|  |
| --- |
|  |
| Error Report on Enhanced Audio Recognition Project |
|  |

|  |
| --- |
| Joshuan Jimenez  July 21, 2019 |

# Table of Contents

[Abstract 2](#_Toc14644704)

[Introduction 3](#_Toc14644705)

[Data Section 4](#_Toc14644706)

[Key Factors of Audio 4](#_Toc14644707)

[Audio Fingerprinting 4](#_Toc14644708)

[Microphones and Interpreting Sound 5](#_Toc14644709)

[Cleaning Sound 5](#_Toc14644710)

[Possible Solutions 7](#_Toc14644711)

[Automatic Cleaning 7](#_Toc14644712)

[Adjusting Sensitivity of Microphone 7](#_Toc14644713)

[Combining Both Solutions 7](#_Toc14644714)

[Conclusion 8](#_Toc14644715)

[Summary of Findings 8](#_Toc14644716)

[Interpretation of Findings 8](#_Toc14644717)

[Recommendations 8](#_Toc14644718)

[References 9](#_Toc14644719)

# Table of Figures

[Figure 1 – Different Audio Fingerprints 4](#_Toc14643144)

[Figure 2 - Diaphragm of Microphone 5](#_Toc14643145)

[Figure 3 - Clean Audio Fingerprint 6](#_Toc14643146)

[Figure 4 - Dirty Audio Fingerprint 6](#_Toc14643147)

# Abstract

Retrieving audio input from the user for use in the application has proven to be a problem, as the audio tends to have background noise. This background noise makes it difficult for the program to be able to compare the audio fingerprints of the input to the ones in the database.

Microphones pick up all sound, even unwanted background noise, which makes the audio fingerprint dirty. This is because microphones on phones tend to be a little sensitive, picking up noise like the wind or outside ambience. There is also the possibility that the microphone quality is not that great, which affects the quality of the audio. These situations cause a fingerprint to be dirty.

Normally, when audio is picked up from a microphone, a person will have to process the audio to clean up the file to make it sound clear for use. This will make the audio fingerprint a lot cleaner and easier to read, while improving the quality of the audio. The main issue is trying to replicate this same idea without having a person do it.

To combat this, the application will have to be able to process the audio input on its own consistently. This will mean that the application will have to learn how to minimize background noise to be able to have an easier time comparing the input with the files on the database.

This idea will be applied to the application through any of the following recommendations:

* Develop an algorithm dedicated to cleaning the audio file as a producer would
* Adjust sensitivity of microphone to diminish background noise being captured
* Combine the two options to minimize background noise

# Introduction

There is an issue concerning how the application compares sound files in the database to the one provided by the user. If the quality of the audio input seems to be lacking, then it will mean that the program will have a hard time analyzing and comparing audio fingerprints. When analyzing these audio files from any source, they can be recognized by their audio fingerprint. This is seen as the fingerprint of a human, as it is specific to one object. This would allow the application being worked on to be able to recognize files like the audio input retrieved from the user.

An audio fingerprint is a compact representation of some audio (Chirp Team, 2018). It is like the fingerprints of a human, as it is specific to a certain person, or instance of sound for audio. With this, we can compare audio files to find similar ones. That is however, if the audio input provided does not have any background noise. When audio is picked up from a microphone, it picks up not only the main focus, but any sounds in the area, such as the wind, ventilation, or even breathing. This is all based on the sensitivity of the mic and the environment.

As stated on an article written by Dan Gravell, audio fingerprinting:

works by combining two basic elements: a database of unique audio fingerprints of millions of audio files, and software that can process and analyze entries while searching that database for a match (Gravell, 2012).

Given the possibility that not every input will be optimal, me and my partner, Randy Rivas, propose to implement an algorithm to assist with being able to clean up the audio input as much as possible. This algorithm can be based on any of the following below:

* An algorithm dedicated to cleaning the audio file as a producer would
* An algorithm to adjust sensitivity of microphone to diminish background noise being captured
* Combining the two options together to minimize background noise

By implementing one of these solutions, we will be able to improve the quality of the audio files. This will then allow a good audio fingerprint to be created and used for comparison.

To show how the problem occurs and how to remedy it, this report documents the research done by me and my partner. It goes over what an audio fingerprint is, how a microphone picks up sound, cleaning an audio fingerprint, and the potential solutions for the error described.

