



Introduction to **Audio Content Analysis**

module 11.0: audio fingerprinting

alexander lerch

introduction

overview

corresponding textbook section

chapter 11

■ lecture content

- introduction to audio fingerprinting
- in-depth example for fingerprint extraction and retrieval

■ learning objectives

- discuss goals and limitations of audio fingerprinting systems as compared to watermarking or cover song detection systems
- describe the processing steps of the Philips fingerprinting system



introduction

overview

corresponding textbook section

chapter 11

■ lecture content

- introduction to audio fingerprinting
- in-depth example for fingerprint extraction and retrieval

■ learning objectives

- discuss goals and limitations of audio fingerprinting systems as compared to watermarking or cover song detection systems
- describe the processing steps of the Philips fingerprinting system



audio fingerprinting

introduction

■ objective:

- represent a recording with a compact and unique digest
(→ *fingerprint, perceptual hash*)
- allow quick matching between previously stored fingerprints and an extracted fingerprint

■ applications:

- *broadcast monitoring*:
automate verification for royalties/infringement claims
- *value-added services*:
offer information and meta data

audio fingerprinting

introduction

■ objective:

- represent a recording with a compact and unique digest
(→ *fingerprint, perceptual hash*)
- allow quick matching between previously stored fingerprints and an extracted fingerprint

■ applications:

- *broadcast monitoring*:
automate verification for royalties/infringement claims
- *value-added services*:
offer information and meta data

audio fingerprinting

introduction

■ objective:

- represent a recording with a compact and unique digest
(→ *fingerprint, perceptual hash*)
- allow quick matching between previously stored fingerprints and an extracted fingerprint

■ applications:

- *broadcast monitoring*:
automate verification for royalties/infringement claims
- *value-added services*:
offer information and meta data

audio fingerprinting

fingerprinting vs. watermarking

■ fingerprinting:

- identifies *recording* (but not musical content)

■ watermarking:

- embeds perceptually “unnoticeable” data block in the audio
- identifies *instance* of recording

Property	Fingerprinting	Watermarking
Allows Legacy Content Indexing	+	–
Allows Embedded (Meta) Data	–	+
Leaves Signal Unchanged	+	–
Identification of	Recording	User or Interaction

audio fingerprinting

fingerprinting vs. watermarking

■ fingerprinting:

- identifies *recording* (but not musical content)

■ watermarking:

- embeds perceptually “unnoticeable” data block in the audio
- identifies *instance* of recording

Property	Fingerprinting	Watermarking
Allows Legacy Content Indexing	+	–
Allows Embedded (Meta) Data	–	+
Leaves Signal Unchanged	+	–
Identification of	Recording	User or Interaction

audio fingerprinting

fingerprint requirements

- **accuracy & reliability:**
minimize false negatives/positives
- **robustness & security:**
robust against distortions and attacks
- **granularity:**
quick identification in a real-time context
- **versatility:**
independent of file format, etc.
- **scalability:**
good database performance
- **complexity:**
implementation possible on embedded devices

audio fingerprinting

fingerprint requirements

- **accuracy & reliability:**
minimize false negatives/positives
- **robustness & security:**
robust against distortions and attacks
- **granularity:**
quick identification in a real-time context
- **versatility:**
independent of file format, etc.
- **scalability:**
good database performance
- **complexity:**
implementation possible on embedded devices

audio fingerprinting

fingerprint requirements

- **accuracy & reliability:**
minimize false negatives/positives
- **robustness & security:**
robust against distortions and attacks
- **granularity:**
quick identification in a real-time context
- **versatility:**
independent of file format, etc.
- **scalability:**
good database performance
- **complexity:**
implementation possible on embedded devices

audio fingerprinting

fingerprint requirements

- **accuracy & reliability:**
minimize false negatives/positives
- **robustness & security:**
robust against distortions and attacks
- **granularity:**
quick identification in a real-time context
- **versatility:**
independent of file format, etc.
- **scalability:**
good database performance
- **complexity:**
implementation possible on embedded devices

audio fingerprinting

fingerprint requirements

- **accuracy & reliability:**
minimize false negatives/positives
- **robustness & security:**
robust against distortions and attacks
- **granularity:**
quick identification in a real-time context
- **versatility:**
independent of file format, etc.
- **scalability:**
good database performance
- **complexity:**
implementation possible on embedded devices

