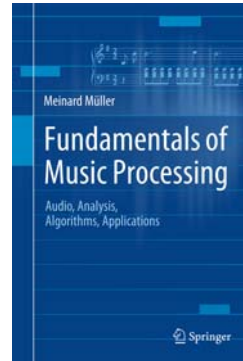


Lecture  
**Music Processing**

## Music Structure Analysis

**Meinard Müller**  
 International Audio Laboratories Erlangen  
 meinard.mueller@audiolabs-erlangen.de

## Book: Fundamentals of Music Processing



Meinard Müller  
 Fundamentals of Music Processing  
 Audio, Analysis, Algorithms, Applications  
 483 p., 249 illus., hardcover  
 ISBN: 978-3-319-21944-8  
 Springer, 2015

Accompanying website:  
[www.music-processing.de](http://www.music-processing.de)

## Book: Fundamentals of Music Processing

Chapter	Music Processing Scenario
1	Music Representations
2	Fourier Analysis of Signals
3	Music Synchronization
4	Music Structure Analysis
5	Chord Recognition
6	Tempo and Beat Tracking
7	Content-Based Audio Retrieval
8	Musically Informed Audio Decomposition

Meinard Müller  
 Fundamentals of Music Processing  
 Audio, Analysis, Algorithms, Applications  
 483 p., 249 illus., hardcover  
 ISBN: 978-3-319-21944-8  
 Springer, 2015

Accompanying website:  
[www.music-processing.de](http://www.music-processing.de)

## Book: Fundamentals of Music Processing

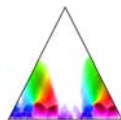
Chapter	Music Processing Scenario
1	Music Representations
2	Fourier Analysis of Signals
3	Music Synchronization
4	Music Structure Analysis
5	Chord Recognition
6	Tempo and Beat Tracking
7	Content-Based Audio Retrieval
8	Musically Informed Audio Decomposition

Meinard Müller  
 Fundamentals of Music Processing  
 Audio, Analysis, Algorithms, Applications  
 483 p., 249 illus., hardcover  
 ISBN: 978-3-319-21944-8  
 Springer, 2015

Accompanying website:  
[www.music-processing.de](http://www.music-processing.de)

## Chapter 4: Music Structure Analysis

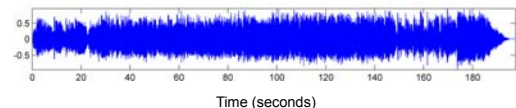
- 4.1 General Principles
- 4.2 Self-Similarity Matrices
- 4.3 Audio Thumbnailing
- 4.4 Novelty-Based Segmentation
- 4.5 Evaluation
- 4.6 Further Notes



In Chapter 4, we address a central and well-researched area within MIR known as music structure analysis. Given a music recording, the objective is to identify important structural elements and to temporally segment the recording according to these elements. Within this scenario, we discuss fundamental segmentation principles based on repetitions, homogeneity, and novelty—principles that also apply to other types of multimedia beyond music. As an important technical tool, we study in detail the concept of self-similarity matrices and discuss their structural properties. Finally, we briefly touch the topic of evaluation, introducing the notions of precision, recall, and F-measure.

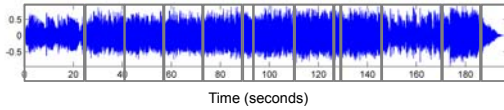
## Music Structure Analysis

**Example:** Zager & Evans “In The Year 2525”



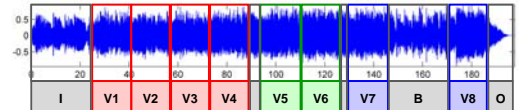
## Music Structure Analysis

**Example:** Zager & Evans "In The Year 2525"



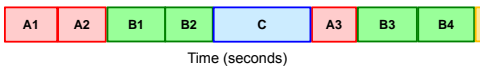
## Music Structure Analysis

**Example:** Zager & Evans "In The Year 2525"



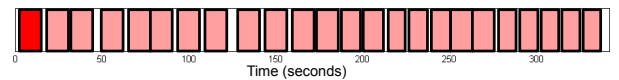
## Music Structure Analysis

**Example:** Brahms Hungarian Dance No. 5 (Ormandy)



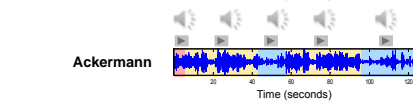
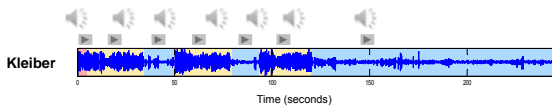
## Music Structure Analysis

**Example:** Folk Song Field Recording (Nederlandse Liederbank)



## Music Structure Analysis

**Example:** Weber, Song (No. 4) from "Der Freischütz"



## Music Structure Analysis

**General goal:** Divide an audio recording into temporal segments corresponding to musical parts and group these segments into musically meaningful categories.

**Examples:**

- Stanzas of a folk song
- Intro, verse, chorus, bridge, outro sections of a pop song
- Exposition, development, recapitulation, coda of a sonata
- Musical form ABACADA ... of a rondo

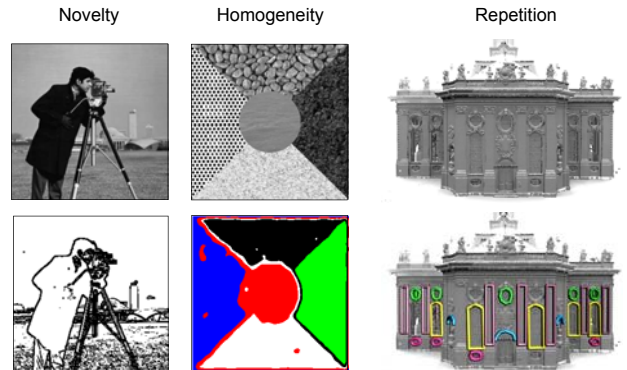
## Music Structure Analysis

**General goal:** Divide an audio recording into temporal segments corresponding to musical parts and group these segments into musically meaningful categories.

**Challenge:** There are many different principles for creating relationships that form the basis for the musical structure.

- **Homogeneity:** Consistency in tempo, instrumentation, key, ...
- **Novelty:** Sudden changes, surprising elements ...
- **Repetition:** Repeating themes, motives, rhythmic patterns,...

## Music Structure Analysis



## Overview

- **Introduction**
  - Feature Representations
  - Self-Similarity Matrices
  - Audio Thumbnailing
  - Novelty-based Segmentation
- Thanks:**
- Clausen, Ewert, Kurth, Grohgan, ...
  - Dannenberg, Goto
  - Grosche, Jiang
  - Paulus, Klapuri
  - Peeters, Kaiser, ...
  - Serra, Gómez, ...
  - Smith, Fujinaga, ...
  - Wiering, ...
  - Wand, Sunkel, Jansen
  - ...

## Overview

- Introduction
  - **Feature Representations**
  - Self-Similarity Matrices
  - Audio Thumbnailing
  - Novelty-based Segmentation
- Thanks:**
- Clausen, Ewert, Kurth, Grohgan, ...
  - Dannenberg, Goto
  - Grosche, Jiang
  - Paulus, Klapuri
  - Peeters, Kaiser, ...
  - Serra, Gómez, ...
  - Smith, Fujinaga, ...
  - Wiering, ...
  - Wand, Sunkel, Jansen
  - ...

## Feature Representation

**General goal:** Convert an audio recording into a mid-level representation that captures certain musical properties while suppressing other properties.

- Timbre / Instrumentation
- Tempo / Rhythm
- Pitch / Harmony

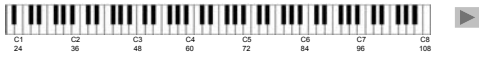
## Feature Representation

**General goal:** Convert an audio recording into a mid-level representation that captures certain musical properties while suppressing other properties.

