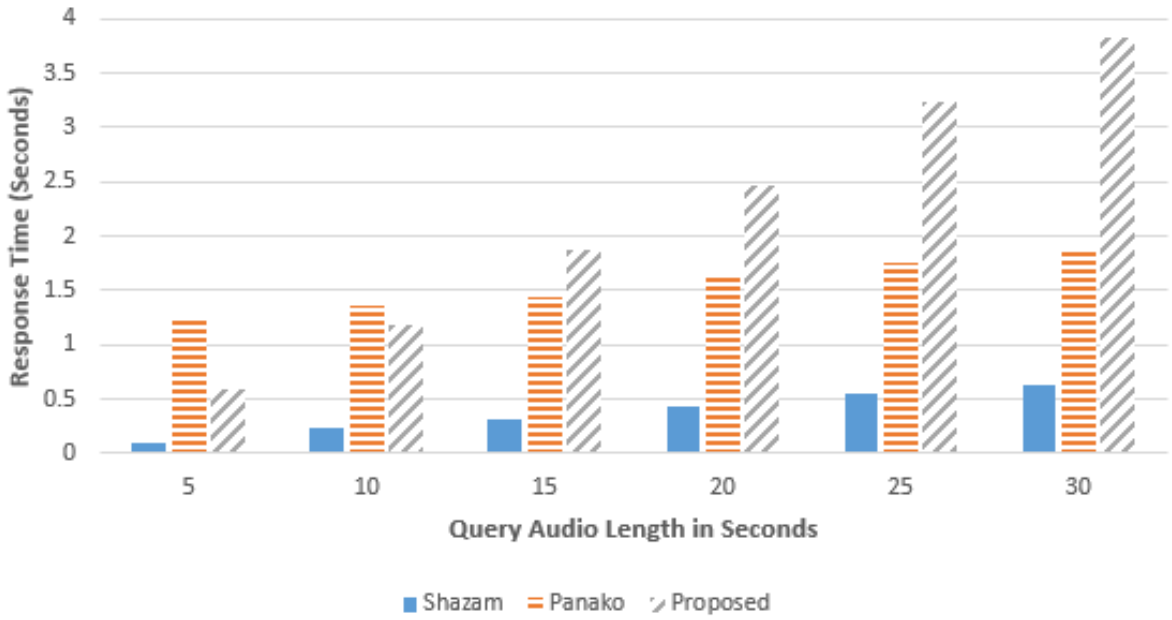


FIGURE 5.7: Results for Experiment III: Granularity

In general, based on the collected results, ***it is possible to infer that triple point geometric hashing is a possible option to develop audio fingerprinting system which is robust to different distortions including linear speed change.***

5.7 Threats to Validity

5.7.1 Threats to Internal Validity

Measurement associated threats, specifically threats to instrumentation might affect the validity of granularity and time complexity experiment. Those threats might arise from the type of machine used for running the experiments and the choice of development tools. In this study, however, all approaches are evaluated on the same machine. Hence, the threats will not affect the results and comparisons made in this study.

5.7.2 Threats to External Validity

Threats to external validity might arise from:

- Choice of parameters (filter size, sampling rate, target zone size, hop length and others): this threat might affect experiments conducted with proposed approach as well as reproduced baseline works. In this study, all approaches are subjected to similar parameter setting (only applicable for common parameters such as

sampling rate, hop length and others depending on each approach implementation detail). Hence, threats related to parameter choice will not affect comparisons.

- Selection in Audio Corpus: variety of audio formats used for experimentation might also impose another threat to external validity. In this study, however, all approaches are evaluated with audio corpus composed of two types of file formats: mp3 and wav .

Chapter 6

Conclusion and Future Works

6.1 Conclusion

This thesis analyzed the effect of linear speed change in shazam based audio fingerprinting systems. Speed change introduces translation and scaling between spectral peaks along both tempo and pitch axis. To handle these effects, a fingerprint extraction technique, which is invariant to scaling and translation is proposed using triple point geometric hashing. The proposed approach is evaluated with different experiments to assess its robustness, reliability, granularity and time-complexity.

Based on the set of experiments conducted to test robustness of the system, the proposed system:

- Showed robustness to additive noise in a range from -5dB to 20dB with an average recognition rate of 0.9 and 97% precision.
- Showed a robustness to linear speed change between -30% to 22% with an average recognition rate of 0.845 and precision of 97%. This is significant improvement compared to Panako (handle speed change between -12% to 6%) and Shazam (failed to handle 2% speed change); and,
- Achieved relatively better robustness on time stretched (handle -10% to 8% with an average recognition rate of 0.57 and 89% precision) and pitch shifted (handle -4% to 4% pitch shifting with an average recognition rate of 0.46 and precision of 85%) query audios compared to Panako and Shazam.

