

INTERNATIONAL AUDIO LABORATORIES ERLANGEN
A joint institution of Fraunhofer IIS and Universität Erlangen-Nürnberg



An Introduction to Music Information Retrieval

Meinard Müller

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

Guest Lecture
Central Conservatory of Music (CCOM)
Beijing, March 2023

Meinard Müller



- Mathematics (Diplom/Master, 1997)
Computer Science (PhD, 2001)
Information Retrieval (Habilitation, 2007)
- Senior Researcher (2007-2012)
- Professor Semantic Audio Processing (since 2012)
- Former President of the International Society for Music Information Retrieval (ISMIR)
- IEEE Fellow for contributions to Music Signal Processing



Meinard Müller: Research Group

Semantic Audio Processing

- Michael Krause
- Yigitcan Özer
- Simon Schwär
- Johannes Zeitler
- Peter Meier (external)
- Christof Weiß
- Sebastian Rosenzweig
- Frank Zalkow
- Christian Dittmar
- Stefan Balke
- Jonathan Driedger
- Thomas Prätzlich
- ...



International Audio Laboratories Erlangen



- Fraunhofer Institute for Integrated Circuits IIS
- Largest Fraunhofer institute with ≈ 1000 members
- Applied research for sensor, audio, and media technology



- Friedrich-Alexander Universität Erlangen-Nürnberg (FAU)
- One of Germany's largest universities with $\approx 40,000$ students
- Strong Technical Faculty

International Audio Laboratories Erlangen



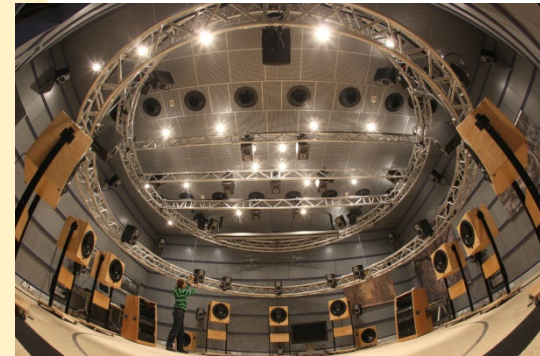
Audio

International Audio Laboratories Erlangen

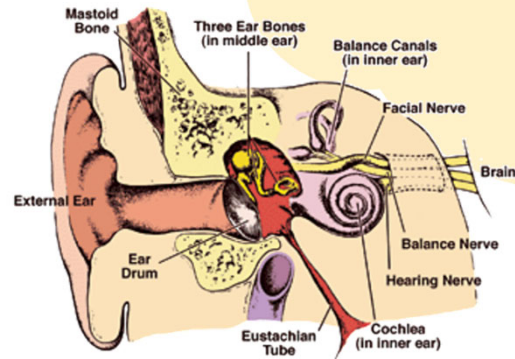
Audio Coding



3D Audio



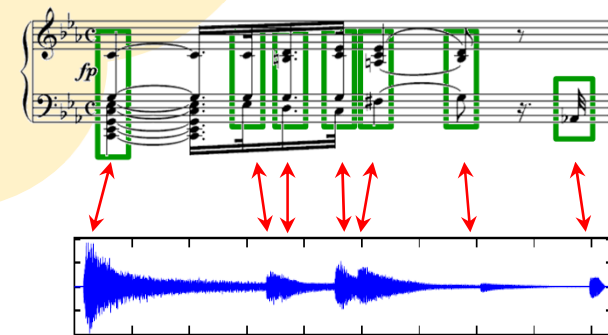
Audio



Psychoacoustics



Internet of Things



Music Processing

AudioLabs – FAU

- Prof. Dr. Jürgen Herre
Audio Coding
- Prof. Dr. Bernd Edler
Audio Signal Analysis
- Prof. Dr. Meinard Müller
Semantic Audio Processing
- Prof. Dr. Emanuël Habets
Spatial Audio Signal Processing
- Prof. Dr. Nils Peters
Audio Signal Processing
- Dr. Stefan Turowski
Coordinator AudioLabs-FAU