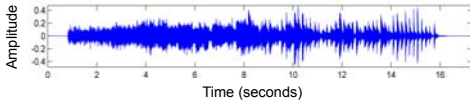
- Timbre / Instrumentation
- Tempo / Rhythm
- **Pitch / Harmony**

## Feature Representation

Example: Chromatic scale

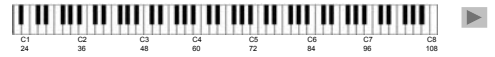


Waveform

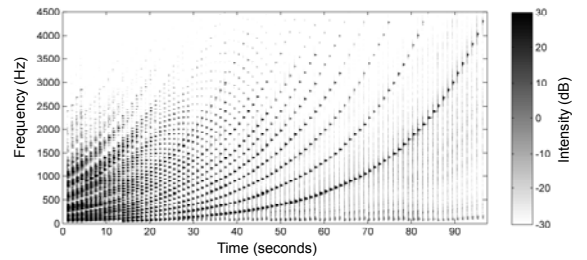


## Feature Representation

Example: Chromatic scale

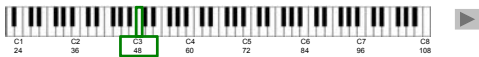


Spectrogram

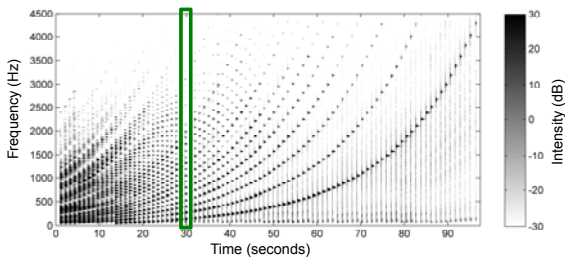


## Feature Representation

Example: Chromatic scale

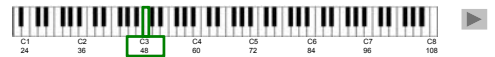


Spectrogram

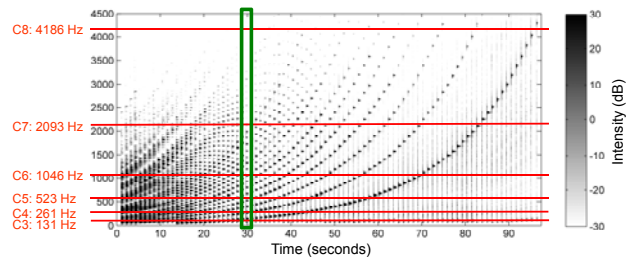


## Feature Representation

Example: Chromatic scale

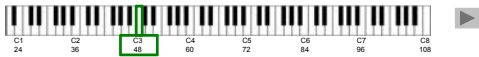


Spectrogram

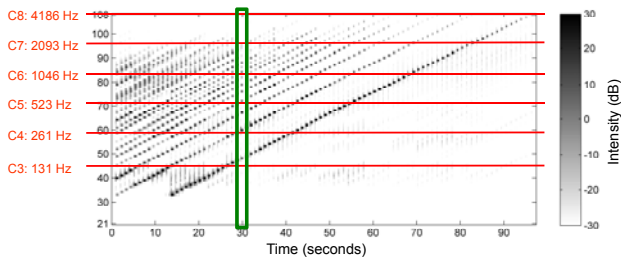


## Feature Representation

Example: Chromatic scale



Log-frequency spectrogram

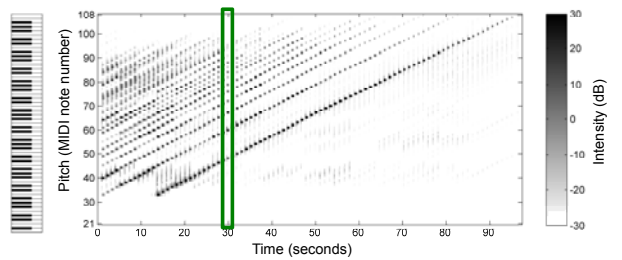


## Feature Representation

Example: Chromatic scale

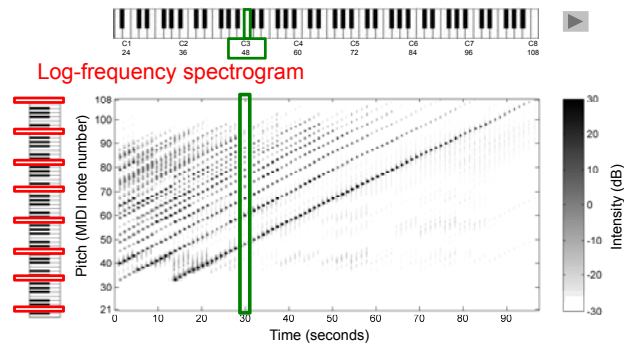


Log-frequency spectrogram



## Feature Representation

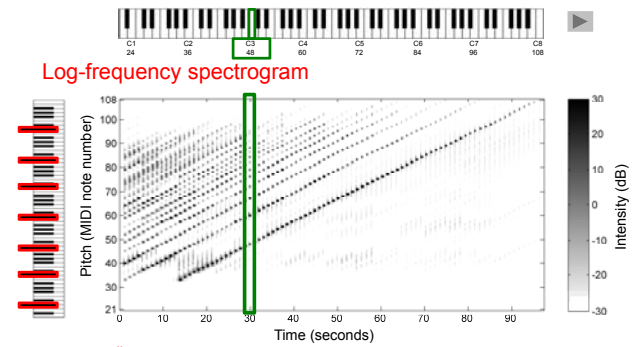
Example: Chromatic scale



Chroma C

## Feature Representation

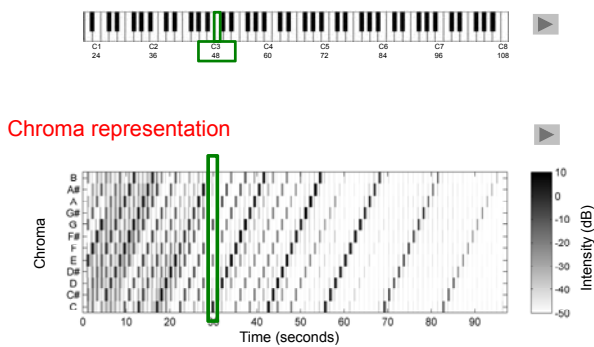
Example: Chromatic scale



Chroma C#

## Feature Representation

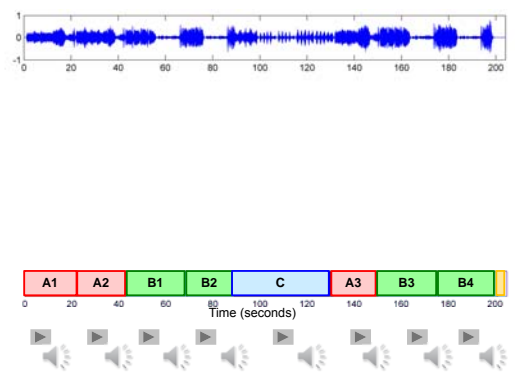
Example: Chromatic scale



Chroma representation

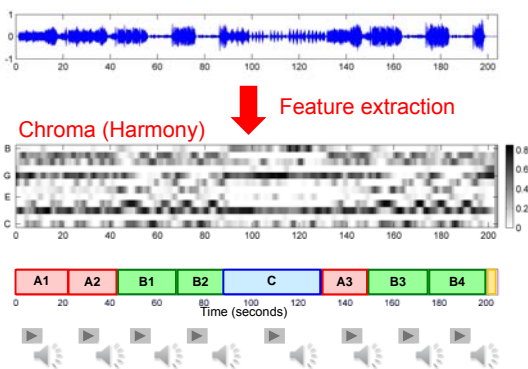
## Feature Representation

Example: Brahms Hungarian Dance No. 5 (Ormandy)



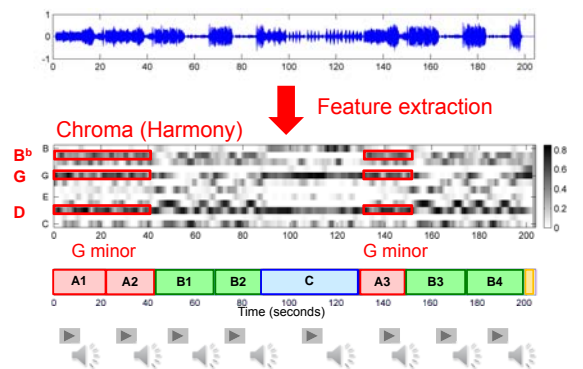
## Feature Representation

Example: Brahms Hungarian Dance No. 5 (Ormandy)



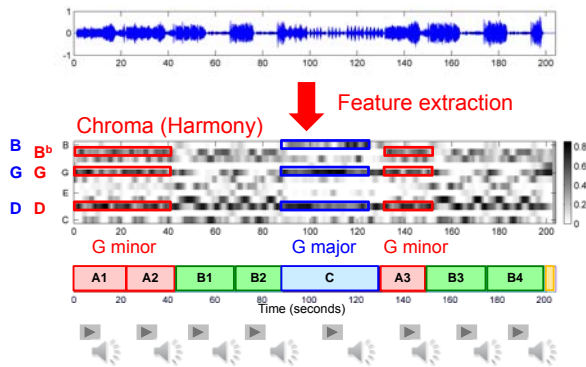
## Feature Representation

Example: Brahms Hungarian Dance No. 5 (Ormandy)



## Feature Representation

**Example:** Brahms Hungarian Dance No. 5 (Ormandy)



## Overview

- Introduction
- Feature Representations
- **Self-Similarity Matrices**
- Audio Thumbnailing
- Novelty-based Segmentation

## Self-Similarity Matrix (SSM)

**General idea:** Compare each element of the feature sequence with each other element of the feature sequence based on a suitable similarity measure.

