NETEASE CLOUD MUSIC AUDIO FINGERPRINTING SYSTEM

Yuanzhong Zheng
NetEase Cloud Music
zhengyuanzhong@corp.netease.com

Huaping Liu
NetEase Cloud Music
liuhuaping@corp.netease.com

ABSTRACT

NetEase Cloud Music focuses on revolutionize the music industry by exploring frontier music technologies. This document shows an audio fingerprinting algorithm based on Wang[1] and a new matching strategy has been brought to improve the efficient of system. This paper demonstrates the designs behind ZL# submissions for MIREX 2018 Audio Fingerprinting task.

1. INTRODUCTION

Audio fingerprinting technology has been widely employed in various applications, such as music retrieval, copyright protect, etc. Comparing to the fingerprinting identification technology, the task of audio fingerprinting is aimed at pairing the query clip recorded in noisy environment with original audio in a large database. The main challenge in developing a commercial audio fingerprinting system is extracting robust fingerprints and providing real-time matching in a database with tens of millions of songs with lower cost. A commercial audio fingerprinting system has been shown in Part 2.

Audio fingerprinting features are defined as important acoustic characterizations and can be extracted by several classical methods, such as Fourier transform, spectral estimation and Mel-frequency cepstral transform, etc. All the different methods try to record fingerprints with high distinguishability and robustness. In [1], Wang proposed a method to extract fingerprints by detecting and pairing landmarks in spectrogram in which a landmark was defined as a local maximum. It has been certified as efficient and robust by lots of researchers.

Drawing on the experience of search engine technology, it is possible to match query audio signals in a large dataset in real time. An improved inverted index algorithm has been employed in our system. It will be demonstrated in Part 4 with more details.

2. SYSTEM OUTLINE

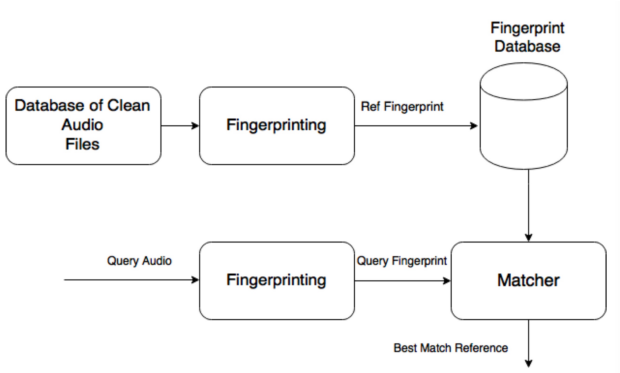

Figure 1. Generalized Audio Fingerprinting System

3. AUDIO FINGERPRINTING EXTRACTION

As stated above, the system extracts audio fingerprint features based on Wang[1]. After receiving audio signals, the system downsamples signals to 8 KHz and converts to mono soundtrack only. Then system divided audio signals into frames with overlap between consecutive frames. Energy peaks will be selected after Fourier transform. Figure 2 highlights energy peaks with black dots.

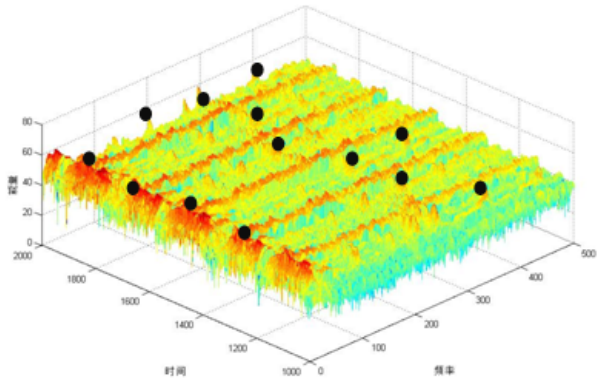

Figure 2. Spectrogram with Highlighting Energy Peaks[1]

Then every landmark will be combined with the other landmark which should be in the target zone. Figure 3 shows how a fingerprint constructs.

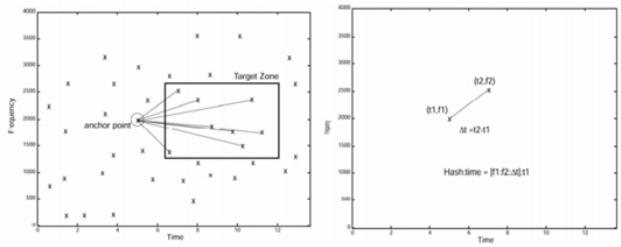

Figure 3. Spectrogram with Highlighting Energy Peaks[1]

Each fingerprint includes three components: anchor's frequency, target's frequency and time difference.

4. MATCHING STRATEGY

Once the query fingerprints are generated, the system tries to search possible positions in songs by improved inverted index algorithm. Then all of possible positions will be go through by comparing similarity between query fingerprints and original fingerprints.

5. COMMERCIAL FEATURES

5.1 Memory Occupation

Due to the fact that more features extracted for audio fingerprints almost always lead to better accuracy, MIREX 2018 sets 50 KB for 1 minute of music at most. However, the system occupies less memory in fact. Less memory occupation reduces cost of commercial system.

5.2 Dataset Maintenance

Staff can manipulate and maintain the large dataset easily. For instance, staff can remove or insert a song without rebuilding the whole dataset.

5.3 High Performance

Although MIREX 2018 provides query audio with about 10 seconds, the system can match query audio fragments with only 2-second-clip. And a high efficient in extracting fingerprints is shown as well. The system generates fingerprints of a single song counted by seconds.

6. FOLDER CONTENTS

The submission folder contains of the following programs and documents:

- **builder**: CPP binary executable file for extracting audio fingerprints and creating dataset.
- **matcher**: CPP binary executable file for extracting and matching query audio fingerprints.
- **ffmpeg**: A binary executable file.
- **Data**: An empty folder to save database.
- **README.txt**: A readme file which contains the details on how to run the scripts, details regarding the necessary command line arguments and details regarding the build system for compiling the binaries.

7. REFERENCES

[1] A.Wang, The Shazam music recognition service, Comm. ACM, vol. 49, no. 8, pp. 44 48, Aug. 2006.