# Data Section

## Key Factors of Audio

Audio has many characteristics to it that allow it to be identified. Therefore, it is crucial to understand its characteristics, how it is captured, and how to maintain the quality.

### Audio Fingerprinting

When it comes to the audio fingerprint, it is vital to ensure the file quality is at its best to easily compare the audio input to the files in the database.

As previously stated, audio fingerprinting is the compacted representation of some audio. The figure below provides an example of what an audio fingerprint looks like:

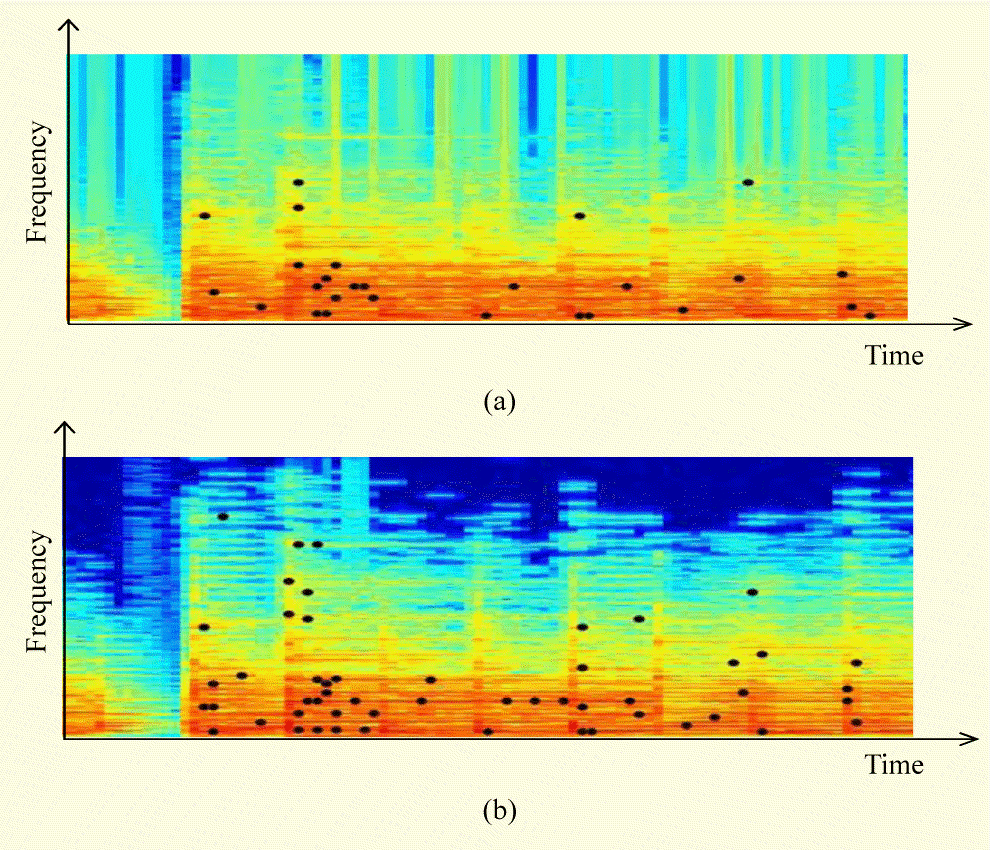


Figure – Different Audio Fingerprints

The audio fingerprint provides a lot of information, such as certain frequencies that stand out at certain intervals of the audio file. This is represented by the points in the audio fingerprints, which gives each audio clip its identity. A clapping audio file will have a certain fingerprint that distinguishes it from sounds like laughing or screaming.

By analyzing this key information exhibited by the fingerprint, me and my partner can be able to distinguish the audio input from the songs that appear in the database. This audio fingerprint is vital in comparing files and finding which song is like the audio input obtained from the user.

### Microphones and Interpreting Sound

Microphones being of low quality or being damaged can lower the quality of the file, which in turn creates a poor-quality audio fingerprint.