→ Quadratic self-similarity matrix

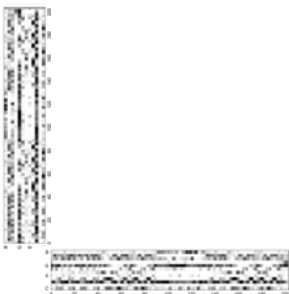
## Self-Similarity Matrix (SSM)

**Example:** Brahms Hungarian Dance No. 5 (Ormandy)



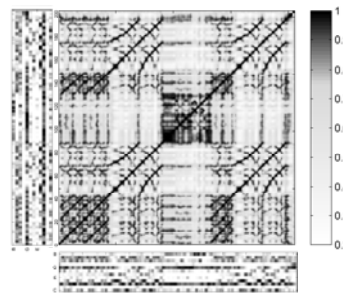
## Self-Similarity Matrix (SSM)

**Example:** Brahms Hungarian Dance No. 5 (Ormandy)



## Self-Similarity Matrix (SSM)

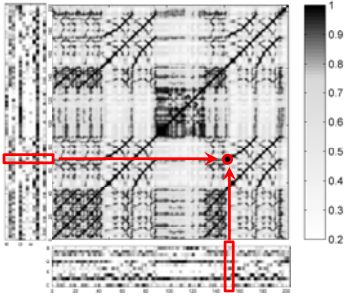
**Example:** Brahms Hungarian Dance No. 5 (Ormandy)





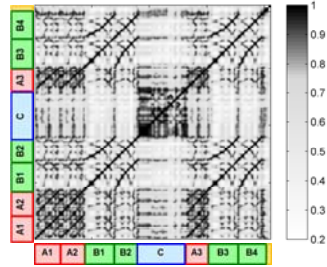
### Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



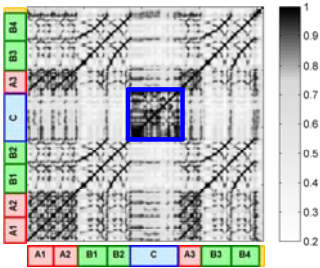
### Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



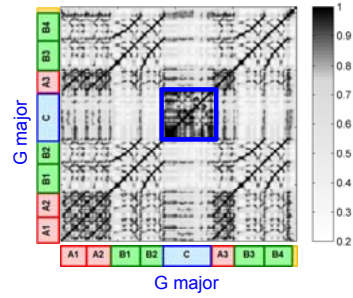
### Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



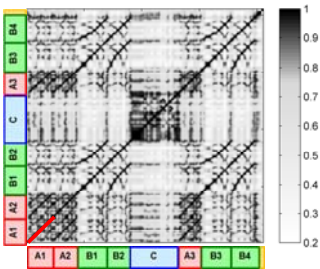
### Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



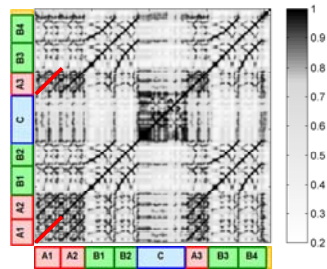
### Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



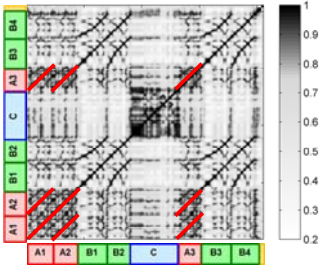
### Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



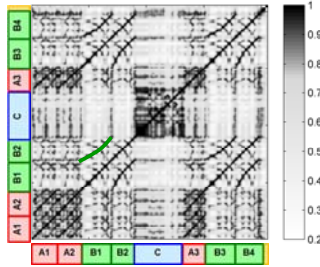
### Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



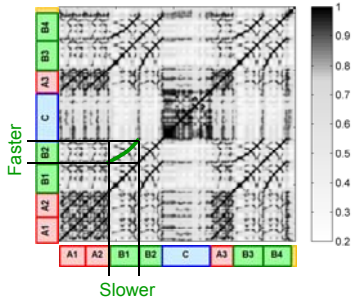
### Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



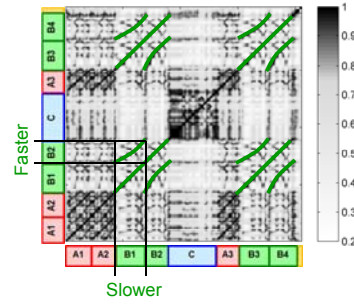
### Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



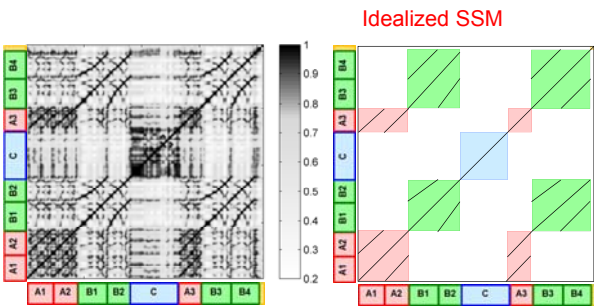
### Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



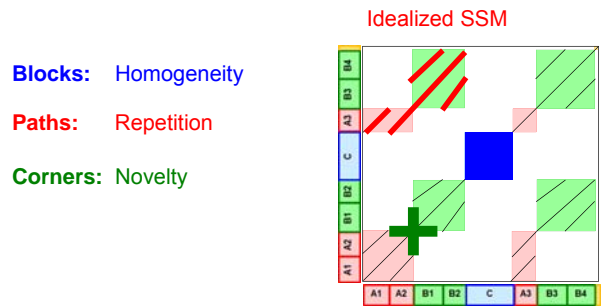
### Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



### Self-Similarity Matrix (SSM)

Example: Brahms Hungarian Dance No. 5 (Ormandy)



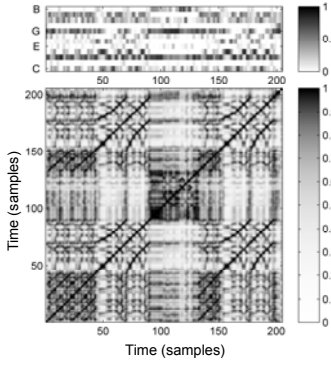
Blocks: Homogeneity

Paths: Repetition

Corners: Novelty



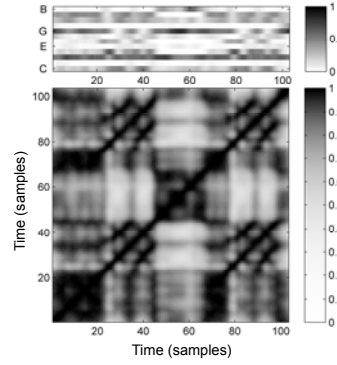
## SSM Enhancement



### Block Enhancement

- Feature smoothing
- Coarsening

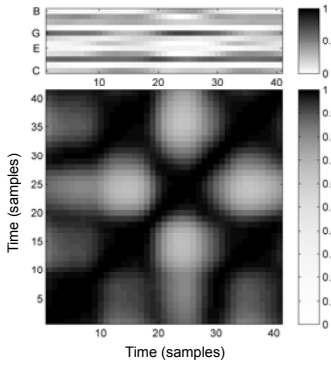
## SSM Enhancement



### Block Enhancement

- Feature smoothing
- Coarsening

## SSM Enhancement



### Block Enhancement

- Feature smoothing
- Coarsening

## SSM Enhancement

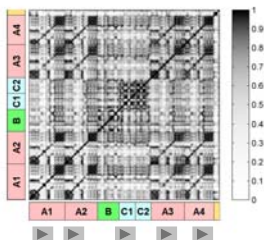
Challenge: Presence of musical variations

