A Short Introduction to Audio Fingerprinting

Simon Froitzheim

RWTH Aachen University

simon.froitzheim@rwth-aachen.de

June 28th, 2017
Overview

1. Audio Fingerprinting Basics
2. An Algorithmic Overview
3. Shazam
   - Overview
   - Technical Details
4. Conclusive Remarks
   - Synopsis
   - References
Audio Fingerprinting Basics
Central Issue

- **Given**: An unlabeled piece of audio in any format
Central Issue

- **Given**: An unlabeled piece of audio in any format
- **Goal**: To encode the given audio piece in a so-called "fingerprint"
Central Issue

- **Given**: An unlabeled piece of audio in any format
- **Goal**: To encode the given audio piece in a so-called "fingerprint"

Why is this useful?
Applications for Audio Fingerprinting

- Content-based audio identification (CBID)
Applications for Audio Fingerprinting

- Content-based audio identification (CBID)
- Content-based integrity verification
Applications for Audio Fingerprinting

- Content-based audio identification (CBID)
- Content-based integrity verification
- Watermarking support
Applications for Audio Fingerprinting

- Content-based audio identification (CBID)
- Content-based integrity verification
- Watermarking support

All of this implies storage, indexing and comparison of fingerprints!
Content-based Audio Identification

**Goal:** To identify audio tracks solely based on the track itself (without any given metadata)
Content-based Audio Identification

Goal: To identify audio tracks solely based on the track itself (without any given metadata)

This can be achieved by:
Content-based Audio Identification

**Goal**: To identify audio tracks solely based on the track itself (without any given metadata)

This can be achieved by:

- Computing a fingerprint for every known audio piece
Content-based Audio Identification

**Goal**: To identify audio tracks solely based on the track itself (without any given metadata)

This can be achieved by:

- Computing a fingerprint for every known audio piece
- Storing each fingerprint in a database
Content-based Audio Identification (continued)

When confronted with an unknown audio excerpt:
Content-based Audio Identification (continued)

When confronted with an unknown audio excerpt:

- Compute the corresponding fingerprint
Content-based Audio Identification (continued)

When confronted with an unknown audio excerpt:
- Compute the corresponding fingerprint
- Match it against the database
Content-based Audio Identification (continued)

When confronted with an unknown audio excerpt:

- Compute the corresponding fingerprint
- Match it against the database
- Identify music piece based on the matching
Requirements for Audio Fingerprinting Techniques
[Cano, 2002]

- Discrimination power
Requirements for Audio Fingerprinting Techniques [Cano, 2002]

- Discrimination power
- Robustness
Requirements for Audio Fingerprinting Techniques
[Cano, 2002]

- Discrimination power
- Robustness
  - Cropping
Requirements for Audio Fingerprinting Techniques
[Cano, 2002]

- Discrimination power
- Robustness
  - Cropping
  - Distortions
Requirements for Audio Fingerprinting Techniques

[Cano, 2002]

- Discrimination power
- Robustness
  - Cropping
  - Distortions
- Compactness
Requirements for Audio Fingerprinting Techniques
[Cano, 2002]

- Discrimination power
- Robustness
  - Cropping
  - Distortions
- Compactness
- Computational simplicity
Requirements for Audio Fingerprinting Techniques
[Cano, 2002]

- Discrimination power
- Robustness
  - Cropping
  - Distortions
- Compactness
- Computational simplicity

The usual trade-off between reliability and efficiency is present here!
An Algorithmic Overview
Various procedures were developed since the very late 20th century
Various procedures were developed since the very late 20th century.
Most use the same general framework.
Various procedures were developed since the very late 20th century
Most use the same general framework
Of high commercial interest for music companies
Various procedures were developed since the very late 20th century
Most use the same general framework
Of high commercial interest for music companies
Especially relevant for mobile devices
State-of-the-Art

- The Philips technique [Haitsma, 2002]
State-of-the-Art

- The Philips technique [Haitsma, 2002]
- **Shazam** [Wang, 2003]
State-of-the-Art

- The Philips technique [Haitsma, 2002]
- **Shazam** [Wang, 2003]
- Google waveprint [Baluja, 2007]
State-of-the-Art

- The Philps technique [Haitsma, 2002]
- **Shazam** [Wang, 2003]
- Google waveprint [Baluja, 2007]
- MASK [Anguera, 2012]
Shazam
Application Details

- Free mobile app that recognizes music, TV and media based on short audio snippets recorded with your phone
Application Details

- Free mobile app that recognizes music, TV and media based on short audio snippets recorded with your phone
- Features special camera interaction for interactive experiences and additional content
Free mobile app that recognizes music, TV and media based on short audio snippets recorded with your phone

Features special camera interaction for interactive experiences and additional content

Possible connections to Google, Snapchat, Facebook, Spotify,...
Application Details

- Free mobile app that recognizes music, TV and media based on short audio snippets recorded with your phone
- Features special camera interaction for interactive experiences and additional content
- Possible connections to Google, Snapchat, Facebook, Spotify,...
- Encourages music purchases
Demonstration

Let’s see Shazam in action!
How Does it Work? - An Overview

The main steps of Shazam are:

- Spectogram computation
How Does it Work? - An Overview

The main steps of Shazam are:

- Spectogram computation
- Constellation map construction
How Does it Work? - An Overview

The main steps of Shazam are:

- Spectogram computation
- Constellation map construction
- Combinatorial hashing
How Does it Work? - An Overview

The main steps of Shazam are:

- Spectogram computation
- Constellation map construction
- Combinatorial hashing
- Database searching
The main steps of Shazam are:

- Spectogram computation
- Constellation map construction
- Combinatorial hashing
- Database searching
- Scoring of possible matches
Spectograms

Visually represent the signal strength (energy) of a signal over time at various frequencies.

