



Introduction to **Audio Content Analysis**

module 3.2: audio pre-processing

alexander lerch

introduction

overview

corresponding textbook section

section 3.2

■ lecture content

- audio pre-processing for feature extraction

■ learning objectives

- list possible pre-processing options
- explain potential use cases



introduction

overview

corresponding textbook section

section 3.2

■ lecture content

- audio pre-processing for feature extraction

■ learning objectives

- list possible pre-processing options
- explain potential use cases



audio pre-processing

introduction

■ **pre-processing:** audio is treated before feature extraction (task dependent)

■ possible goals

- *reduce amount of data* (e.g., down-sampling)
- *remove irrelevant information* (e.g., surround channels of multi-channel signal)
- *remove information that might impact analysis* (e.g., DC offset)
- *remove redundant information*
- *make the signal invariant to irrelevant information*
- *increase robustness* (e.g., normalization)

Every pre-processing which improves the algorithm's accuracy or its robustness, or minimizes its complexity or computational workload is beneficial

⇒ improve accuracy/robustness and reduce complexity/workload

audio pre-processing

introduction

■ **pre-processing**: audio is treated before feature extraction (task dependent)

■ **possible goals**

- *reduce amount of data* (e.g., down-sampling)
- *remove irrelevant information* (e.g., surround channels of multi-channel signal)
- *remove information that might impact analysis* (e.g., DC offset)
- *remove redundant information*
- *make the signal invariant to irrelevant information*
- *increase robustness* (e.g., normalization)

Every pre-processing which improves the algorithm's accuracy or its robustness, or minimizes its complexity or computational workload is beneficial

⇒ improve accuracy/robustness and reduce complexity/workload

audio pre-processing

introduction

■ **pre-processing**: audio is treated before feature extraction (task dependent)

■ **possible goals**

- *reduce amount of data* (e.g., down-sampling)
- *remove irrelevant information* (e.g., surround channels of multi-channel signal)
- *remove information that might impact analysis* (e.g., DC offset)
- *remove redundant information*
- *make the signal invariant to irrelevant information*
- *increase robustness* (e.g., normalization)

Every pre-processing which improves the algorithm's accuracy or its robustness, or minimizes its complexity or computational workload is beneficial

⇒ improve accuracy/robustness and reduce complexity/workload

audio pre-processing

down-mixing

■ down-mixing

$$x(i) = \frac{1}{C} \sum_{c=0}^{C-1} x_c(i)$$

- *variants*: different channel weights, $\pi/2$ phase shift in one channel, ...

audio pre-processing

dc removal

■ dc removal

$$x(i) = x_{\text{DC}}(i) - \frac{1}{I} \sum_{i=0}^{I-1} x_{\text{DC}}(i)$$

■ *variants*: high pass, IIR

audio pre-processing

normalization

■ normalization

$$x(i) = \frac{x_s(i)}{\max_{\forall i} (|x_s(i)|)}$$

- *variants*: RMS, LUFS normalization
- real-time?

audio pre-processing

other pre-processing options

■ filtering

- low pass: remove noise, high frequencies
- high pass: remove rumble

■ sample rate conversion

- ensure all input files have identical sample rate
- bandwidth reduction

■ quality enhancement

- denoising
- bandwidth reduction

■ . . .

summary

lecture content

■ pre-processing goals

- remove irrelevant data
- clean relevant data
- reduce amount of data

■ pre-processing examples

- down-mixing
- dc removal
- normalization
- sample rate conversion