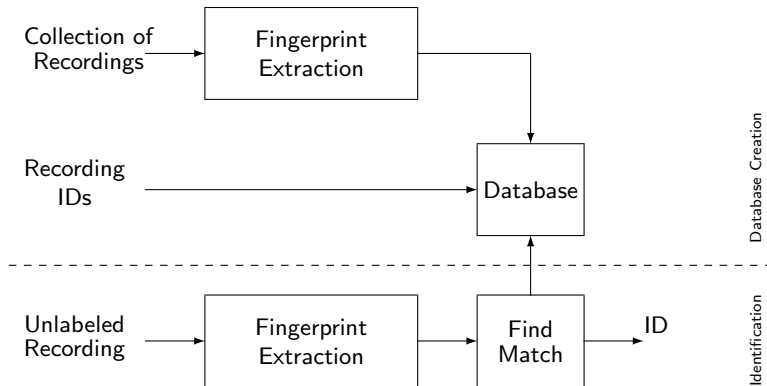
audio fingerprinting

fingerprint requirements

- **accuracy & reliability:**
minimize false negatives/positives
- **robustness & security:**
robust against distortions and attacks
- **granularity:**
quick identification in a real-time context
- **versatility:**
independent of file format, etc.
- **scalability:**
good database performance
- **complexity:**
implementation possible on embedded devices

audio fingerprinting

general fingerprinting system



audio fingerprinting

brainstorm

How does it work? MD5?



audio fingerprinting

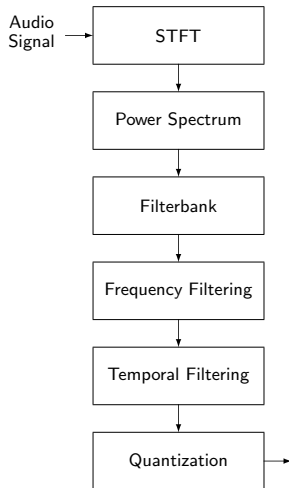
brainstorm

How does it work? MD5?



audio fingerprinting

system example: philips extraction 1/3



1 pre-processing:
downmixing & downsampling (5 kHz)

2 STFT: $\mathcal{K} = 2048$, overlap $\frac{31}{32}$

3 log frequency bands:
33 bands from 300–2000Hz

4 freq derivative: 33 bands

5 time derivative: 32 bands

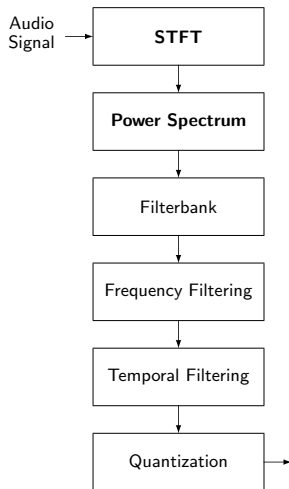
6 quantization:

$$v_{FP}(k, n) = \begin{cases} 1 & \text{if } (\Delta E(k, n) - \Delta E(k, n-1)) > 0 \\ 0 & \text{otherwise} \end{cases}$$

⇒ 32 bit *subfingerprint*

audio fingerprinting

system example: philips extraction 1/3



1 pre-processing:
downmixing & downsampling (5 kHz)

2 STFT: $\mathcal{K} = 2048$, overlap $\frac{31}{32}$

3 log frequency bands:
33 bands from 300–2000Hz

4 freq derivative: 33 bands

5 time derivative: 32 bands

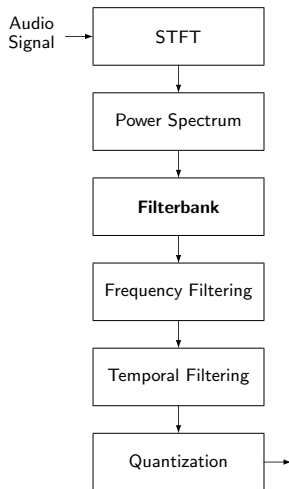
6 quantization:

$$v_{FP}(k, n) = \begin{cases} 1 & \text{if } (\Delta E(k, n) - \Delta E(k, n-1)) > 0 \\ 0 & \text{otherwise} \end{cases}$$