- Fragmented paths and gaps
- Paths of poor quality
- Regions of constant (high) similarity
- Curved paths

Idea: Enhancement of path structure

## SSM Enhancement

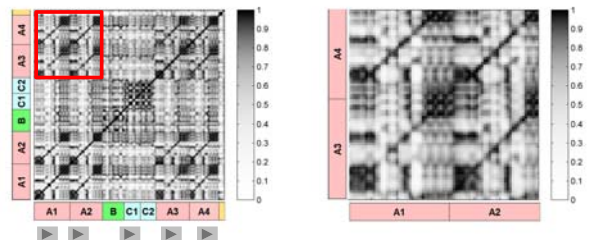
Shostakovich Waltz 2, Jazz Suite No. 2 (Chailly)



SSM

## SSM Enhancement

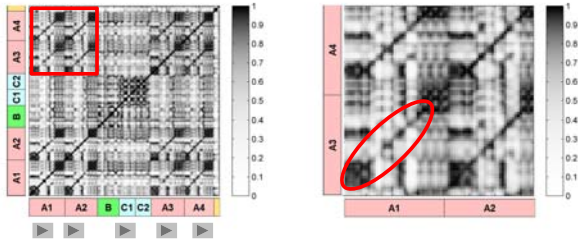
Shostakovich Waltz 2, Jazz Suite No. 2 (Chailly)



SSM

## SSM Enhancement

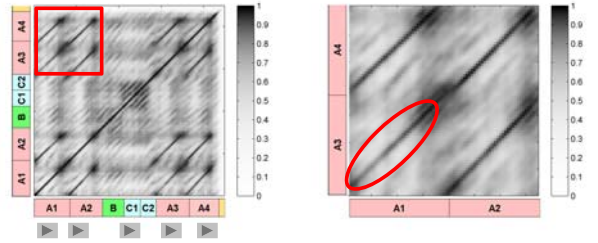
Shostakovich Waltz 2, Jazz Suite No. 2 (Chailly)



SSM

## SSM Enhancement

Shostakovich Waltz 2, Jazz Suite No. 2 (Chailly)



Enhanced SSM

Filtering along main diagonal

## SSM Enhancement

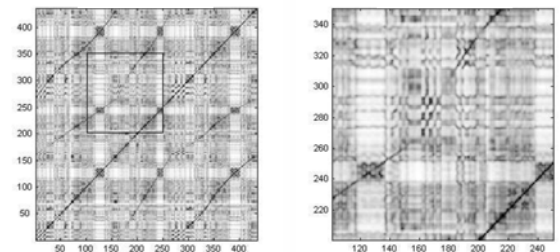
Idea: Usage of contextual information (Foote 1999)

$$\mathbf{S}_L(n, m) := \frac{1}{L} \sum_{\ell=0}^{L-1} \mathbf{S}(n + \ell, m + \ell)$$

- Comparison of entire sequences
- $L$  = length of sequences
- $\mathbf{S}_L$  = enhanced SSM

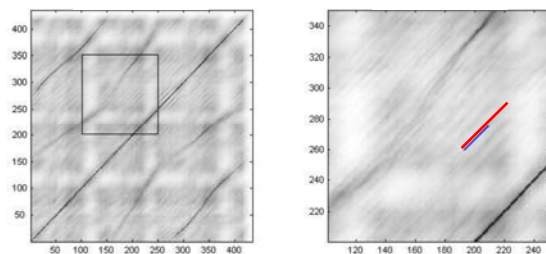
↪ smoothing effect

## SSM Enhancement



SSM  $\mathbf{S}$

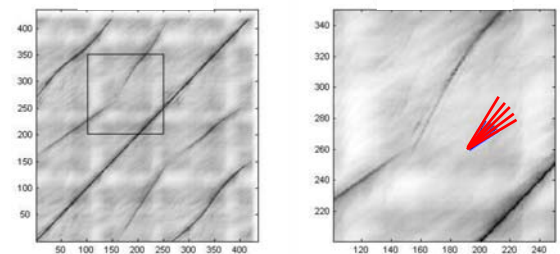
## SSM Enhancement



Enhanced SSM  $\mathbf{S}_L$  with  $L = 20$

Filtering along main diagonal

## SSM Enhancement



Enhanced SSM  $\mathbf{S}_{L,\Theta}$  with  $L = 20$

Filtering along 8 different directions and minimizing

## SSM Enhancement

Idea: Smoothing along various directions and minimizing over all directions

$$\Theta = \{0.66, 0.81, 1.00, 1.22, 1.50\}$$

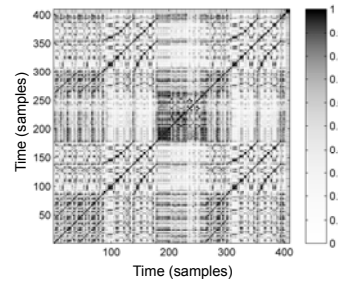
$$S_{L,\theta}(n,m) := \frac{1}{L} \sum_{\ell=0}^{L-1} S(n+\ell, m + [\ell \cdot \theta])$$

$$S_{L,\Theta}(n,m) := \max_{\theta \in \Theta} S_{L,\theta}(n,m)$$

↔ Tempo changes of -50 to +50 percent

## SSM Enhancement

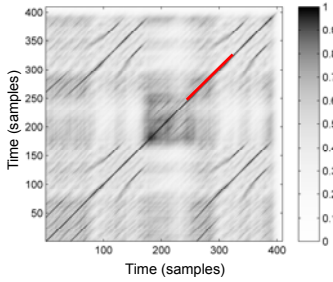
Path Enhancement



## SSM Enhancement

Path Enhancement

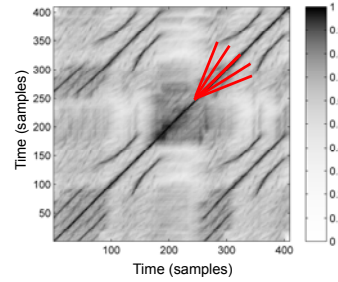
- Diagonal smoothing



## SSM Enhancement

Path Enhancement

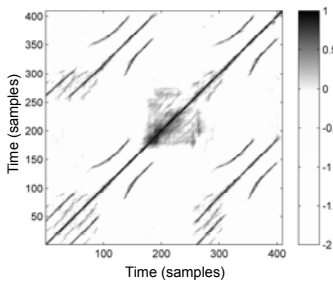
- Diagonal smoothing
- Multiple filtering



## SSM Enhancement

Path Enhancement

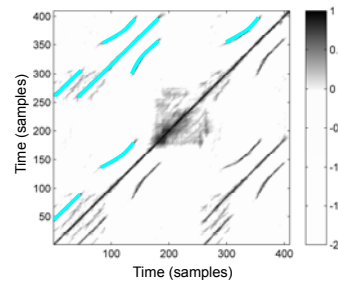
- Diagonal smoothing
- Multiple filtering
- Thresholding (relative)
- Scaling & penalty