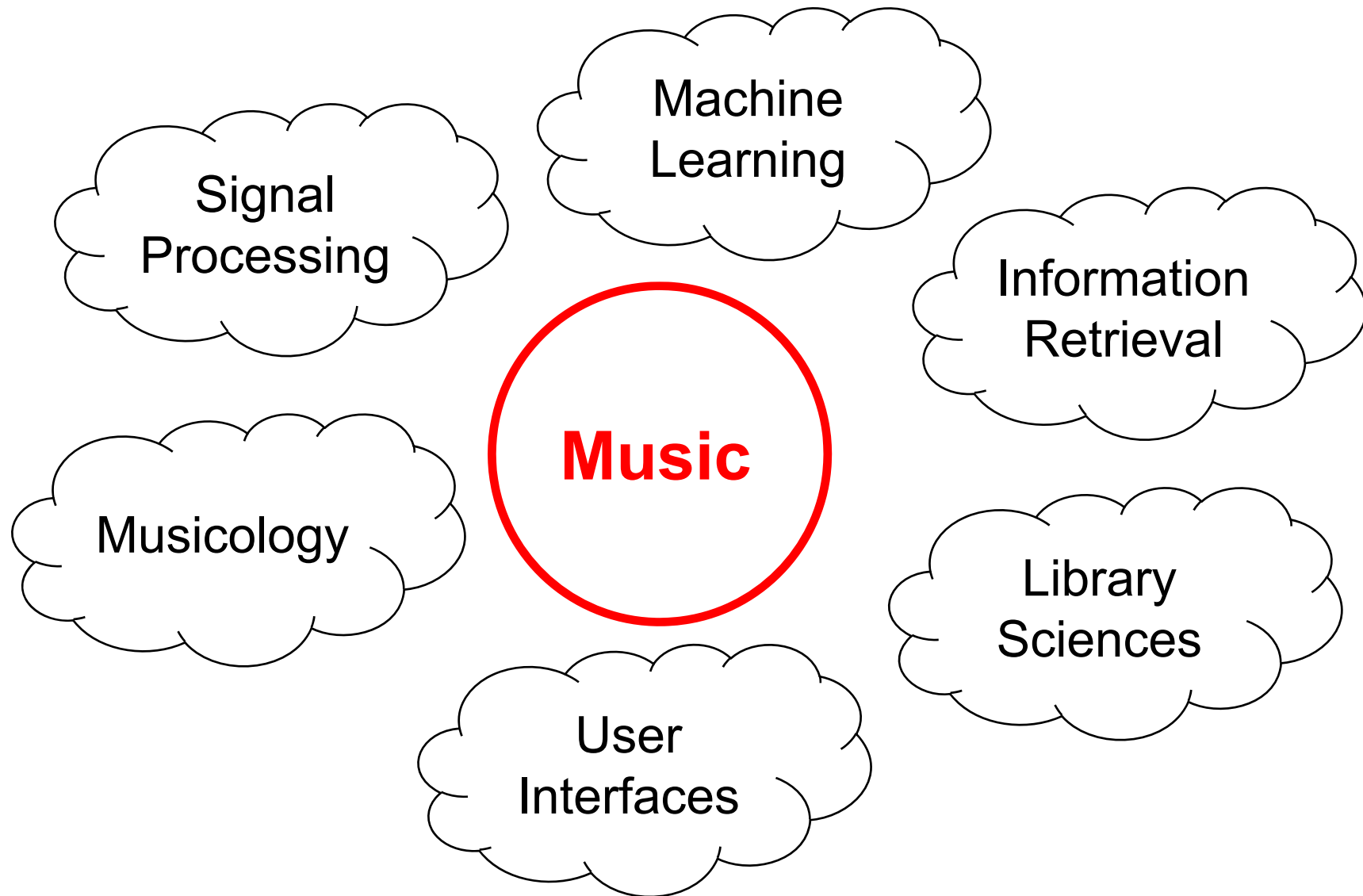




Music



Music Information Retrieval (MIR)

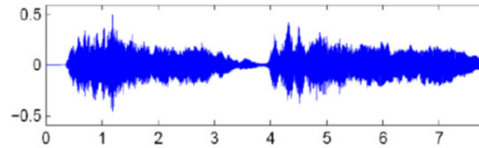


Music Information Retrieval (MIR)