Reliability of the proposed approach is also evaluated with 7,400 query audios (query audios extracted from reference audios which have not been fingerprinted) and the proposed approach were able to specify 89% of query audios as true negatives. The effect of query audio length on recognition rate (granularity) of proposed approach is also assessed and the proposed approach shows an increase in recognition rate from 0.4 for 5 seconds query audio length to 0.9 for 30 seconds query audio length. However,

the increase in recognition rate comes along with an increase in average response time of the proposed system.

6.2 Future Works and Recommendations

Finally, the following gaps are identified as prospective future research directions:

- Enhancing robustness of the system to pitch shifted and time stretched query audios. It would be worthwhile to conduct further experiments with pitch shifted and time stretched query audios to devise a way which can enhance robustness to these modifications.
- In its current form, the developed system can be applied in different application scenarios, such as advertisement tracking, royalty tracking, indoor localization and other application scenarios which doesn't require dealing with large scale audio collections. However, to apply developed system in application scenarios which deals with large scale audio collections, such as content based audio identification and audio file organization, it is better to first evaluate scalability as well as versatility of the system.

References

[1] J Downie, K. West, A. Ehmann, and E. Vincent, "The 2005 music information retrieval evaluation exchange (mirex 2005): Preliminary overview", 2005.

[2] J. Haitsma and T. Kalker, "Speed-change resistant audio fingerprinting using auto-correlation", in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, IEEE, vol. 4, 2003, pp. IV–728.

[3] C. Ouali, P. Dumouchel, and V. Gupta, "A robust audio fingerprinting method for content-based copy detection", in *2014 12th International Workshop on Content-Based Multimedia Indexing (CBMI)*, IEEE, 2014, pp. 1–6.

[4] X. Zhang, B. Zhu, L. Li, W. Li, X. Li, W. Wang, P. Lu, and W. Zhang, "Sift-based local spectrogram image descriptor: A novel feature for robust music identification", *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2015, no. 1, p. 6, 2015.

[5] J. Six and M. Leman, "Panako: A scalable acoustic fingerprinting system handling time-scale and pitch modification", in *15th International Society for Music Information Retrieval Conference (ISMIR-2014)*, 2014.

[6] R. Sonnleitner and G. Widmer, "Robust quad-based audio fingerprinting", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 3, pp. 409–421, 2015.

[7] R. Sonnleitner and G. Widmer, "Quad-based audio fingerprinting robust to time and frequency scaling.", in *DAFx*, Citeseer, 2014, pp. 173–180.

[8] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system with an efficient search strategy", *Journal of New Music Research*, vol. 32, no. 2, pp. 211–221, 2002.

[9] B. Logan, "Mel frequency cepstral coefficients for music modeling.", in *ISMIR*, vol. 270, 2000, pp. 1–11.

[10] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform", *IEEE transactions on Computers*, vol. 100, no. 1, pp. 90–93, 1974.

[11] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[12] J. C. Brown and M. S. Puckette, "An efficient algorithm for the calculation of a constant q transform", *The Journal of the Acoustical Society of America*, vol. 92, no. 5, pp. 2698–2701, 1992.

[13] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis", *IEEE transactions on information theory*, vol. 36, no. 5, pp. 961–1005, 1990.

[14] N. Kehtarnavaz and N. Kim, *Digital signal processing system-level design using LabVIEW*. Elsevier, 2011.

[15] B. Blankertz, "The constant q transform", *URL http://doc. ml. tu-berlin. de/bbci/material/publications/Bla_constQ. pdf*, 2001.

[16] J. H. Jensen, *Feature extraction for music information retrieval*. Multimedia Information and Signal Processing, Aalborg University, 2010.

[17] A. Wang, "An industrial strength audio search algorithm.", in *Ismir*, Washington, DC, vol. 2003, 2003, pp. 7–13.

[18] P. Cano, E. Batlle, T. Kalker, and J. Haitsma, "A review of audio fingerprinting", *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 41, no. 3, pp. 271–284, 2005.

[19] S. M. Bernsee, "Time stretching and pitch shifting of audio signals", *The DSP Dimension*, 2003.

[20] J. L. Flanagan and R. Golden, "Phase vocoder", *Bell System Technical Journal*, vol. 45, no. 9, pp. 1493–1509, 1966.

[21] R. V. Cox, R. E. Crochiere, and J. D. Johnston, "Real-time implementation of time domain harmonic scaling of speech for rate modification and coding", *IEEE Journal of Solid-State Circuits*, vol. 18, no. 1, pp. 10–24, 1983.

