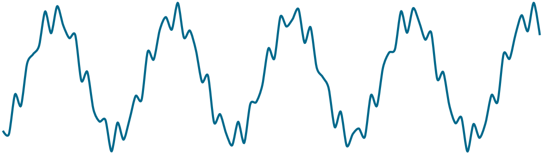
Audio Fingerprinting

Opportunities for digital musicology

Joren Six

IPEM, University Ghent



November 27, 2014

Overview

What is Audio Fingerprinting

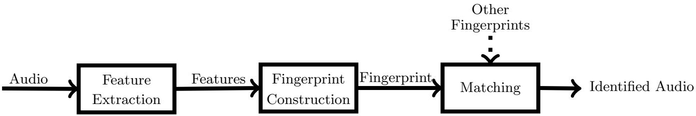


Figure: A generalized audio fingerprinter scheme.

1. Audio is fed into the system,
2. Features are extracted and fingerprints constructed
3. The fingerprints are compared with a database containing fingerprints of reference audio.
4. The audio is either identified or, if no match is found, labeled as unknown.

Why Audio Fingerprinting?

Fig: Shazam music recognition service

- Identifying short audio fragments
- Duplicate detection in large digital music archives
- Digital rights management applications (SABAM)
- Music structure analysis
- Analysis of techniques and repertoire in DJ-sets
- Synchronization of audio (and video) streams

Desired Properties of an Audio Fingerprinter System

An ideal fingerprinting system has the following properties [1]:

- **Random, short query fragments** can be identified correctly.
- It has **good query performance**. Matching fragments against a large data set, of *millions of songs*, is done within milliseconds.
- **Storage requirements** for fingerprints are minimal.
- **Extracting fingerprints** from audio is computationally inexpensive.
- Additional **noise** or other artefacts in queries do not affect retrieval performance.
- The system does not yield **false positives**. A fingerprinting system should be reliable.

Audio Fingerprinter System Design

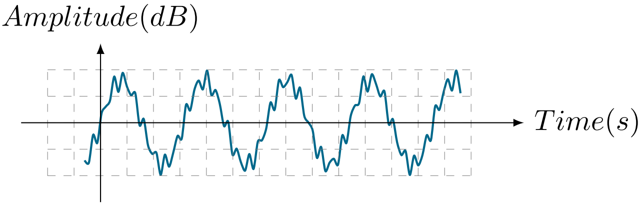


Fig: Waveform of a sound.

Features that can be employed to construct a fingerprint:

- Frequency - Pitch - *melody* - *harmony*
- Onsets - *beats* - *pattern* - *tempo* - *rhythm*
- Spectrum - timbre - *instrumentation*
- Intensity - loudness - dynamics

Audio Fingerprinter System Design

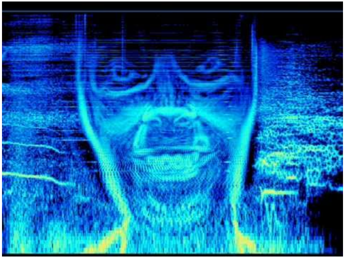


Fig: Spectrogram in Aphex Twin's *Windowlicker*

Current audio fingerprinting systems use fingerprints based on:

- **Spectral Peaks [8, 7, 4]**
- Onsets in spectral bands [3]
- Other features [1, 5, 6, 2]

Fingerprinting with Spectral Peaks

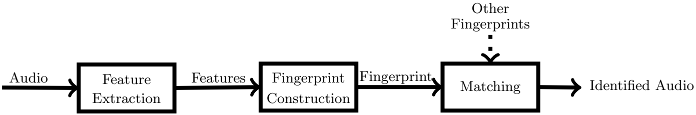


Figure: A generalized audio fingerprinter scheme.

An audio fingerprinter based on spectral peaks[8] follows the general fingerprinting scheme:

1. Audio is fed into the system
2. A *spectrogram* is extracted and fingerprints are constructed using a combination of *two spectral peaks*
3. The fingerprints are compared with a database containing fingerprints of reference audio.
4. The audio is either identified or, if no match is found, labeled as unknown.