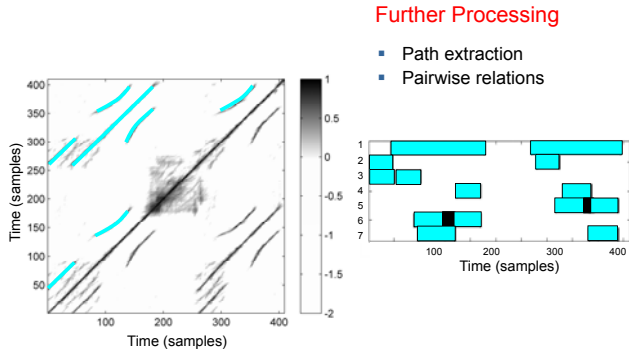
## SSM Enhancement

Further Processing

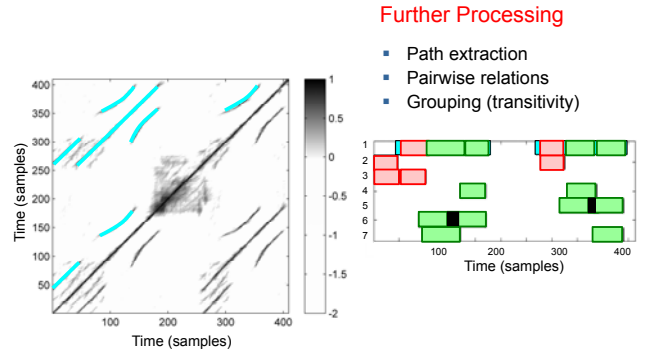
- Path extraction



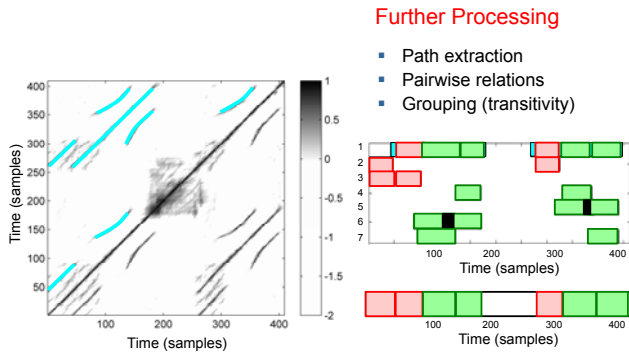
## SSM Enhancement



## SSM Enhancement

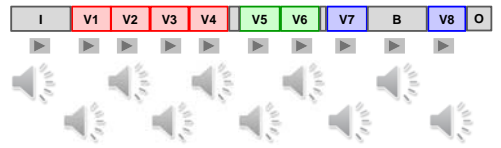


## SSM Enhancement



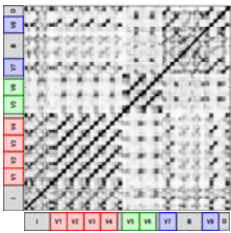
## SSM Enhancement

**Example:** Zager & Evans "In The Year 2525"



## SSM Enhancement

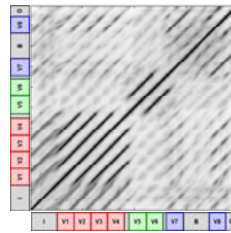
**Example:** Zager & Evans "In The Year 2525"



## SSM Enhancement

**Example:** Zager & Evans "In The Year 2525"

Missing relations because of transposed sections

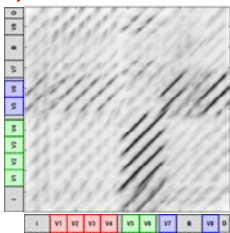


## SSM Enhancement

**Example:** Zager & Evans "In The Year 2525"

Idea: Cyclic shift of one of the chroma sequences

One semitone up

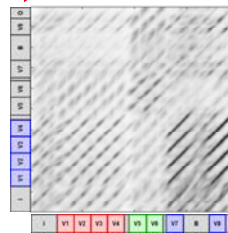


## SSM Enhancement

**Example:** Zager & Evans "In The Year 2525"

Idea: Cyclic shift of one of the chroma sequences

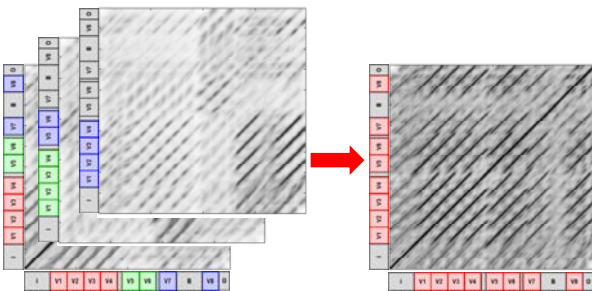
Two semitones up



## SSM Enhancement

**Example:** Zager & Evans "In The Year 2525"

Idea: Overlay & Maximize → Transposition-invariant SSM

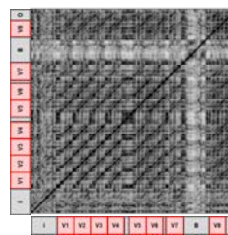


## SSM Enhancement

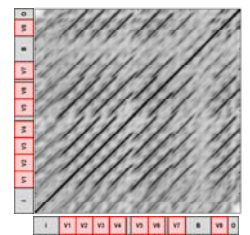
**Example:** Zager & Evans "In The Year 2525"

Note: Order of enhancement steps important!

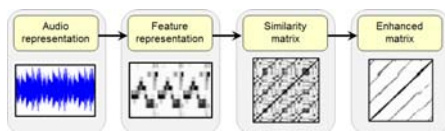
Maximization



Smoothing & Maximization



## Similarity Matrix Toolbox



Meinard Müller, Nanzhu Jiang, Harald Grohganz  
SM Toolbox: MATLAB Implementations for Computing and  
Enhancing Similarity Matrices

<http://www.audiolabs-erlangen.de/resources/MIR/SMtoolbox/>

## Overview

- Introduction
- Feature Representations
- Self-Similarity Matrices
- **Audio Thumbnailing**
- Novelty-based Segmentation

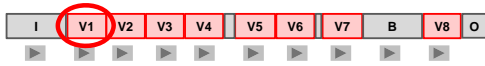
**Thanks:**

- Jiang, Grosche
- Peeters
- Cooper, Foote
- Goto
- Levy, Sandler
- Mauch
- Sapp

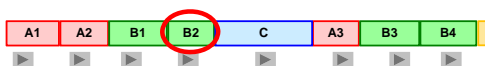
## Audio Thumbnailing

**General goal:** Determine the most representative section ("Thumbnail") of a given music recording.

**Example:** Zager & Evans "In The Year 2525"



**Example:** Brahms Hungarian Dance No. 5 (Ormandy)



Thumbnail is often assumed to be the most repetitive segment

## Audio Thumbnailing

**Two steps**

1. Path extraction

**Both steps are problematic!**

- Paths of poor quality (fragmented, gaps)
- Block-like structures
- Curved paths
- Noisy relations (missing, distorted, overlapping)
- Transitivity computation difficult

2. Grouping

**Main idea:** Do both, path extraction and grouping, jointly

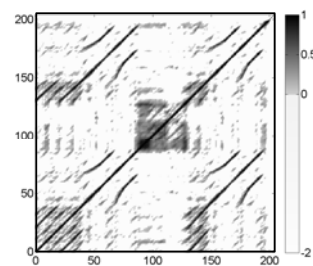
- One optimization scheme for both steps
- Stabilizing effect
- Efficient

## Audio Thumbnailing

Main idea: Do both path extraction and grouping jointly

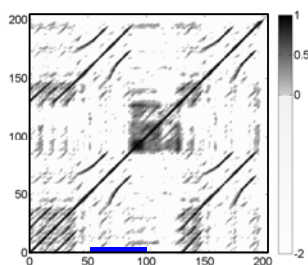
- For each audio **segment** we define a **fitness** value
- This fitness value expresses "how well" the segment explains the entire audio recording
- The segment with the highest fitness value is considered to be the **thumbnail**
- As main technical concept we introduce the notion of a **path family**

## Fitness Measure



Enhanced SSM

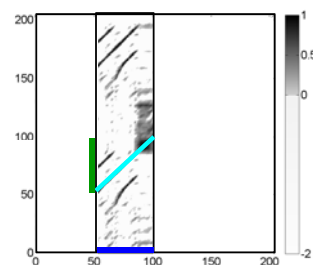
## Fitness Measure



Path over segment

- Consider a fixed **segment**

## Fitness Measure

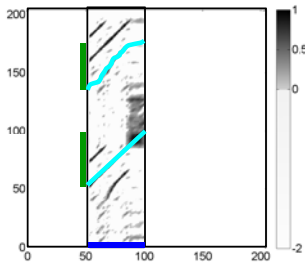


Path over segment

- Consider a fixed **segment**
- **Path** over **segment**
- **Induced segment**
- Score is high



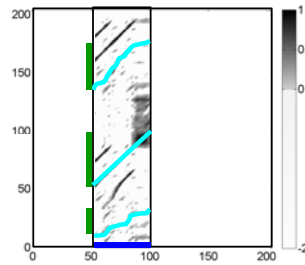
## Fitness Measure



### Path over segment

- Consider a fixed **segment**
- Path** over **segment**
- Induced segment**
- Score is high
- A second path** over **segment**
- Induced segment**
- Score is not so high

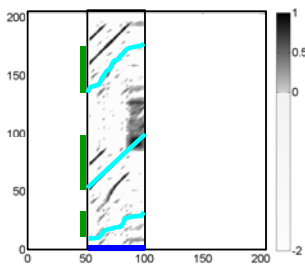
## Fitness Measure



### Path over segment

- Consider a fixed **segment**
- Path** over **segment**
- Induced segment**
- Score is high
- A second path** over **segment**
- Induced segment**
- Score is not so high
- A third path** over **segment**
- Induced segment**
- Score is very low

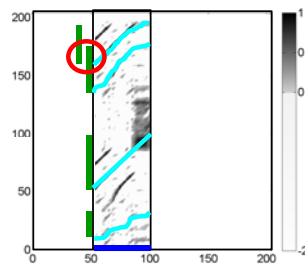
## Fitness Measure



### Path family

- Consider a fixed **segment**
- A path family over a **segment** is a family of paths such that the **induced segments** do **not overlap**.