Can also be used for sound waves.
Spectograms

- Visually represent the signal strength (energy) of a signal over time at various frequencies
Spectograms

- Visually represent the signal strength (energy) of a signal over time at various frequencies
- Can also be used for sound waves
From Spectograms to Constellation Maps

Shazam chooses high energy candidate peaks with respect to density.

Why high amplitudes? Robustness!

Results in a constellation map.
Shazam chooses high energy candidate peaks with respect to density.
From Spectograms to Constellation Maps

- Shazam chooses high energy candidate peaks with respect to density
- Why high amplitudes? Robustness!
From Spectograms to Constellation Maps

- Shazam chooses high energy candidate peaks with respect to density
- Why high amplitudes? Robustness!
- Results in a constellation map
Combinatorial Hashing

Anchor points and target zones are determined.
Pairwise hashing between anchor and points in the zone.
Two frequency components plus the time difference form one hash.
Combinatorial Hashing

- Anchor points and target zones are determined
Combinatorial Hashing

- Anchor points and target zones are determined
- Pairwise hashing between anchor and points in the zone
Combinatorial Hashing

- Anchor points and target zones are determined
- Pairwise hashing between anchor and points in the zone
- Two frequency components plus the time difference form one hash
Why is Hashing Necessary?

- **Problem**: Constellation maps are very sparse which results in long matching times
Why is Hashing Necessary?

- **Problem**: Constellation maps are very sparse which results in long matching times
- Hashing enables the usage of 64-Bit structs (32 bits hash, 32 bits time offset and track ID)
Why is Hashing Necessary?

- **Problem**: Constellation maps are very sparse which results in long matching times.
- Hashing enables the usage of 64-Bit structs (32 bits hash, 32 bits time offset and track ID).
- Limited number of points in every target zone to limit combinatorial explosion.
Why is Hashing Necessary?

- **Problem**: Constellation maps are very sparse which results in long matching times.
- Hashing enables the usage of 64-Bit structs (32 bits hash, 32 bits time offset and track ID).
- Limited number of points in every target zone to limit combinatorial explosion.
- Overall trade: 10 times more disk space for 10000 times faster matching.
Searching and Scoring

- Matching of all sample hashes with database hashes
Searching and Scoring

- Matching of all sample hashes with database hashes
- Association of time pairs for every matching hash (both offset times)
Searching and Scoring

- Matching of all sample hashes with database hashes
- Association of time pairs for every matching hash (both offset times)
- Distribution of time pairs into bins according to track ID
Searching and Scoring

- Matching of all sample hashes with database hashes
- Association of time pairs for every matching hash (both offset times)
- Distribution of time pairs into bins according to track ID
- Construction of a scatterplot of association between sample and database files
Searching and Scoring

- Matching of all sample hashes with database hashes
- Association of time pairs for every matching hash (both offset times)
- Distribution of time pairs into bins according to track ID
- Construction of a scatterplot of association between sample and database files
- Detection of point clusters that form a diagonal line
A Successful Match

Scatterplot of matching hash locations: Diagonal Present
Conclusive Remarks
Audio fingerprinting aims for the construction of a compact, discriminative, robust and efficient encoding of audio data.
Audio fingerprinting aims for the construction of a compact, discriminative, robust and efficient encoding of audio data.

Works on very small song snippets reliably.
Audio fingerprinting aims for the construction of a compact, 
discriminative, robust and efficient encoding of audio data
Works on very small song snippets reliably
Shazam as an exemplary algorithm is based on constellation maps 
and uses combinatorial hashing for speedup
References

J. Haitsma and T. Kalker (2002)
A Highly Robust Audio Fingerprinting System
*Journal of New Music Research* 32 (2), 211 – 221

E. Batlle, P. Cano, J. Haitsma and T. Kalker (2002)
A Review of Algorithms for Audio Fingerprinting
*IEEE Workshop on Multimedia Signal Processing 2002*

An Industrial-Strength Audio Search Algorithm
*Proc. of the 4th International Conference on Music Information Retrieval*

S. Baluja and M. Covell (2007)
Audio Fingerprinting: Combining Computer Vision & Data Stream Processing
*Acoustics, Speech and Signal Processing*
X. Anguera, A. Garzon and T. Adamek (2012)
MASK: Robust Local Features for Audio Fingerprinting
*IEEE International Conference on Multimedia and Expo*

All graphics in this presentation are taken from [Wang, 2003].
Thank you for listening