⇒ 32 bit *subfingerprint*

audio fingerprinting

system example: philips extraction 1/3



1 pre-processing:
downmixing & downsampling (5 kHz)

2 STFT: $\mathcal{K} = 2048$, overlap $\frac{31}{32}$

3 log frequency bands:
33 bands from 300–2000Hz

4 freq derivative: 33 bands

5 time derivative: 32 bands

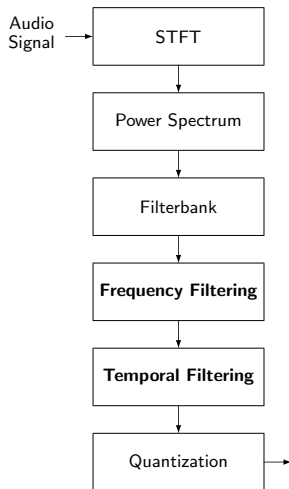
6 quantization:

$$v_{FP}(k, n) = \begin{cases} 1 & \text{if } (\Delta E(k, n) - \Delta E(k, n-1)) > 0 \\ 0 & \text{otherwise} \end{cases}$$

⇒ 32 bit *subfingerprint*

audio fingerprinting

system example: philips extraction 1/3



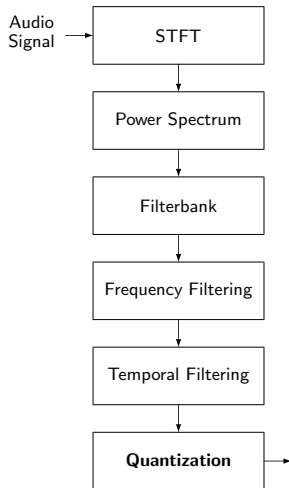
- 1 pre-processing:**
downmixing & downsampling (5 kHz)
- 2 STFT:** $\mathcal{K} = 2048$, overlap $\frac{31}{32}$
- 3 log frequency bands:**
33 bands from 300–2000Hz
- 4 freq derivative:** 33 bands
- 5 time derivative:** 32 bands
- 6 quantization:**

$$v_{FP}(k, n) = \begin{cases} 1 & \text{if } (\Delta E(k, n) - \Delta E(k, n-1)) > 0 \\ 0 & \text{otherwise} \end{cases}$$

⇒ 32 bit *subfingerprint*

audio fingerprinting

system example: philips extraction 1/3



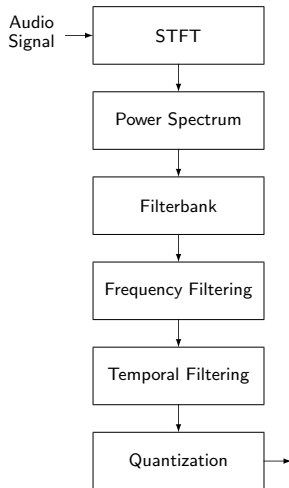
- 1 pre-processing:**
downmixing & downsampling (5 kHz)
- 2 STFT:** $\mathcal{K} = 2048$, overlap $\frac{31}{32}$
- 3 log frequency bands:**
33 bands from 300–2000Hz
- 4 freq derivative:** 33 bands
- 5 time derivative:** 32 bands
- 6 quantization:**

$$v_{FP}(k, n) = \begin{cases} 1 & \text{if } (\Delta E(k, n) - \Delta E(k, n-1)) > 0 \\ 0 & \text{otherwise} \end{cases}$$

⇒ 32 bit *subfingerprint*

audio fingerprinting

system example: philips extraction 1/3



- 1 pre-processing:**
downmixing & downsampling (5 kHz)
- 2 STFT:** $\mathcal{K} = 2048$, overlap $\frac{31}{32}$
- 3 log frequency bands:**
33 bands from 300–2000Hz
- 4 freq derivative:** 33 bands
- 5 time derivative:** 32 bands
- 6 quantization:**

$$v_{FP}(k, n) = \begin{cases} 1 & \text{if } (\Delta E(k, n) - \Delta E(k, n-1)) > 0 \\ 0 & \text{otherwise} \end{cases}$$