Sheet Music (Image)



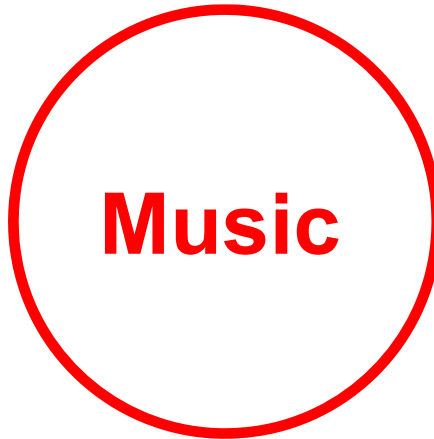
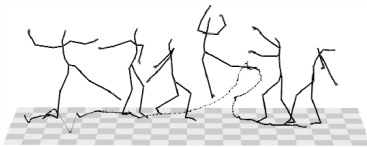
CD / MP3 (Audio)



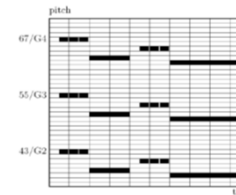
MusicXML (Text)

```
<note>  
  <pitch>  
    <step>E</step>  
    <alter>-1</alter>  
    <octave>4</octave>  
  </pitch>  
  <duration>2</duration>  
  <type>half</type>  
</note>
```

Dance / Motion (Mocap)



MIDI



Singing / Voice (Audio)



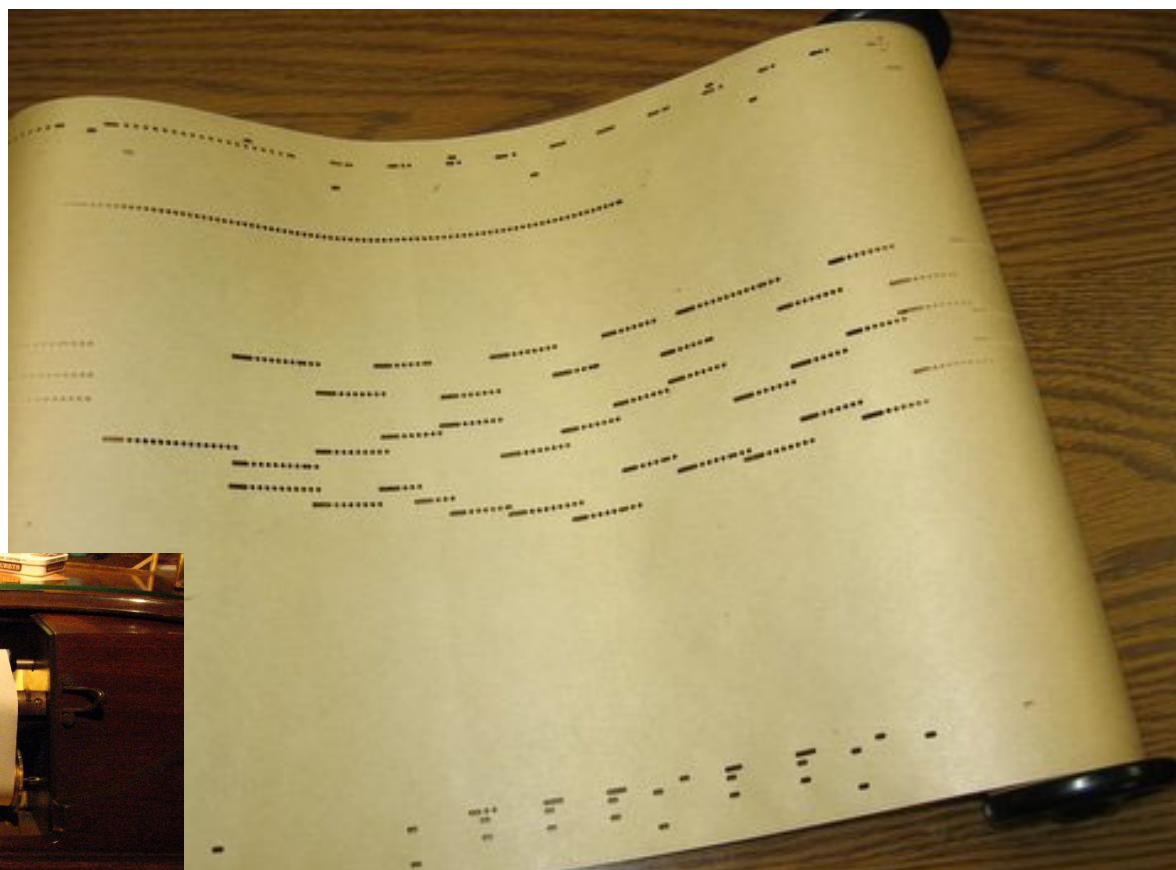
Music Film (Video)