[22] D. Jang, M. Jin, J. S. Lee, S. Lee, S. Lee, J. S. Seo, and C. D. Yoo, "Automatic commercial monitoring for tv broadcasting using audio fingerprinting", in *Audio Engineering Society Conference: 29th International Conference: Audio for Mobile and Handheld Devices*, Audio Engineering Society, 2006.

[23] M. Pora and Z. Skolicki, *Determining tv program information based on analysis of audio fingerprints*, US Patent 8,843,952, 2014.

[24] R. D. Major, *Pre-distribution identification of broadcast television content using audio fingerprints*, US Patent App. 13/836,688, 2014.

[25] S. Bilobrov and I. Poutivski, *Commercial detection based on audio fingerprinting*, US Patent 9,258,604, 2016.

[26] L. d. C. Gomes, P. Cano, E. Gómez, M. Bonnet, and E. Batlle, "Audio watermarking and fingerprinting: For which applications?", *Journal of New Music Research*, vol. 32, no. 1, pp. 65–81, 2003.

[27] T.-K. Hon, L. Wang, J. D. Reiss, and A. Cavallaro, "Audio fingerprinting for multi-device self-localization", *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 10, pp. 1623–1636, 2015.

[28] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik, "Indoor localization without infrastructure using the acoustic background spectrum", in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, ACM, 2011, pp. 155–168.

[29] F. Liu, "Audio fingerprinting for speech reconstruction and recognition in noisy environments", 2017.

[30] S. Seyoum, L. Alfonso, S. J. Van Andel, W. Koole, A. Groenewegen, and N. Van De Giesen, "A shazam-like household water leakage detection method", *Procedia Engineering*, vol. 186, pp. 452–459, 2017.

[31] Y. Ke, D. Hoiem, and R. Sukthankar, "Computer vision for music identification", in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE, vol. 1, 2005, pp. 597–604.

[32] J. S. Seo, J. Haitsma, and T. Kalker, "Linear speed-change resilient audio fingerprinting", in *Proc. IEEE Workshop on Model based Processing and Coding of Audio*, 2002.

[33] S. Yao, B. Niu, and J. Liu, "Enhancing sampling and counting method for audio retrieval with time-stretch resistance", in *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*, IEEE, 2018, pp. 1–5.

[34] S. Yao, B. Niu, and J. Liu, "Audio identification by sampling sub-fingerprints and counting matches", *IEEE Transactions on Multimedia*, vol. 19, no. 9, pp. 1984–1995, 2017.

[35] S. Baluja and M. Covell, "Audio fingerprinting: Combining computer vision & data stream processing", in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, IEEE, vol. 2, 2007, pp. II–213.

[36] D. Williams, A. Pooransingh, and J. Saitoo, "Efficient music identification using orb descriptors of the spectrogram image", *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2017, no. 1, p. 17, 2017.

[37] S. Fenet, G. Richard, and Y. Grenier, "A scalable audio fingerprint method with robustness to pitch-shifting.", in *ISMIR*, 2011, pp. 121–126.

[38] A. Arzt, S. Böck, and G. Widmer, "Fast identification of piece and score position via symbolic fingerprinting.", in *ISMIR*, 2012, pp. 433–438.

[39] D. Lang, D. W. Hogg, K. Mierle, M. Blanton, and S. Roweis, "Astrometry. net: Blind astrometric calibration of arbitrary astronomical images", *The astronomical journal*, vol. 139, no. 5, p. 1782, 2010.

[40] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and music signal analysis in python", in *Proceedings of the 14th python in science conference*, vol. 8, 2015.

[41] F. C. Tsai, "Geometric hashing with line features", *Pattern Recognition*, vol. 27, no. 3, pp. 377–389, 1994.

[42] H. J. Wolfson and I. Rigoutsos, "Geometric hashing: An overview", *IEEE computational science and engineering*, vol. 4, no. 4, pp. 10–21, 1997.

[43] A. Guttman, *R-trees: A dynamic index structure for spatial searching*, 2. ACM, 1984, vol. 14.

[44] J. Laroche and M. Dolson, "Phase-vocoder: About this phasiness business", in *Proceedings of 1997 Workshop on Applications of Signal Processing to Audio and Acoustics*, IEEE, 1997, 4–pp.

[45] J. Driedger and M. Müller, "A review of time-scale modification of music signals", *Applied Sciences*, vol. 6, no. 2, p. 57, 2016.

[46] S. Kraft, M. Holters, A. von dem Knesebeck, and U. Zölzer, "Improved pvsola time-stretching and pitch-shifting for polyphonic audio", in *Proceedings of the International Conference on Digital Audio Effects (DAFx), York, UK*, 2012, pp. 17–21.