Panako - A Scalable Acoustic Fingerprinting System Handling Time-Scale and Pitch Modification

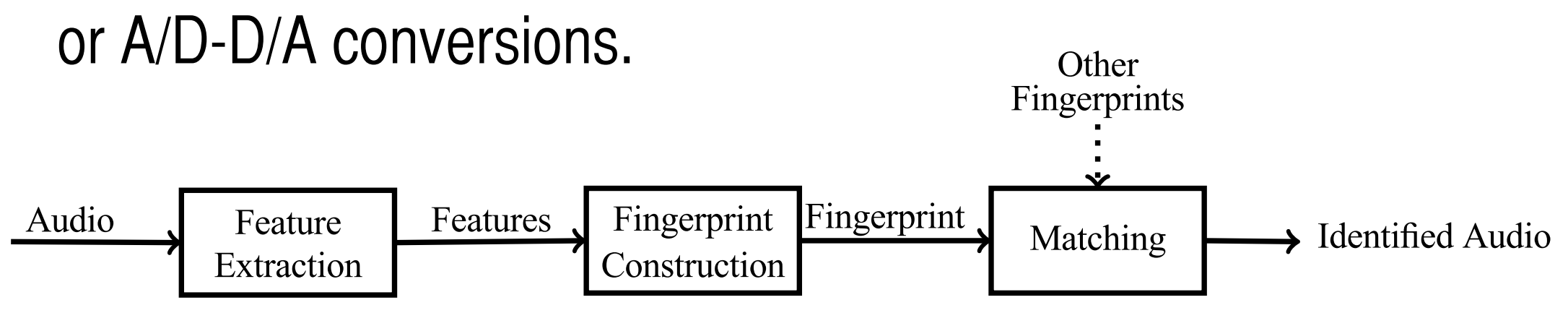
Joren Six and Marc Leman - joren.six@ugent.be - IPEM, University Ghent - Belgium

Abstract

This poster presents a scalable granular acoustic fingerprinting system. An acoustic fingerprinting system uses condensed representations of audio signals, acoustic fingerprints, to identify short audio fragments in large audio databases. The system presented here is shown to answer queries quickly and reliably even when queries are subjected to time-scale and pitch modifications. The design of this system is the main contribution of this research.

Introduction

Acoustic fingerprinting systems have many practical uses cases. They follow the scheme depicted in Figure 1. Ideally, a fingerprinting system only needs a short audio fragment to find a match in large set of reference audio. One of the challenges is to design a system in a way that the reference database can grow to contain millions of entries. Another challenge is that a robust fingerprinting should handle noise and other modifications well, while limiting the amount of false positives and processing time [1]. These modifications typically include dynamic range compression, equalization, added background noise and artifacts introduced by audio coders or A/D-D/A conversions.


Audio → Feature Extraction → Features → Fingerprint Construction → Fingerprint → Matching → Identified Audio

Other Fingerprints → Matching

Fig 1. General acoustic fingerprinting scheme.

Over the years several efficient acoustic fingerprinting methods have been introduced [2,3]. These methods perform well, even with degraded audio quality and with industrial sized reference databases. However, these systems are not designed to handle queries with modified time-scale or pitch although these distortions can be present in replayed material. During radio broadcasts songs are occasionally played faster to make them fit into a time slot. During a DJ-set pitch-shifting and time-stretching are present almost continuously. To correctly identify audio in these cases as well, a fingerprinting system robust against pitch-shifting and time-stretching is desired.

Some fingerprinting systems have been developed that take pitch-shifts into account [6]. Others are designed to handle both pitch and time-scale modification[8,9]. To find a match, these computationally expensive systems iterate the whole database. To the best of our kowledge, a description of a practical fingerprinting system that allows substantial pitch-shift and time-scale modification can only be found in [7], and in this work.

Method

The proposed method is inspired by three works[2,4,6]. Combining key components of those works results in a design of a granular acoustic fingerprinter that is robust to noise and substantial compression, has a scalable method for fingerprint storage and matching, and allows time-scale modification and pitch-shifting.

The method presented here uses local maxima in a spectral representation[2]. It combines three event points, and takes time ratios to form time-scale invariant fingerprints[4]. It leverages the Constant-Q transform, and only stores frequency differences for pitch-shift invariance[6]. The fingerprints are designed with an exact hashing matching algorithm in mind[2]. The whole process is depicted in Figure 2.