⇒ **32 bit subfingerprint**

audio fingerprinting

system example: philips extraction 2/3

fingerprint

- 256 subsequent subfingerprints

⇒

- *length*: 3 s
- *size*: $256 \cdot 4 \text{ Byte} = 1 \text{ kByte}$

example:

- 5 min song

$$1 \text{ kByte} \cdot \frac{5 \cdot 60 \text{ s}}{3 \text{ s}} = 100 \text{ kByte}$$

- database with 1 million songs (avg. length 5 min)

$$10^6 \cdot 256 \cdot \frac{5 \cdot 60 \text{ s}}{3 \text{ s}} = 25.6 \cdot 10^9 \text{ subfingerprints}$$

⇒ 100 GByte storage

audio fingerprinting

system example: philips extraction 2/3

fingerprint

- 256 subsequent subfingerprints

⇒

- *length*: 3 s
- *size*: $256 \cdot 4 \text{ Byte} = 1 \text{ kByte}$

example:

- 5 min song

$$1 \text{ kByte} \cdot \frac{5 \cdot 60 \text{ s}}{3 \text{ s}} = 100 \text{ kByte}$$

- database with 1 million songs (avg. length 5 min)

$$10^6 \cdot 256 \cdot \frac{5 \cdot 60 \text{ s}}{3 \text{ s}} = 25.6 \cdot 10^9 \text{ subfingerprints}$$

⇒ 100 GByte storage

audio fingerprinting

system example: philips extraction 2/3

fingerprint

- 256 subsequent subfingerprints

⇒

- *length*: 3 s
- *size*: $256 \cdot 4 \text{ Byte} = 1 \text{ kByte}$

example:

- 5 min song

$$1 \text{ kByte} \cdot \frac{5 \cdot 60 \text{ s}}{3 \text{ s}} = 100 \text{ kByte}$$

- database with 1 million songs (avg. length 5 min)

$$10^6 \cdot 256 \cdot \frac{5 \cdot 60 \text{ s}}{3 \text{ s}} = 25.6 \cdot 10^9 \text{ subfingerprints}$$

⇒ 100 GByte storage

audio fingerprinting

system example: philips extraction 2/3

fingerprint

- 256 subsequent subfingerprints

⇒

- *length*: 3 s
- *size*: $256 \cdot 4 \text{ Byte} = 1 \text{ kByte}$

example:

- 5 min song

$$1 \text{ kByte} \cdot \frac{5 \cdot 60 \text{ s}}{3 \text{ s}} = 100 \text{ kByte}$$

- database with 1 million songs (avg. length 5 min)


$$10^6 \cdot 256 \cdot \frac{5 \cdot 60 \text{ s}}{3 \text{ s}} = 25.6 \cdot 10^9 \text{ subfingerprints}$$

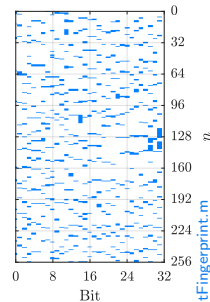
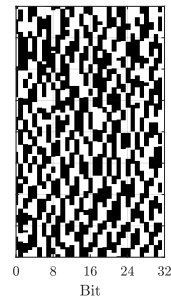
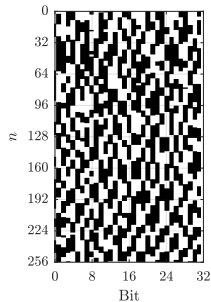
⇒ 100 GByte storage

audio fingerprinting

system example: philips extraction 3/3

■ original: 

■ low quality encoding: 



audio fingerprinting

system example: philips identification 1/3

■ database

- contains all subfingerprints for all songs
- previous example database: 25 billion subfingerprints

■ problem

- how to identify fingerprint efficiently?

audio fingerprinting

system example: philips identification 1/3

■ database

- contains all subfingerprints for all songs
- previous example database: 25 billion subfingerprints

■ problem

- how to identify fingerprint efficiently?

audio fingerprinting

system example: philips identification 1/3

■ database

- contains all subfingerprints for all songs
- previous example database: 25 billion subfingerprints

