Zürcher Hochschule
für Angewandte Wissenschaften
zh aw School of Engineering
InIT Institut für angewandte Informationstechnologie

Multimedia Analysis
Audio Segmentation

Material based on the lecture *„Video Retrieval“* by
Thilo Stadelmann, Ralph Ewerth, Bernd Freisleben
AG Verteilte Systeme, Fachbereich Mathematik & Informatik
Philipps Universität Marburg

Content

1. Introduction
   – Audio acquisition and representation
   – From signal to features
   – Audio segmentation

2. Audio type classification
   – The algorithm by Lu et al.

3. Speaker change detection
   – The algorithm by Kotti et al.
   – General Considerations

From video to soundtrack

- "Video" normally means: a stream of pictures (3D) *and* a sound stream (2D)


- ```
  ffmpeg -i input.mpg
         -vn -acodec pcm_s16le -ar 16000
         -ac 1 output.wav
  ```

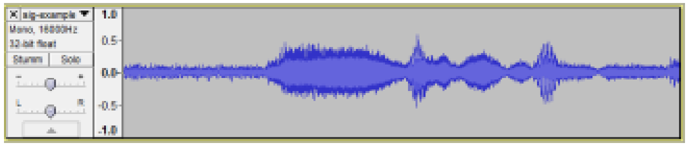
➔ pure audio signal (16 bit/sample, 16000 samples/second, mono)

➔ Technically: array of `short, s[n]`, n = 0..N-1
(N = videoLength in samples)