Feature extraction: the first step is to transform the audio to a spectral representation. The Constant-Q transform is used to get an equal amount of frequency bins in each octave. This is depicted in Figure 2a. The next step is to locate peaks within the time-frequencyplane (Fig 2b). This is done using a 2D peak extraction algorithm.

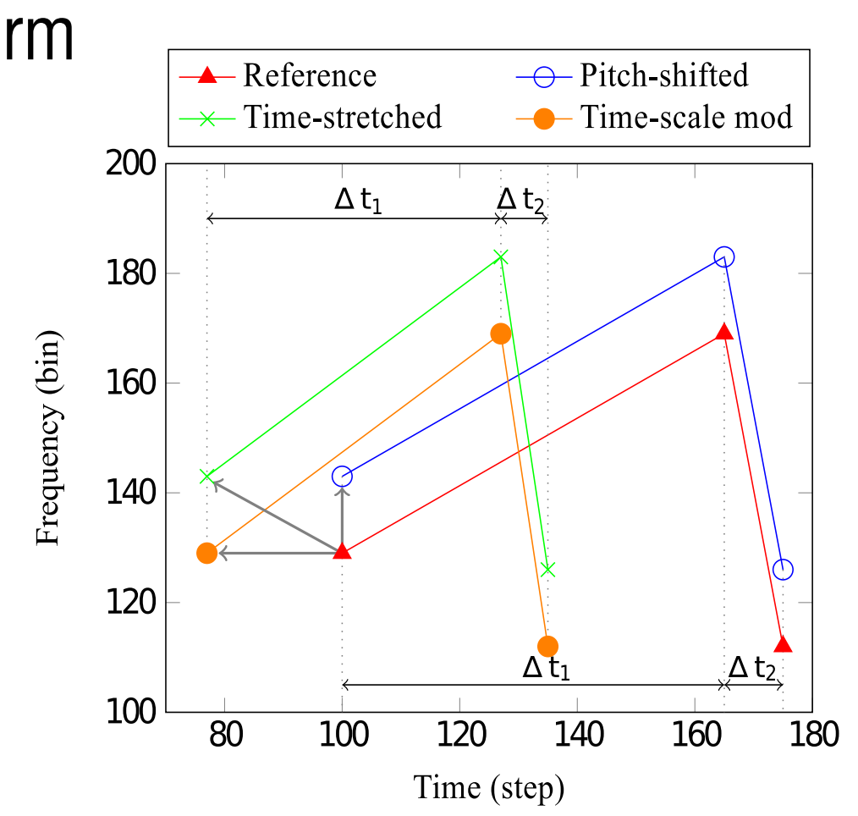


Fig 3. The effect of time-scale and pitch modifications on a fingerprint.

Fingerprint construction: to form fingerprints, three peaks are combined, as in Figure 2c. The effects on a fingerprint extracted from reference audio and a fingerprint extracted from the same audio after pitch-shifting, time-stretching and time-scale modification can be seen in Figure 3. The ratios between the time differences and the differences between the frequency components are invariant. These constants are employed in the fingerprint hash. To add discriminative power to the hash, coarse frequency location indicators are included as well. These hashes are stored in a B-tree:

$$\left(f_1 - f_2; f_2 - f_3; \tilde{f}_1; \tilde{f}_3; \frac{t_2-t_1}{t_3-t_1}\right); t_1; f_1; t_3 - t_1; id$$

Matching: to match a query with the reference audio, fingerprints, and their corresponding hashes, are extracted. For each hash, matching reference audio items are fetched from the datastore. Random matches are removed from this resultset by only keeping reference audio items that are present multiple times. To limit the amount of false positives, alignment in time is checked. Also, a match is only marked as valid if the time stretch-factor and pitch-shift factor between query and reference audio are constant. Finally, if a match is found, the reference audio identifier is returned.

Results & Conclusions

The system has been evaluated using a freely available data set of 30,000 songs and compared with a baseline system (see Figure 4,5,6,7).