■ problem

- how to identify fingerprint efficiently?

audio fingerprinting

system example: philips identification 2/3

■ simple system:

- 1 create lookup table with all possible subfingerprints (2^{32}) pointing to occurrences
- 2 assume at least one of the extracted 256 subfingerprints is error-free
⇒ only entries listed at 256 positions of the table have to be checked
- 3 compute *Hamming* distance between extracted fingerprint and candidates

audio fingerprinting

system example: philips identification 2/3

■ simple system:

- 1 create lookup table with all possible subfingerprints (2^{32}) pointing to occurrences
- 2 assume at least one of the extracted 256 subfingerprints is error-free
⇒ only entries listed at 256 positions of the table have to be checked
- 3 compute *Hamming* distance between extracted fingerprint and candidates

audio fingerprinting

system example: philips identification 2/3

■ simple system:

- 1 create lookup table with all possible subfingerprints (2^{32}) pointing to occurrences
- 2 assume at least one of the extracted 256 subfingerprints is error-free
⇒ only entries listed at 256 positions of the table have to be checked
- 3 compute *Hamming* distance between extracted fingerprint and candidates

audio fingerprinting

system example: philips identification 3/3

■ variant 1:

- allow *one* bit error

⇒ workload increase by factor ≈ 33

■ variant 2:

- introduce concept of bit error probability into fingerprint extraction
 - ▶ small energy difference → high error probability
 - ▶ large energy difference → low error probability
- rank bits per subfingerprint by error probability and check only for bit errors at likely positions

audio fingerprinting

system example: philips identification 3/3

■ variant 1:

- allow *one* bit error
- ⇒ workload increase by factor ≈ 33

■ variant 2:

- introduce concept of bit error probability into fingerprint extraction
 - ▶ small energy difference → high error probability
 - ▶ large energy difference → low error probability
- rank bits per subfingerprint by error probability and check only for bit errors at likely positions

audio fingerprinting

system example: philips identification 3/3

■ variant 1:

- allow *one* bit error
- ⇒ workload increase by factor ≈ 33

■ variant 2:

- introduce concept of bit error probability into fingerprint extraction
 - ▶ small energy difference → high error probability
 - ▶ large energy difference → low error probability
- rank bits per subfingerprint by error probability and check only for bit errors at likely positions

audio fingerprinting

system example: philips identification 3/3

■ variant 1:

- allow *one* bit error
- ⇒ workload increase by factor ≈ 33

■ variant 2:

- introduce concept of bit error probability into fingerprint extraction
 - ▶ small energy difference → high error probability
 - ▶ large energy difference → low error probability
- rank bits per subfingerprint by error probability and check only for bit errors at likely positions

audio fingerprinting

other systems: shazam

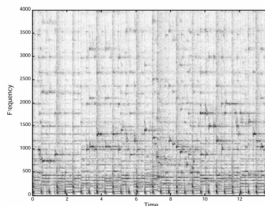


Fig. 1A - Spectrogram

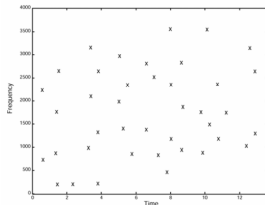


Fig. 1B - Constellation Map

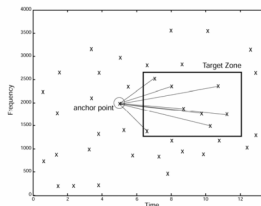


Fig. 1C - Combinatorial Hash Generation

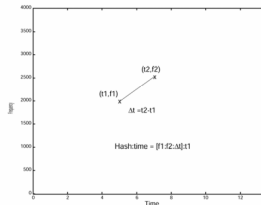


Fig. 1D - Hash details

plot from¹

¹A. Wang, "An Industrial Strength Audio Search Algorithm," in *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR)*, Washington, 2003.

summary

lecture content

■ audio fingerprinting

- represent recording with compact, robust, and unique fingerprint
- focus on (perceptual) audio representation rather than “musical” content
- allow efficient matching of this fingerprint with database

■ often confused with other tasks

- 1 audio watermarking
- 2 cover song detection