To be able to retrieve the audio input, a microphone is needed to pick up the sound. The key factor with this project is understanding how a microphone works. A microphone functions by converting acoustical energy into electrical energy (How do Microphones Work?, n.d.). When acoustical energy hits a microphone, the diaphragm, which is a thin piece of material, vibrates and causes other components in the microphone to vibrate. These vibrations are then turned into the audio signal, which is needed to create an audio clip. The diaphragm in a regular microphone can be seen in Figure 2:

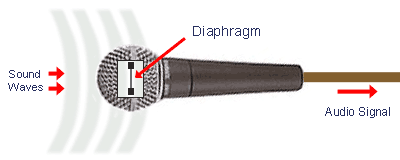


Figure - Diaphragm of Microphone

The diaphragm is one of the most important components of the microphone, as it serves as the transformer for energy. This part can be damaged, causing audio files to sound low quality due to the diaphragm not being able to convert the energy as well. If this is the case, it will mean that the audio input can be difficult to analyze due to the quality of the microphone. This tends to be the reason why the user’s voice can sound low if using an old phone with a worn-out microphone.

### Cleaning Sound

Processing the audio file, or cleaning up the audio file, is vital as it ensures the quality of the file is at its best. When taking in audio input, whether the microphone is in proper working order or not, an audio fingerprint can be generated from the input. The issue is the quality of the audio input can differ as background noise or static is noticeable. This in turn makes it hard to listen to a file or even analyze its fingerprint.

Since this can be an issue, normally a producer will listen to the audio and try to clean it using programs like FL Studio, Audacity, or even Pro Tools. The idea is to remove the “unwanted frequencies,” especially in the lowest and highest frequencies, as it is the most offensive at those extremes (How to Clean Audio Files, n.d.).

Figure 3 demonstrates a clean audio fingerprint, one that does not have any background noise or static:

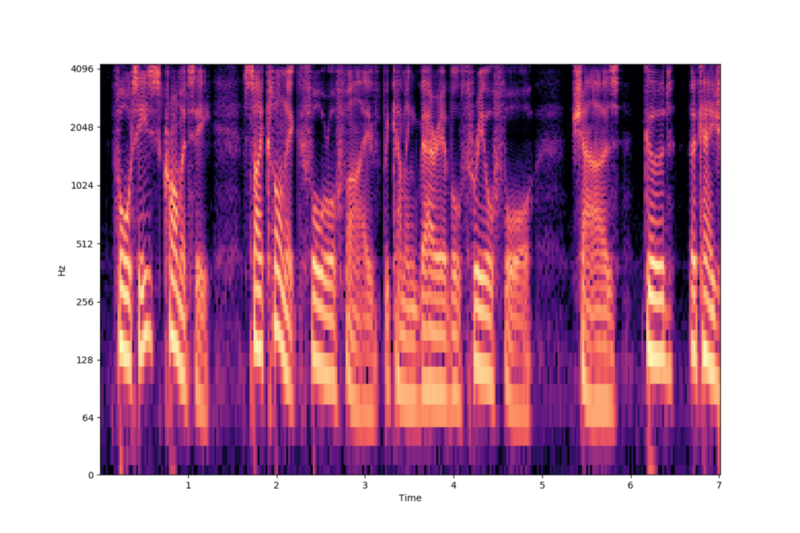


Figure - Clean Audio Fingerprint

This means that the audio file is void of any static or unwanted noise, allowing it to be able to be scanned and analyzed with ease. The fingerprint will be drawn much easier and can then be compared to the files in the database.

However, the issue comes when the microphone captures the wanted sound, along with some static. This static in turn causes the fingerprint to be dirty, along with the file having unwanted background noise upon playing it back. Figure 4 demonstrates this dirty audio fingerprint:

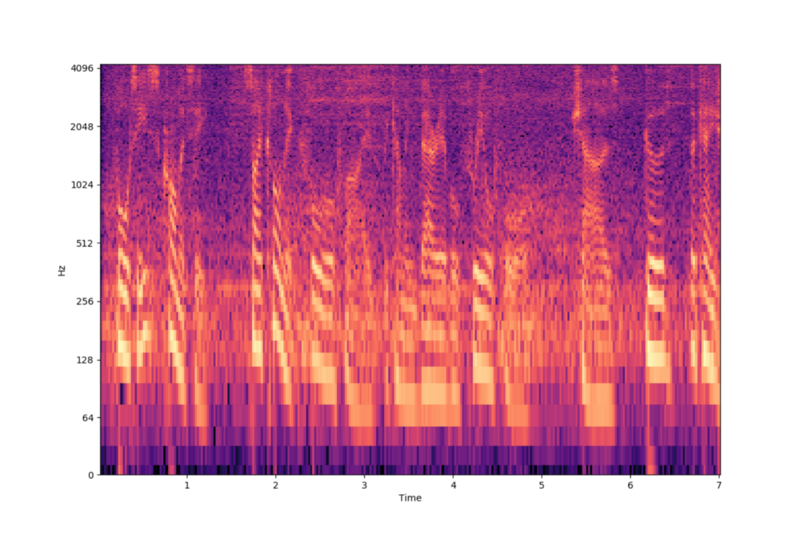


Figure - Dirty Audio Fingerprint

With a file like this, it will be nearly impossible to compare it and find the actual song desired.