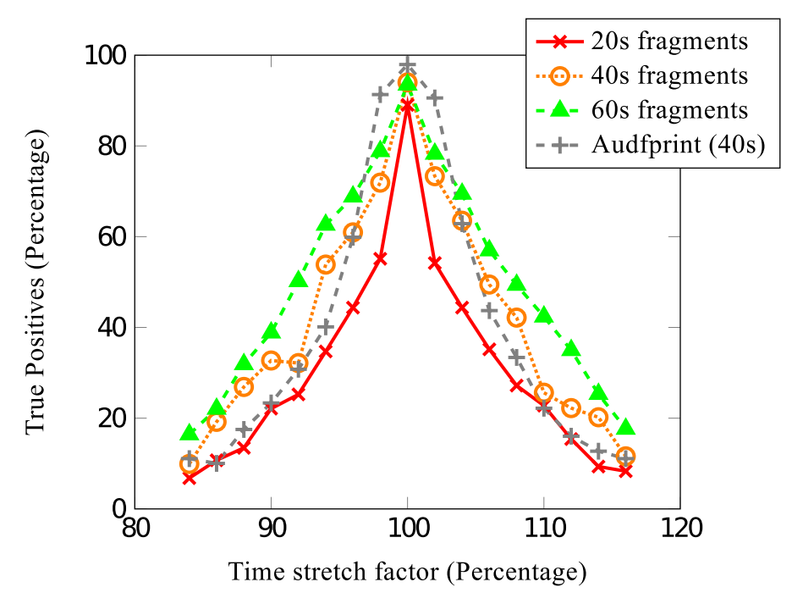


Fig 4. The effect of time-stretching on retrieval performance.

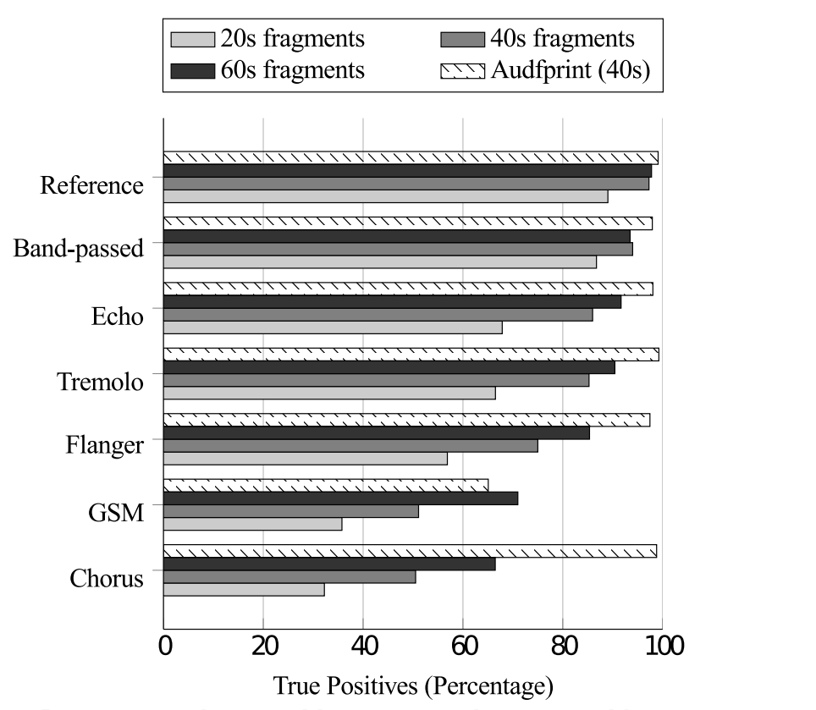


Fig 5. The effect various effects on retrieval performance.

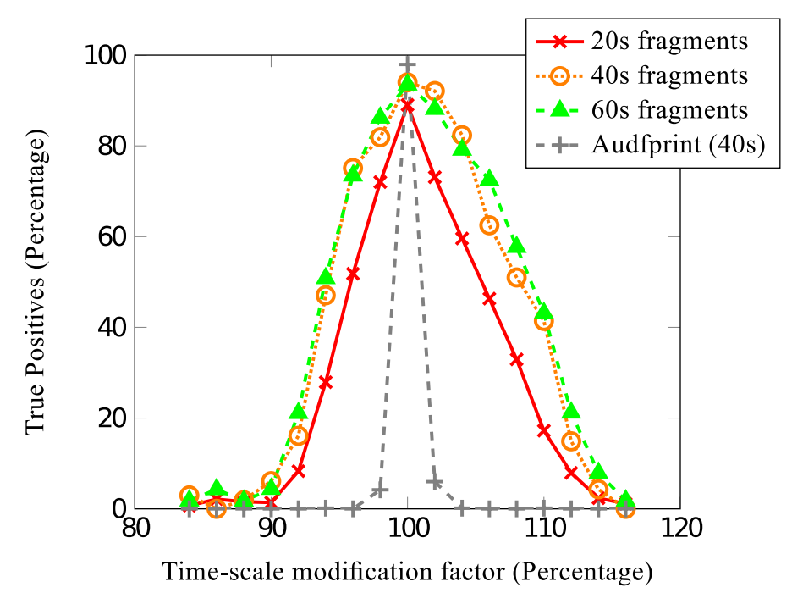


Fig 6. The effect of time-scale modification on retrieval performance.