## Fitness Measure

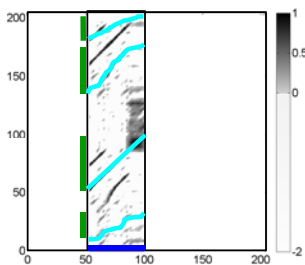


### Path family

- Consider a fixed **segment**
- A path family over a **segment** is a family of paths such that the **induced segments** do **not overlap**.

This is **not** a path family!

## Fitness Measure



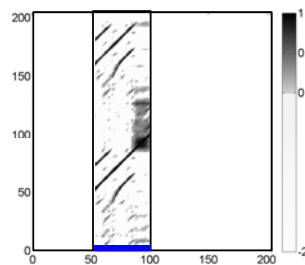
### Path family

- Consider a fixed **segment**
- A path family over a **segment** is a family of paths such that the **induced segments** do **not overlap**.

This is a path family!

(Even though not a good one)

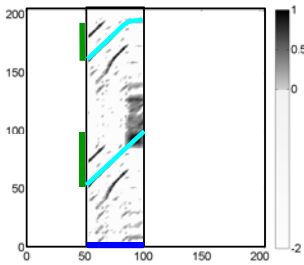
## Fitness Measure



### Optimal path family

- Consider a fixed **segment**

## Fitness Measure

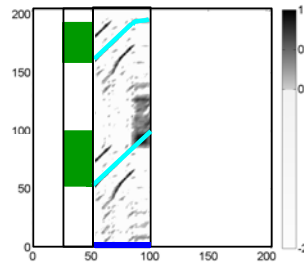


### Optimal path family

- Consider a fixed **segment**
- Consider over the **segment** the **optimal path family**, i.e., the path family having maximal overall score.
- Call this value:  
 $\text{Score}(\text{segment})$

Note: This optimal path family can be computed using dynamic programming.

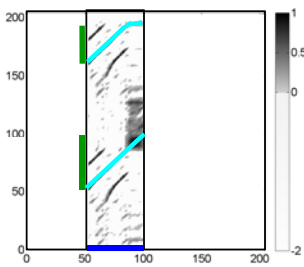
## Fitness Measure



### Optimal path family

- Consider a fixed **segment**
- Consider over the **segment** the **optimal path family**, i.e., the path family having maximal overall score.
- Call this value:  
 $\text{Score}(\text{segment})$
- Furthermore consider the amount covered by the **induced segments**.
- Call this value:  
 $\text{Coverage}(\text{segment})$

## Fitness Measure

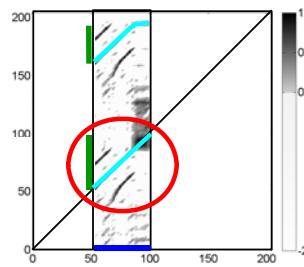


### Fitness

- Consider a fixed **segment**

$P := \text{Score}(\text{segment})$   
 $R := \text{Coverage}(\text{segment})$

## Fitness Measure

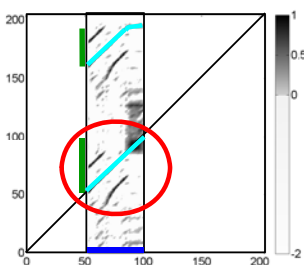


### Fitness

- Consider a fixed **segment**
- Self-explanation are trivial!**

$P := \text{Score}(\text{segment})$   
 $R := \text{Coverage}(\text{segment})$

## Fitness Measure

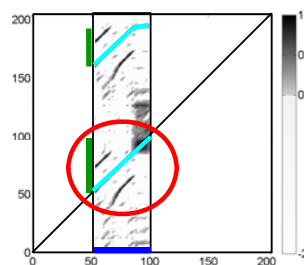


### Fitness

- Consider a fixed **segment**
- Self-explanation are trivial!**
- Subtract length of **segment**

$P := \text{Score}(\text{segment}) - \text{length}(\text{segment})$   
 $R := \text{Coverage}(\text{segment}) - \text{length}(\text{segment})$

## Fitness Measure

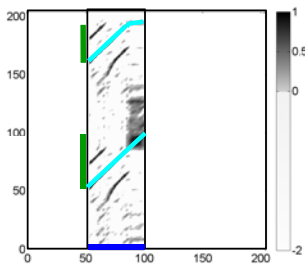


### Fitness

- Consider a fixed **segment**
- Self-explanation are trivial!**
- Subtract length of **segment**
- Normalization

$P := \text{Normalize}(\text{Score}(\text{segment}) - \text{length}(\text{segment})) \in [0,1]$   
 $R := \text{Normalize}(\text{Coverage}(\text{segment}) - \text{length}(\text{segment})) \in [0,1]$

## Fitness Measure



Fitness

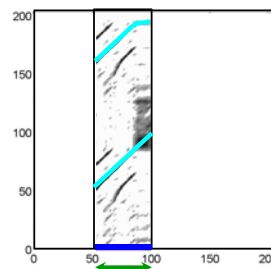
- Consider a fixed segment

$$F := 2 \cdot P \cdot R / (P + R)$$

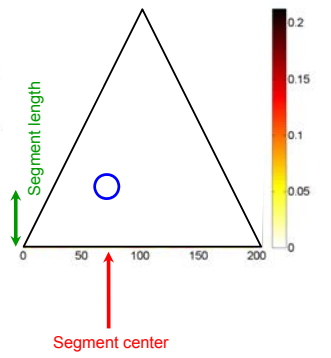
$$P := \text{Normalize}(\text{Score}(\text{segment}) - \text{length}(\text{segment})) \in [0,1]$$

$$R := \text{Normalize}(\text{Coverage}(\text{segment}) - \text{length}(\text{segment})) \in [0,1]$$