Fingerprinting with Spectral Peaks

LIVE DEMO

Step 1: Feature Extraction

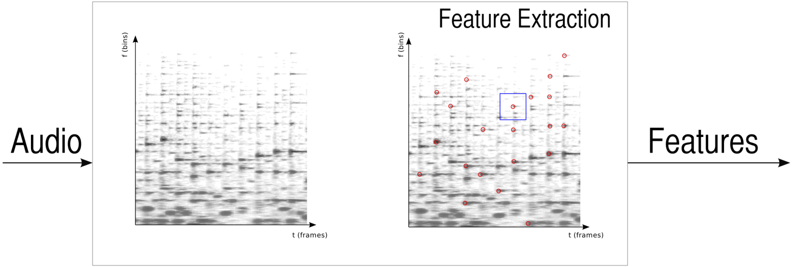


Fig: After an FFT analysis on sound, spectral peaks are extracted.

Step 2: Fingerprint Construction

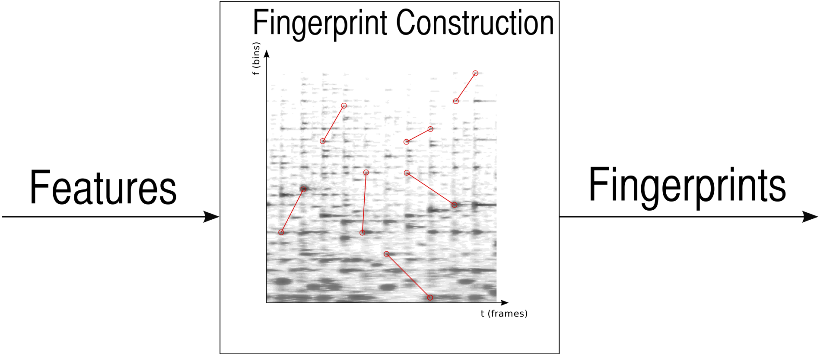


Fig: Detecting Key Points

Step 2: Fingerprint Construction

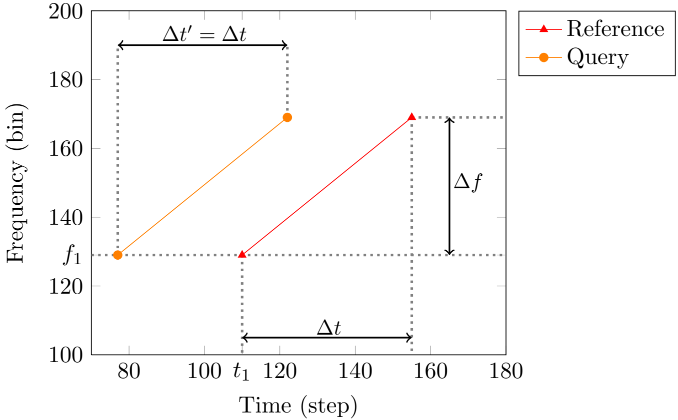


Figure: A fingerprint

Fig: A Fingerprint consists of: $id; t_1; hash(f_1; \Delta f; \Delta t)$

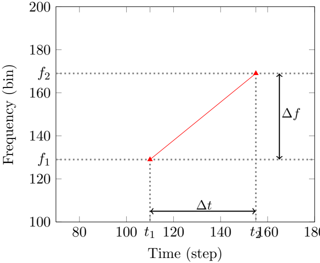
Step 2: Fingerprint Construction

Save every fingerprint by combining $f_1; \Delta f; \Delta t$ with the identifier of a song id.

- f_1 in $[0 - 256]$
- Δf in $[0 - 64]$
- Δt in $[0 - 512]$

One fingerprint hash fits in an integer 2^{32}. An audio identifier and t_1 can be encoded using an integer as well.
With 10 landmarks per seconds and $100k$ songs and on average 4mins per song this means:
$10/s \times 100000 \times 4 \times 60s \times 3 \times 32bits = 2.7GB$

Step 2: Fingerprint Construction, Example



$(t_1, f_1) = (110, 129)$, $(t_2, f_2) = (155, 169)$,$\Delta t = 45$, $\Delta f = 40$

Hash function $hash(f, \Delta f; \Delta t) = f + \Delta f \times 10^3 + \Delta t \times 10^6$

$id; t_1; hash(f_1; \Delta f; \Delta t) = 1452; 110; hash(129; 45; 40)$
$id; t_1; hash(f_1; \Delta f; \Delta t) = 1452; 110; 40045129$