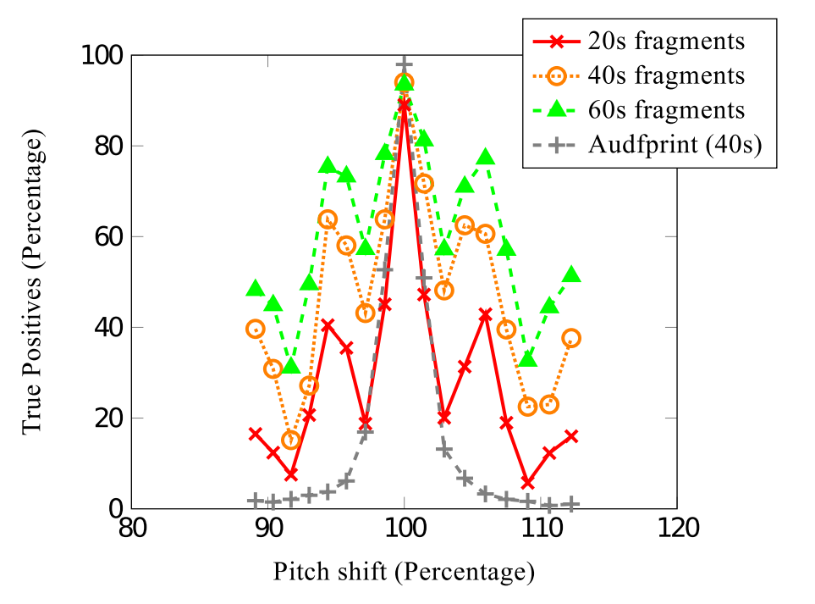


Fig 7. The effect of pitch shift on retrieval performance.

This work presented a practical acoustic fingerprinting system. The system allows fast and reliable identification of small audio fragments in a large set of audio, even when the fragment has been pitch-shifted and time-stretched with respect to the reference audio. If a match is found the system reports where in the reference audio a query matches, and how much time/frequency has been modified. To achieve this, the system uses local maxima in a Constant-Q spectrogram. It combines event points into groups of three, and uses time ratios to form a time-scale invariant fingerprint component. To form pitch-shift invariant fingerprint components only frequency differences are stored. For retrieval, an exact hashing matching algorithm is used.

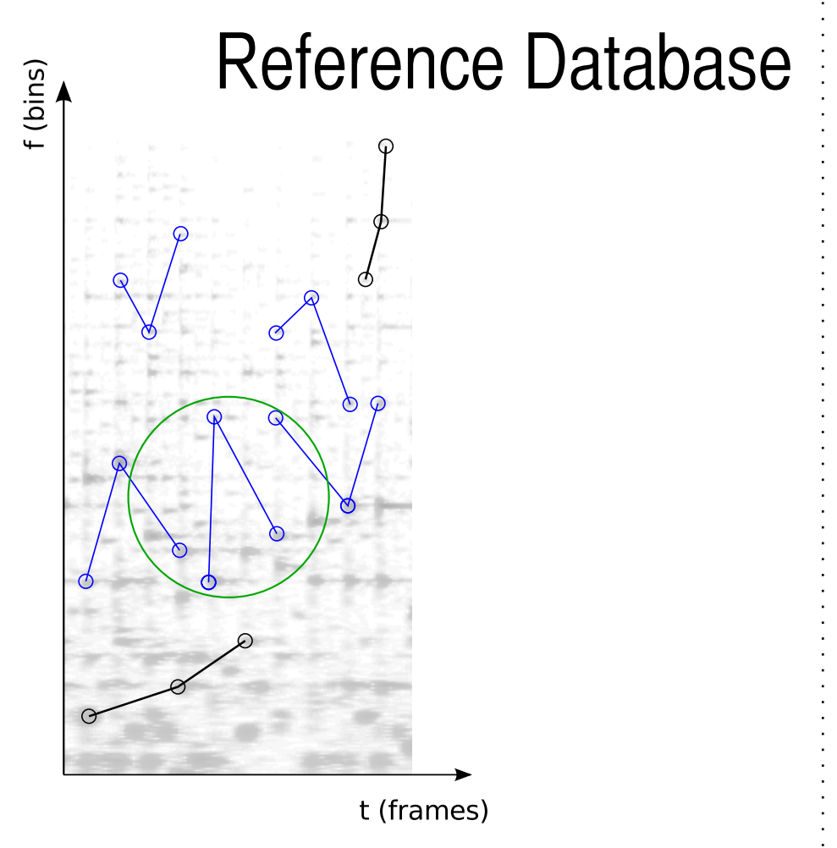


Fig 2e. Fingerprint hases in the reference database. Here, the query is time-streched.