## Possible Solutions

With an issue concerning the quality of the audio file, it is vital to have the application adjust the file as needed to improve the quality of the input’s fingerprint.

### Automatic Cleaning

A possible way of improving the audio fingerprint of the input is to develop an algorithm that will automatically clean the file as a producer would.

By having the file be cleansed, the fingerprint will be able to be analyzed much better and with a higher success rate. This will be done by having the algorithm search for abnormal frequencies depicted in the fingerprint. Upon analyzing these abnormalities, it will then edit them to diminish or eliminate the background noise. In return, a clean fingerprint is seen and can be analyzed.

With the help of my partner, we can determine what the algorithm will consider background noise in a file and what it will do to handle it. As previously mentioned, abnormal frequencies that are depicted by the fingerprint will be handled. The can refer to constant low or high frequencies happening over a long interval of time.

We will be determining what the algorithm will shift, in terms of basic settings such as bass, treble, to advanced settings, such as gain, wet/dry, low, and high. These are settings used to modify the sound in a file.

### Adjusting Sensitivity of Microphone

Another recommendation is adjusting the sensitivity of the microphone, to minimize background noise or possible feedback from the microphone itself. If it is too sensitive, the microphone will pick up static or background noise, so the application should lower it. If the microphone is not responsive, then it will increase the sensitivity until a reasonable level is found.

For this to function, an algorithm would have to be developed where it analyzes the fingerprint to see if there is unwanted noise. Upon sensing this, the algorithm would then determine, based on the frequencies being picked up, whether to increase or lower the sensitivity.

By lowering the chances of picking up unwanted noise, the quality of the input can be improved.

### Combining Both Solutions

If needed, both solutions can be combined to ensure the highest quality audio file, as both can complement each other in different aspects.

The second algorithm suggested can be implemented to adjust the microphone as needed. This then allows the first algorithm suggested to be done with much more ease as not much is truly needed to be cleaned. This will ensure the highest quality input is received, allowing the application to compare files easier.

In the end, it will require analysis and thinking to come up with algorithms that will maintain the integrity of the audio input to produce a high-quality audio fingerprint.

# Conclusion

## Summary of Findings

It is important overall to understand the characteristics of audio and understanding how it is changed, read, and received.

As stated, the audio is transformed from acoustical to electrical due to the diaphragm located in a microphone. This in turn produces an audio file that can be played back. Along with this, an audio fingerprint, which is a compact representation of an audio file, can be generated for analysis and comparison. Therefore, it is important to clean the audio file if needed to improve the quality for use, whether it be due to bad recording equipment or background noise.

### Interpretation of Findings

Based on all this research, a dirty audio fingerprint can be the result of either faulty equipment or environmental sounds.

Due to this error, it is necessary to have the application be able to process the audio input on its own to allow the application to compare files accurately between the database and the input. To fix this issue, one of the following solutions will be implemented:

* Come up with an algorithm to have the application clean the audio file
* Come up with an algorithm to adjust the sensitivity of the microphone
* Combine both solutions

By allowing the application to improve the quality of the input provided by the user, the comparison will be done more efficiently and more accurately.

### Recommendations

Concerning the error described, we will ensure to work on the bug and fix it within a two-week time frame, while still following our schedule. Based on what was discussed, we would suggest allowing us to look further into the issue while experimenting with the proposed solutions.

# References

Chirp Team. (2018, March 15). *Audio Fingerprinting*. Retrieved from The Date Over Sound Blog: https://blog.chirp.io/audio-fingerprinting-what-is-it-and-why-is-it-useful/

Gravell, D. (2012, August 21). *What is audio fingerprinting?* Retrieved from Bliss HQ: https://www.blisshq.com/music-library-management-blog/2012/08/21/what-is-audio-fingerprinting/

*How do Microphones Work?* (n.d.). Retrieved from Media College: https://www.mediacollege.com/audio/microphones/how-microphones-work.html

*How to Clean Audio Files*. (n.d.). Retrieved from It Still Works: https://itstillworks.com/clean-audio-files-5916566.html