Step 3: Matching

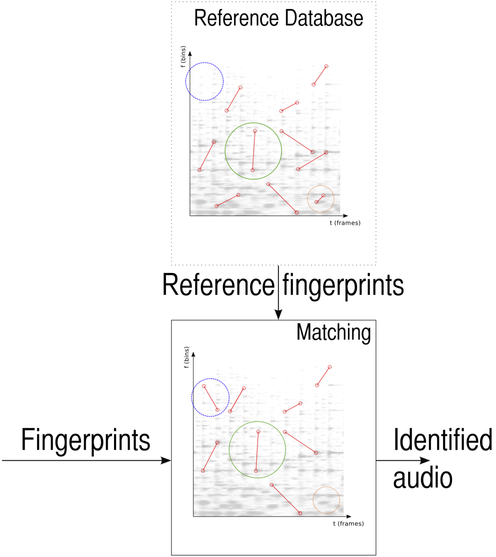


Fig: Matching fingerprints with the reference database

Step 3: Matching

1. Extract fingerprints from query
2. Compute hashes from query
3. Retrieve all matches from reference dataset
4. Order the matches by number of matching audio identifiers. Ignore random chance hits by ignoring audio identifiers that only occur one or a few times (4).
5. Check if the matches appear in the correct order in both query and reference.
6. Return the match.

Step 3: Matching

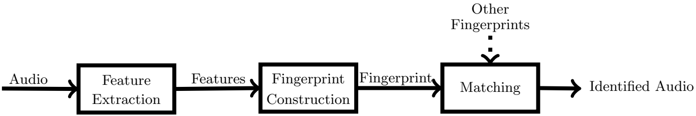


Fig: Generalized fingerprinting scheme

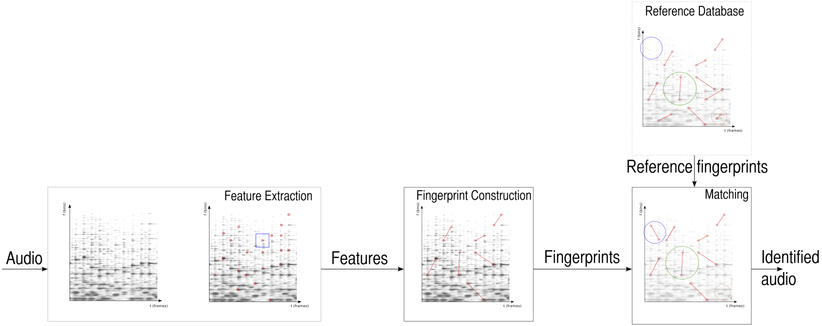


Fig: Spectral peak based fingerprinting scheme

Opportunities for digital musicology

Acoustic fingerprinting can provide opportunities for digital musicology:

1. Analysis of repetition within songs
2. Comparison of versions/edits
3. Audio and audio feature alignment to share datasets
4. DJ-set analysis

Musical structure analysis

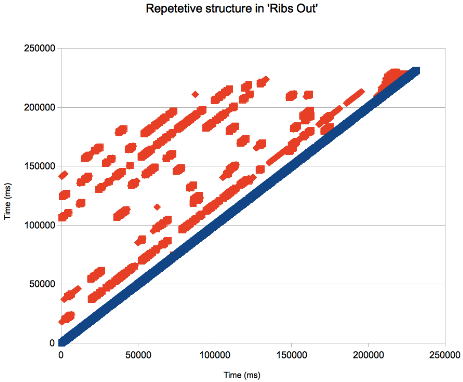


Fig: Repetition in 'Ribs Out' by Fuck Buttons[1].

[1] Unfortunately the best example I could find

Radio Edit vs. Original

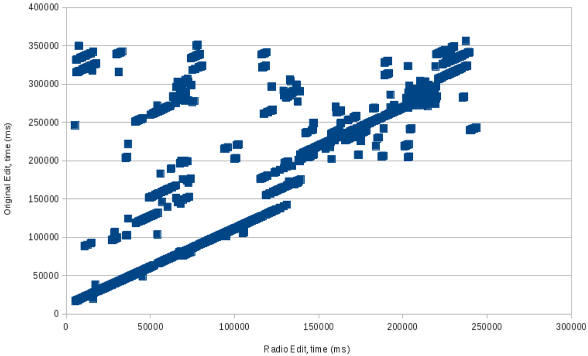


Fig: Radio edit vs. original version of Daft Punk's *Get Lucky*.