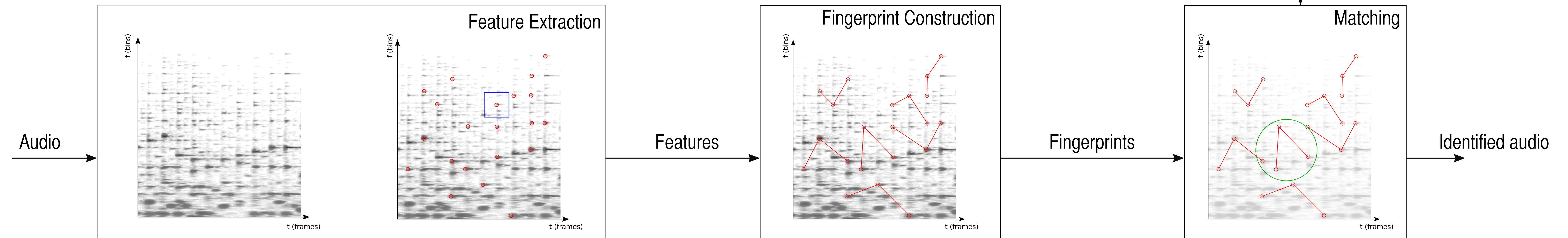


Fig 2a. Step 1, Constant-Q transform of the incoming audio.

Fig 2b. Step 2, Extract event points from the spectrogram with a tiled 2D peak extraction algorithm.

Fig 2c. Step 3, Connect the event points in sets of three. Each triplet forms a fingerprint.

Fig 2d. Step 4, Each fingerprint is Hashed. The hashes are matched with the reference database.

Fig 2. The Panako fingerprinting system combines triplets of peaks in a Constant-Q spectrogram to form fingerprints.

References

[1] P. Cano, E. Batlle, T. Kalker, and J. Haitsma. A review of audio fingerprinting. The Journal of VLSI Signal Processing, 2005.
[2] A. L. Wang. An Industrial-Strength Audio Search Algorithm. In Proceedings of ISMIR 2003, 2003.
[3] D. Ellis, B. Whitman, and A. Porter. Echoprint - an open music identification service. In Proceedings ISMIR 2011, 2011.
[4] A. Arzt, S. Böck, and G. Widmer. Fast identification of piece and score position via symbolic fingerprinting. Proceedings of ISMIR 2012, 2012
[5] J. Six, O. Cornelis, and M. Leman. TarsosDSP, a Real-Time Audio Processing Framework in Java. In Proceedings of the 53rd AES Conference, 2014.
[6] S. Fenet, G. Richard, and Y. Grenier. A Scalable Audio Fingerprint Method with Robustness to Pitch-Shifting. In Proceedings of ISMIR 2011, 2011
[7] R. Sonnleitner, G. Widmer. Quad-based Audio Fingerprinting Robust to Time and Frequency Scaling. Proceedings of the Conference on Digital Audio Effects (DAFx 2014), September 2014
[8] B. Zhu, W. Li, Z. Wang, and X. Xue. A novel audio fingerprinting method robust to time scale modification and pitch shifting. In Proceedings of the conference on Multimedia. ACM, 2010.
[9] M. Malekesmaeili and R. K. Ward. A local fingerprinting approach for audio copy detection. Computing Research Repository (CoRR), 2013.

Availability & Reproducibility

The Panako software is available on http://panako.be under the AGPL. Panako is tested on Debian and Mac OS X, but should work on every platform with a recent Java Runtime Environment.

The dataset used in the validation is freely available from Jamendo.com, a website where artists share their work freely, under various creative commons licenses.

To reproduce the results, scripts are available to download the audio dataset, generate query files, store the reference audio and query the system. Supporting tools to analyse the query results are available as well.

.

15th ISMIR Conference, Taipei, Taiwan, 27-31 October, 2014