- More on audio representation: Camastra, Vinciarelli, "Machine Learning for Audio, Image and Video Analysis - Theory and Applications", 2008, Chapter 2
  ```

The audio signal

- `examples/sig-example.wav`

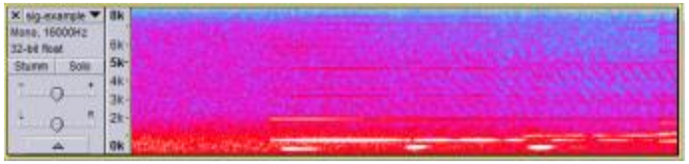


- *Time domain* information (2D):
  - energy
  - prominent frequency
    (for monophonic signals)

$$NRG = \frac{1}{N} \cdot \sum_n s[n]^2$$

$$ZCR = \frac{1}{N} \cdot \sum_n (s[n] \cdot s[n-1] < 0) ? 1 : 0$$

- *Frequency domain* information (3D):
  - time frequency representations via FFT or DWT,
  - discard phase



- More on signal processing: Smith, "Digital Signal Processing - A Practical Guide for Engineers and Scientists", 2003

Frame-based Processing (1)

- Feature extraction:
  - **Reduction** in **overall information**
  - while maintaining or even **emphasizing** the **useful information**

- Audio signal:
  - Neither stationary
    (➔problem with transformations like DFT when viewed as a whole)
  - nor conveys its meaning in single samples

$\Rightarrow$ **chop into** short, usually **overlapping chunks** called frames

$\Rightarrow$ extract **features per frame**

Frame-based Processing (2)

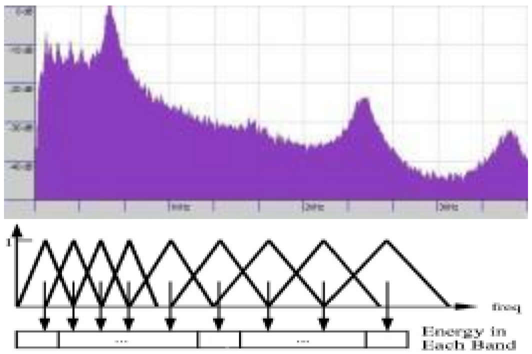
- Prominent parameters:
  - 16ms frame-step,
  - 32ms frame-size (50% overlap)

⇒ Technically: double-matrix `f[T][D]`, `T`=frame-count, `D`=feature-dimensionality

$$\Rightarrow \quad T = 1 + floor\left(\frac{ceil(sampleCount - frameSize)}{frameStep}\right)$$

Feature example: Mel Frequency Cepstral Coefficients

- MFCC: A compact representation of a frame's **smoothed spectral shape**
  - Preemphasize: `s[n] = s[n] - α*s[n-1]`
    (**boost high frequencies** to improve SNR; α close to 1)
  - Compute magnitude spectrum: `|FFT(s[n])|`
  - Accumulate under triangular Mel-scaled filter bank
    (resembles **human ear**)



(from developer.nokia.com & phys.unsw.edu.au/~jw)

  - Take DCT of filter bank output, discard all coefficients >M (for e.g. M=20)
    (i.e. low-pass → **compression**)

⇒ Low-pass filtered **spectrum of a spectrum**: "Cepstrum"

- MFCCs convey **most** of the **useful information** in a speech or music signal, **but no pitch** information

Content of audio signals

- The sample-array is 1D
- Nevertheless sound carries **information in** many **different** layers or "**dimensions**“
  - Silence ⬄ non-silence
  - Speech ⬄ music ⬄ noise
  - Voiced speech ⬄ unvoiced speech
  - Different musical genres, speakers, dialects, linguistical units, polyphony, emotions, . . .
- Segmentation: **temporally** separate one ore more of the above types from each other into **consecutive segments** by more or less specialized algorithms

Typical approaches to segmentation

- **Classification**
  - build models for each type a priori,
  - test which fits best for a given chunk of frames
- (Statistical) **change point detection**
  - Find changes in feature distribution parameters

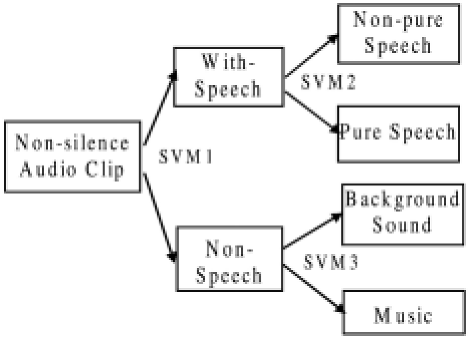
- **Local**
  - (sliding window based)
- **Global**
  - (genetic algorithms, Viterbi segmentation)

Algorithmic Overview

- **Audio type classification**:
  - discriminate between basic types
  - Prerequisite for any further audio analysis if ground truth is unavailable

- Example: Lu, Zhang, Li, *"Content-based Audio Classification and Segmentation by Using Support Vector Machines"*, 2003

- Taxonomy: **Sliding-window** based hierarchical **classification**:
  1. Silence ⇔ non-silence (via empirical threshold)
  2. Non-silence: speech ⇔ non-speech (via SVM)
  3. Speech: pure ⇔ non-pure
     Non-Speech: music ⇔ background (via SVMs)



Used features (1)

- Use **7+1 different fea**tures to cope with diverse signal properties
  - **NRG**
    (for silence detection alone, together with ZCR: both must be smaller than a threshold)
  - **ZCR**
  - 8 **MFCCs**

  - Sub band Power
    (**ratio of power** in each of **4 sub bands to overall** power)
  - Brightness and Bandwidth
    (**frequency centroid** and **spectral spread** width)

Used features (2)

- Spectrum Flux
  (average **spectral variation between** two successive **frames**)

- Band Periodicity
  (**periodicity** in 4 **sub bands**: $0-\frac{1}{8}, \frac{1}{8}-\frac{1}{4}, \frac{1}{4}-\frac{1}{2}, \frac{1}{2}-1$)

- Noise Frame Ratio
  (**ratio of noisy frames in a sub-clip**, i.e. frames with no prominent periodicity)

Feature construction

- **Sliding window** spans several frames (**1s**)
  → called a **sub-clip**

- What is a **representative** feature vector of such a sub-clip**?**
  - remember: a 1D array or a single row in a matrix

- **Aggregate frame-based features** per sub-clip:
  1. **Concatenate** (**columns** of) 7+1 different feature vectors to one big vector
  2. **Compute** mean **μ** and standard deviation **σ** of these vectors in each sub-clip

⇒ **Feature vector of** one **sub-clip**: **concatenated μ and σ** of each individual feature dimension

At runtime (1)

- Training the algorithm
  - (huge **annotated data** corpus needed, e.g. **30h**)
  - **Find** suitable **thresholds** on NRG and ZCR for **silence detection**
  - **Train SVMs** for each pair to discriminate between
- Training **runtime**: approximately **1 week**

At runtime (2)

- Using it („testing“ phase)
  - **preclassify** single frames as **silence**
  - for each **sub-clip** do . . .
    - extract and aggregate and normalize features
    - **classify** them **using SVM** tree
  - **smooth** the label series `l[i]`:

```
IF (l[i+1]!=l[i] AND
    l[i+2]!=l[i+1] AND
    l[i+1]!=SILENCE) THEN l[i+1]=l[i]
```

  - store result for all non-silence frames (silence stored before)

- Implementation effort: approximately 3 month

Experimental results

- **Accuracy** in [%] after smoothing

hypo/gt	Pure speech	Non-pure speech	Music	Background
Pure speech	90.53	8.3	0.26	0.91
Non-pure speech	0.0	96.2	2.28	1.52
Music	0.53	1.85	95.45	2.17
Background	1.66	6.65	4.07	87.62

  - Tested on **3 hours** mixed sample rate data from **TV**, **CD** and the **web**
  - The **smoothing** yielded **2-5%** additional performance

What is speaker change detection?

- Take a **speech-only** audio stream
  - i.e. do ATC and discard all non-speech frames
- Find all **change points**,
  - i.e. all samples spoken by a speaker different from the speaker of the previous sample
- Example: Kotti, Benetos, Kotropoulos, *"Computationally Efficient and Robust BIC-Based Speaker Segmentation"*, 2008
- Taxonomy: (adaptive) **sliding window** based **statistical** change point detection

The basic idea: BIC (1)

- Take a chunk of frames (Z) and **divide** it **into two chunks** X, Y
  - not necessarily half-way
- **Model** X, Y and Z each **with** a **single multivariate Gaussian**,
  - i.e estimate μ and Σ for each
- Compute **log likelihood** L of each **(sub-)chunk given its model**,
  - i.e. for a chunk A and its frames a_t:

$$L_A = |A| \cdot \frac{d}{2} \log(2\pi) \cdot \frac{1}{2} \log|\Sigma_A| \cdot \frac{1}{2} \sum_{t=1}^{|A|} (a_t - \mu_A)' \cdot \Sigma_A^{-1} \cdot (a_t - \mu_A)$$

The basic idea: BIC (2)

- Let a **model selection criterion** decide
  - two separate or **one single model** is **to prefer**?
  - Bayesian Information Criterion, **BIC**

$$BIC = L_X + L_Y - L_Z - \underbrace{\frac{\lambda}{2} \cdot \left(d + d \cdot \frac{d+1}{2}\right) \cdot \log\left|Z\right|}_{\text{penalty-term for more complex models}}$$

penalty-term for more complex models

- Decision:
  - cp. ⇔ BIC > 0,
  - **tune λ** for each data set

Design decisions

- What shall be the **size of** a **Z** chunk?
- **Where inside** a **Z** shall be the **splitting point**?
  - (i.e. hypothesized change point)
- What shall be the **window step size**?

- Solution:
  - Estimate **r**, the **mean** of **speaker turn length**
  - **Initial** chunk **size**: **2r**
  - **Grow** chunk **by r** if no change point found, otherwise reset to 2r
  - In each chunk, perform BIC checks (**split**) **at** each specific **submultiple of r**, e.g, r/3

What about features? (1)

- **MFCCs** are often applied to SCD problems,
  - but **dimensionality** and **parameters vary greatly**

- Idea:
  - **Fix** frame- and **DSP-parameters to** some **common standard**
  - Use upper bound of dimensionality and **find the best subset** comprising reasonable amount of dimensions (24 out of 36)
  - **Add δ and δδ** coefficients to the final subset

What about features? (2)

- Feature (**subset**) **selection**:
  - Create a training data set:
    - files containing one change point and
    - files containing no change point
  - Define a performance measure J
  - Find best 24-dimensional subset according to it
  - $\binom{36}{24} = 1.251.677.700$ 24-dimensional subsets possible

$\Rightarrow$ **need heuristic strategy**

Feature selection algorithm: details

- Use **depth-first search branch & bound** search strategy
  - (i.e. with backtracking)
  - Search tree has 36-24+1 = 13 levels

- Traverse the tree,
  - skip branches that have lower J then the so far seen best performance for the current level

$$J = tr(S_W^{-1} \cdot S_b)$$

  - S_w is **within class scatter**: deviation of sample vectors from their respective class means
  - S_b is **between class scatter**: deviation of sample vectors from the gross (overall, combined) mean

Experimental results

- Kotti et al. report on conTIMIT data:
  - **Precision** PRC=0.67
    - correctFoundChanges / hypothesizedChanges
  - **Recall** RCL=0.949
    - correctFoundChanges / actualChanges
  - F-Measure F_1=0.777
    - RCL*PRC / (RCL+PRC)
    - harmonic mean of RCL and PRC
  - False alarm rate FAR=0.289
    - falseAlarms / (actualChanges+falseAlarms)
  - Missed detections rate MDR=0.051
    - missedChanged / actualChanges

Literature survey result: what makes a good SCD algorithm? (1)

- Do **multi step analysis**, **reducing FAR** in each step
- **Use area surrounding** a **change point**, e.g. self-similarity-matrix for continuity-signal
  - (maybe as a last step?)
- Employ a method that **treats** the **stream holistically**
  - (e.g. Viterbi resegmentation, GA)
- Use **complementary features**, also **on different levels**
- **Fuse** different **classifiers** already **in each step**
- **Create multiple chances** for a change point to get detected

Literature survey result: what makes a good SCD algorithm? (2)

- **Model expected** segment **durations**
- Regression instead of classification learning?
- Use a Gauss window instead of a fixed sized window?
- **Move windows** with the **smallest possible increment**
- Use **1st order statistic** in 1st stage (more robust)
- Use **outer product matrix** to produce **equal size feature** vectors from differently sized segments
- Employ **AANN**s **on LPC residual** frames for short speaker turns

Review

- Audio Analysis is usually **frame-based**
- Basic **features** resemble the **smoothed magnitude spectrum**

- Audio segmentation works either by classification or (statistical) change point detection
- Local and holistic approaches exist

- **Audio segmentation works well**; **Speaker change detection not** yet ready