Exact Repetition Over Time

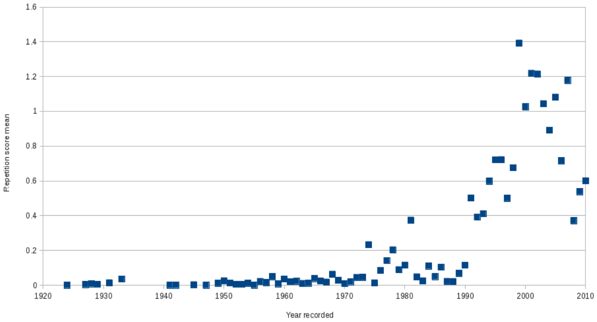


Fig: How much *cut-and-paste* is used on average for a set of 20000 recordings.

Synchronization of audio streams

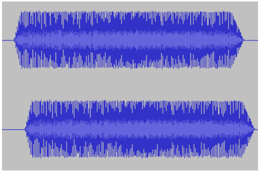


Fig: Two similar audio streams out of sync

Audio synchronization can be used for:

- Aligning unsynchronized audio streams from several microphones
- Aligning video footage by using audio
- **Aligning audio and extracted features**[a]

[a] e.g. http://acoustid.org/, http://echonest.com, http://acousticbrainz.org/

Synchronization of audio streams

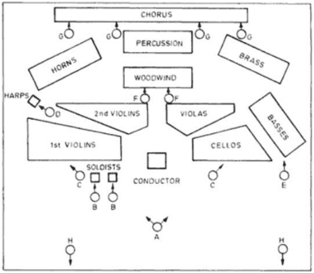


Fig: Microphone placement for symphonic orchestra and synchronization

Audio synchronization using acoustic fingerprinting is *submillisecond accurate*. If microphone placement spans several meters and with the speed of sound being 340.29m/s:

Distance (m)	Delay (ms)
1	3
2	6
3	9

Analysis of repertoire and techniques used in DJ-Sets

Fig: a DJ

An extension of the spectral peak fingerprinting method allows time-stretching, pitch-shifting and tempo change[7]. Given a DJ-set and reference audio[a] the following can be extracted automatically:

- Which parts of which songs were played and for how long
- Which modifications were applied (percentage modification of time and frequency)

[a] Tracklists of DJ-Sets can be found on `http://www.1001tracklists.com/`

Practical Audio Fingerprinting

Panako[7] was used to generate the example data[2]. It is an open source audio fingerprinting system available on `http://panako.be`. To use Panako the Java JRE needs to be installed.

More specifically the these subapplications were used:

- `monitor` during the live demo
- `compare` for the comparison, structure analysis
- `monitor` can also be used for DJ-set analysis.

Other fingerprinters are audfprint and echoprint.

[2]Some methods implemented within Panako are patented (US6990453).

Bibliography I

Pedro Cano, Eloi Batlle, Ton Kalker, and Jaap Haitsma.
A review of audio fingerprinting.
The Journal of VLSI Signal Processing, 41:271–284, 2005.

Michele Covell and Shumeet Baluja.
Known-Audio Detection using Waveprint: Spectrogram Fingerprinting by Wavelet Hashing.
In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)*, 2007.

Dan Ellis, Brian Whitman, and Alastair Porter.
Echoprint - an open music identification service.
In *Proceedings of the 12th International Symposium on Music Information Retrieval (ISMIR 2011)*, 2011.

Bibliography II

Sébastien Fenet, Gaël Richard, and Yves Grenier.
A Scalable Audio Fingerprint Method with Robustness to Pitch-Shifting.
In *Proceedings of the 12th International Symposium on Music Information Retrieval (ISMIR 2011)*, pages 121–126, 2011.

Jaap Haitsma and Ton Kalker.
A highly robust audio fingerprinting system.
In *Proceedings of the 3th International Symposium on Music Information Retrieval (ISMIR 2002)*, 2002.

Bibliography III

M. Ramona and G. Peeters.
AudioPrint: An efficient audio fingerprint system based on a novel cost-less synchronization scheme.
In *Proceedings of the 2013 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP 2013)*, pages 818–822, 2013.

Joren Six and Marc Leman.
Panako - A Scalable Acoustic Fingerprinting System Handling Time-Scale and Pitch Modification.
In *Proceedings of the 15th ISMIR Conference (ISMIR 2014)*, 2014.

Bibliography IV

Avery L. Wang.
An Industrial-Strength Audio Search Algorithm.
In *Proceedings of the 4th International Symposium on Music Information Retrieval (ISMIR 2003)*, pages 7–13, 2003.