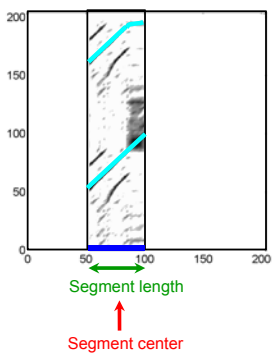
## Thumbnail



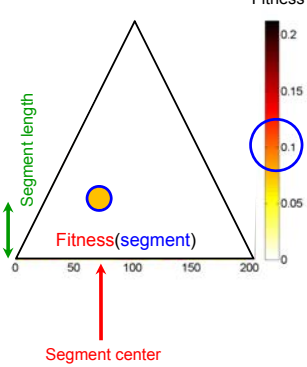
## Fitness Scape Plot



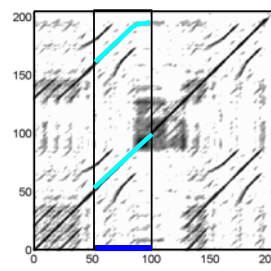
## Thumbnail



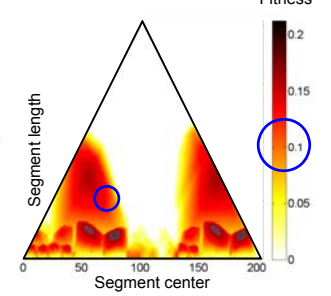
## Fitness Scape Plot



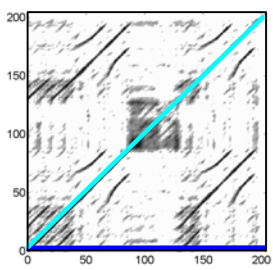
## Thumbnail



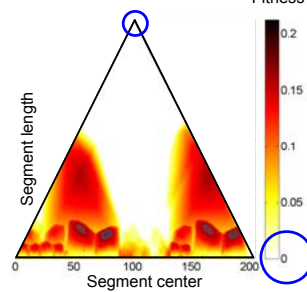
## Fitness Scape Plot



## Thumbnail

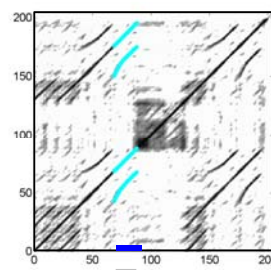


## Fitness Scape Plot

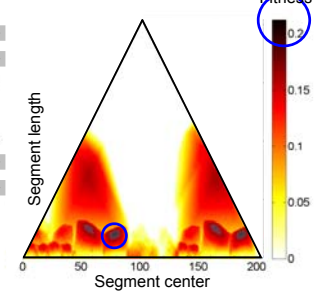


Note: Self-explanations are ignored → fitness is zero

## Thumbnail

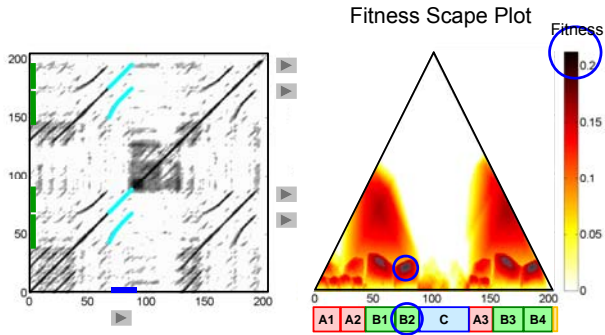


## Fitness Scape Plot



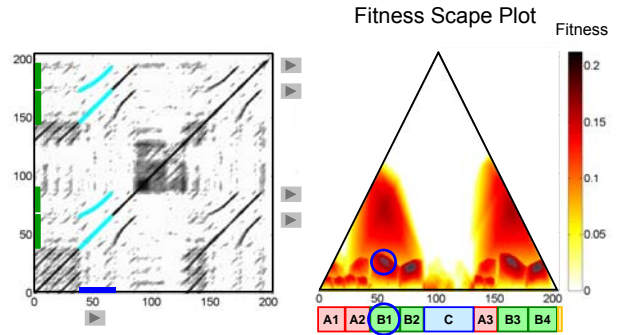
Thumbnail := segment having the highest fitness

### Thumbnail



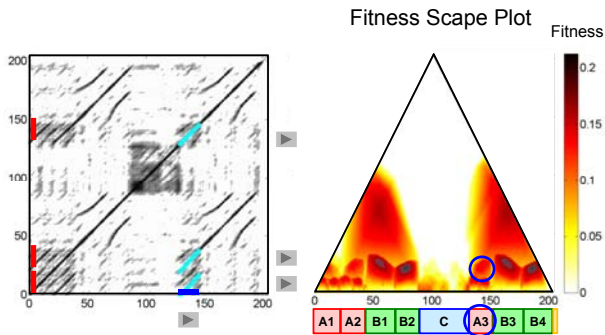
Example: Brahms Hungarian Dance No. 5 (Ormandy)

### Thumbnail



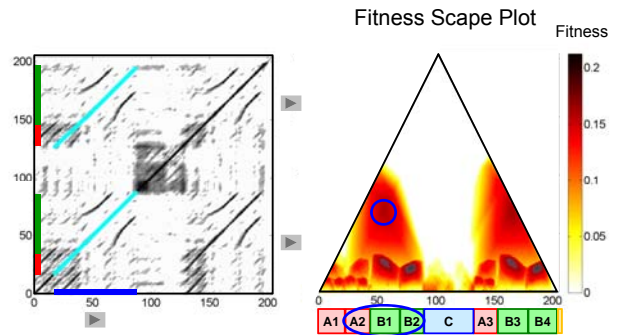
Example: Brahms Hungarian Dance No. 5 (Ormandy)

### Thumbnail



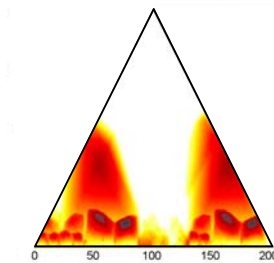
Example: Brahms Hungarian Dance No. 5 (Ormandy)

### Thumbnail



Example: Brahms Hungarian Dance No. 5 (Ormandy)

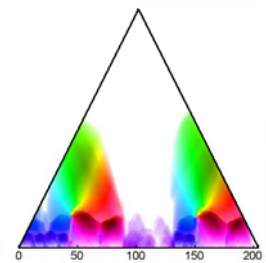
### Scape Plot



Example: Brahms Hungarian Dance No. 5 (Ormandy)

### Scape Plot

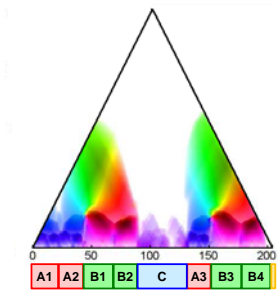
Coloring according to clustering result (grouping)



Example: Brahms Hungarian Dance No. 5 (Ormandy)

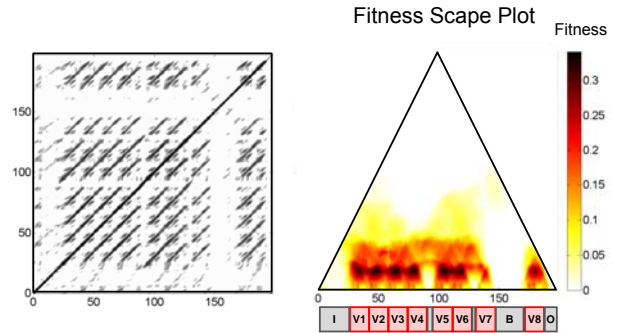
## Scape Plot

Coloring according to clustering result (grouping)



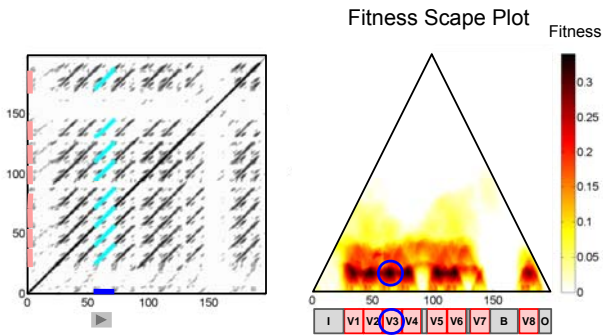
**Example:** Brahms Hungarian Dance No. 5 (Ormandy)

## Thumbnail



**Example:** Zager & Evans "In The Year 2525"

## Thumbnail



**Example:** Zager & Evans "In The Year 2525"

## Overview

- Introduction
- Feature Representations
- Self-Similarity Matrices
- Audio Thumbing
- Novelty-based Segmentation

**Thanks:**

- Foote
- Serra, Grosche, Arcos
- Goto
- Tzanetakis, Cook

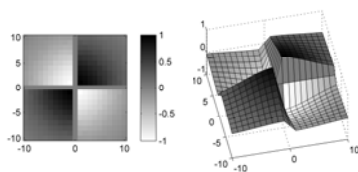
## Novelty-based Segmentation

### General goals:

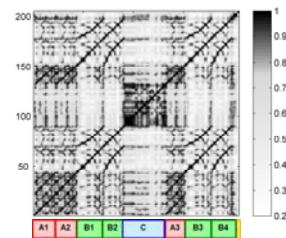
- Find instances where musical changes occur.
- Find transition between subsequent musical parts.

### Idea (Foote):

Use checkerboard-like kernel function to detect corner points on main diagonal of SSM.



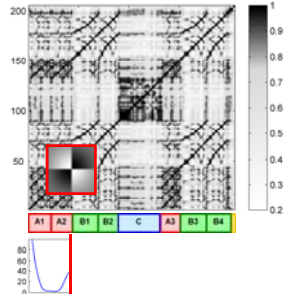
## Novelty-based Segmentation



### Idea (Foote):

Use checkerboard-like kernel function to detect corner points on main diagonal of SSM.

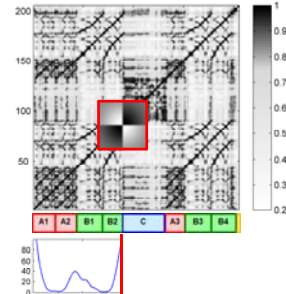
## Novelty-based Segmentation



### Idea (Foote):

Use checkerboard-like kernel function to detect corner points on main diagonal of SSM.

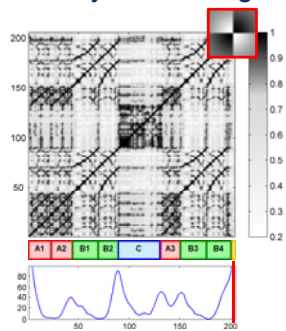
## Novelty-based Segmentation



### Idea (Foote):

Use checkerboard-like kernel function to detect corner points on main diagonal of SSM.

