THE 2020 NETEASE AUDIO FINGERPRINT SYSTEM

Peng Li
NetEase Cloud Music
hzlipeng@corp.netease.com

Songsheng Pan
NetEase Cloud Music
hzpansongsheng@corp.netease.com

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

ABSTRACT

This document describes our submission to audio fingerprinting task of MIREX 2020. It is an improved version of our previous submissions in 2017 and 2018. While the approach is again based on robust landmark (i.e., local maxima) detection and fast matching with inverted index, it is expected to achieve better recognition performance due to the fact that more advanced algorithms have been implemented in landmark detection and fingerprint generation without increasing the size of fingerprint. In addition, searching cost is reduced significantly with optimized code implementation in this submission.

1. INTRODUCTION

Audio fingerprinting has been widely used in various applications such as music retrieval for end users and detection of copyright infringement for music companies, etc. It is a typical case of query-by-example. There exist mainly two competitive approaches in the past several years. The first one, developed by Haitsma [1] in 2002, is based on the hamming distance between binarized spectrograms of query and db tracks. The other one, developed by Wang [2] in 2003, is based on the counts of aligned landmarks in spectrograms. In our submission we adopted the latter approach due to its robustness and conciseness.

As discussed in our previous submissions, Wang [2] proposed a method to detect the landmarks in spectrogram in which a landmark was defined as a local maxima. Then these landmarks are made as pairs according to a predefined scheme and hashed thereafter. As a result, each pair corresponds to a hash value along with a time stamp of the first landmark (i.e., anchor landmark) of this pair. This is the so called fingerprint defined in Wang's method. For efficient matching, fingerprints are organized as inverted index where hash value is used as key and time stamp is stored in the index. During matching, for each query fingerprint we access the inverted index and calculate the time difference between db tracks and input query. After that the histogram of time difference is analyzed for each candidate db track, resulting a final output if a predefined threshold has been exceeded.

This document is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 License. http://creativecommons.org/licenses/by-nc-sa/3.0/
© 2020 The Authors

Since queries are usually collected in noisy environment, the main challenge to a successful audio fingerprint system is noise robustness and searching efficiency in a music database with millions of songs. To overcome these challenges, we have made some modifications to our previous submissions on landmark detection and hashing strategy, which will be explained in Section 2.

2. FINGERPRINTS EXTRACTION

As in our previous submissions, audio segments of db tracks or query recordings are firstly re-sampled to make sure the sampling rate are the same for all the segments. Usually 8khz is enough for audio fingerprinting but one can use higher sampling rate at the expense of higher computational cost.

To extract landmarks, the re-sampled audio data is firstly converted to time-frequency domain with STFT. Then, following Wang's idea, we detect the local maxima in spectrogram. In our latest implementation, an adaptive window-based algorithm is employed in which a local maxima is selected only if it is higher than all the other locations within the window in spectrogram. Then a double check is applied to determine whether the selected local maxima should be preserved. For example, if the window has relatively low energy and the local maxima is not high enough, it is then discarded. Meanwhile, window size is adaptively adjusted depending on the local energy of the segments. For example, when detecting landmarks at the *j*th frame, we calculate the energy of the time range [*j*-*k*, *j*+*k*], where *k* is the context size. If this energy value is small, we use a small window size for the detection of local maxima.

Once detection is done, for each anchor landmark we select a list of reference landmarks within its target zone, as described in [2]. One of the modifications we have made here is that a landmark could be a reference landmark at most M times, where M is a configurable parameter.

3. MATCHING

The matching process remains unchanged in this submission. For more details please refer to the document we submitted in 2017.

4. SYSTEM IMPLEMENTATION

Both builder and matcher are implemented in C++. Parallel processing is supported for acceleration. Please follow the instructions described in readme when running the programmes.

5. DISCUSSION

In this document we present our submission to audio fingerprinting task of MIREX 2020. It is an improved version of our previous submissions. We have run intensive tests and the experimental results have shown a notable improvement on our own datasets.

6. REFERENCES

[1] J. Haitsma and T. Kalker: "A highly robust audio fingerprinting system". In 3rd Int. Conf. on Music Information Retrieval ISMIR 2002, 2002.

[2] A. Wang: “An industrial strength audio search algorithm,” *Proceedings of the International Symposium on Music Information Retrieval*, 2003.