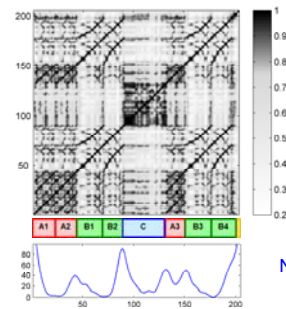
## Novelty-based Segmentation



### Idea (Foote):

Use checkerboard-like kernel function to detect corner points on main diagonal of SSM.

## Novelty-based Segmentation



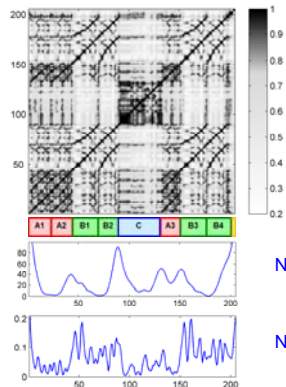
### Idea (Foote):

Use checkerboard-like kernel function to detect corner points on main diagonal of SSM.

Novelty function using



## Novelty-based Segmentation



### Idea (Foote):

Use checkerboard-like kernel function to detect corner points on main diagonal of SSM.

Novelty function using



Novelty function using



## Novelty-based Segmentation

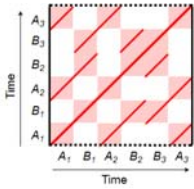
### Idea:

- Find instances where **structural** changes occur.
- Combine **global** and **local** aspects within a unifying framework

### Structure features



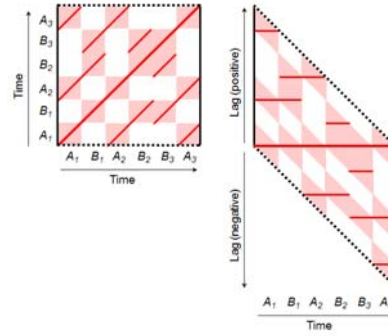
## Novelty-based Segmentation



### Structure features

- Enhanced SSM

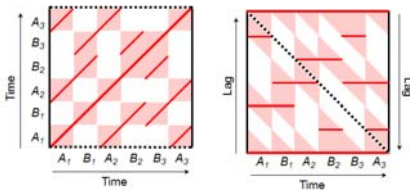
## Novelty-based Segmentation



### Structure features

- Enhanced SSM
- Time-lag SSM

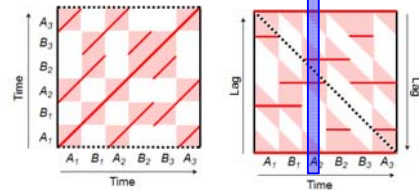
## Novelty-based Segmentation



### Structure features

- Enhanced SSM
- Time-lag SSM
- Cyclic time-lag SSM

## Novelty-based Segmentation

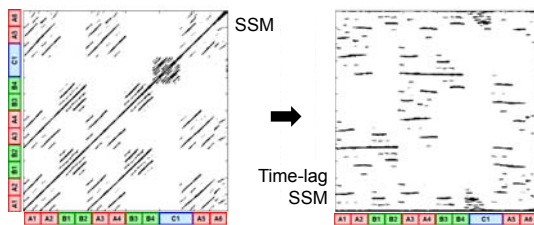


### Structure features

- Enhanced SSM
- Time-lag SSM
- Cyclic time-lag SSM
- Columns as features

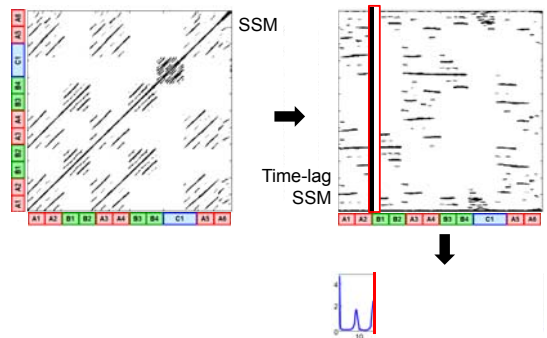
## Novelty-based Segmentation

**Example:** Chopin Mazurka Op. 24, No. 1



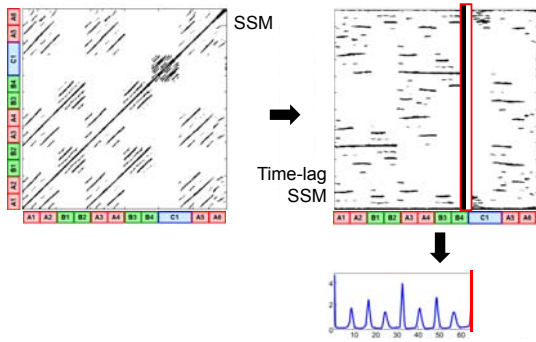
## Novelty-based Segmentation

**Example:** Chopin Mazurka Op. 24, No. 1



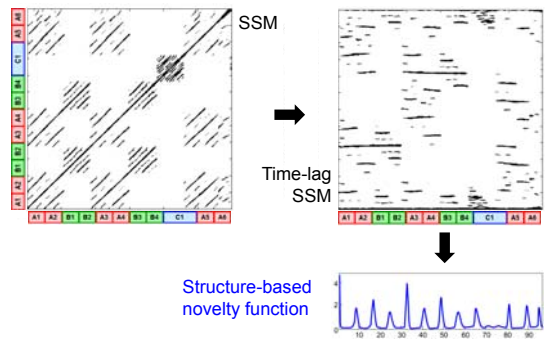
## Novelty-based Segmentation

Example: Chopin Mazurka Op. 24, No. 1



## Novelty-based Segmentation

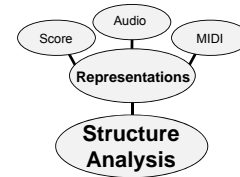
Example: Chopin Mazurka Op. 24, No. 1



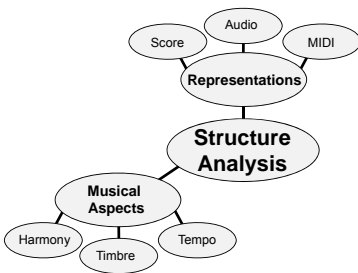
## Conclusions

Structure Analysis

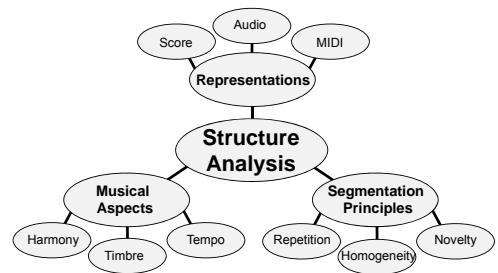
## Conclusions



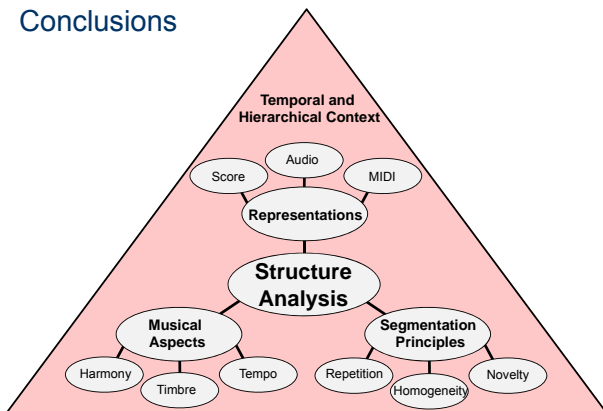
## Conclusions



## Conclusions

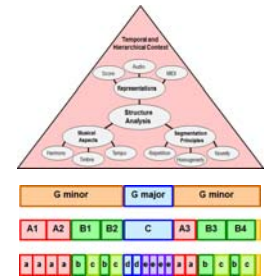


## Conclusions



## Conclusions

- Combined Approaches
- Hierarchical Approaches
- Evaluation
- Explaining Structure



- MIREX
- SALAMI-Project
- Smith, Chew

## Links

- SM Toolbox (MATLAB)  
<http://www.audiolabs-erlangen.de/resources/MIR/SMtoolbox/>
- MSAF: Music Structure Analysis Framework (Python)  
<https://github.com/urinieto/msaf>
- SALAMI Annotation Data  
<http://ddmal.music.mcgill.ca/research/salami/annotations>
- LibROSA (Python)  
<https://librosa.github.io/librosa/>
- Evaluation: mir\_eval (Python)  
[https://craffel.github.io/mir\\_eval/](https://craffel.github.io/mir_eval/)
- Deep Learning: Boundary Detection  
Jan Schlüter (PhD thesis)