Music Literature (Text)



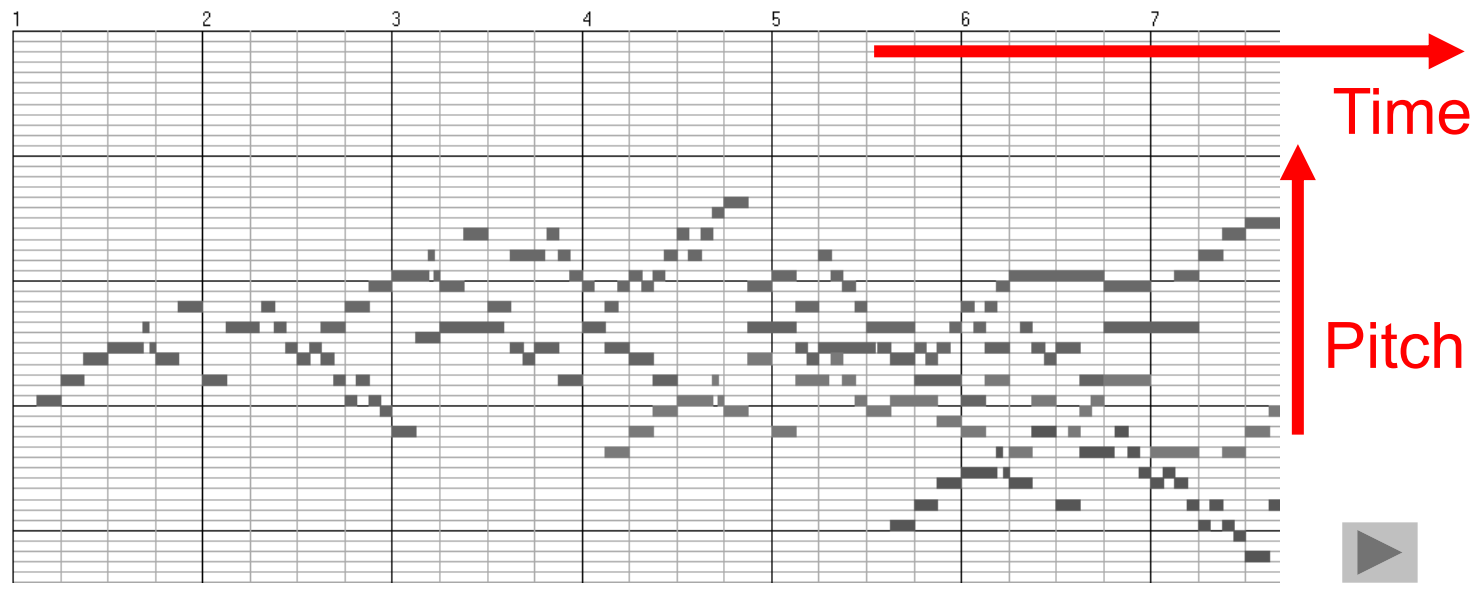
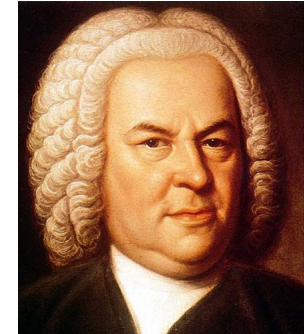
Piano Roll Representation (1900)



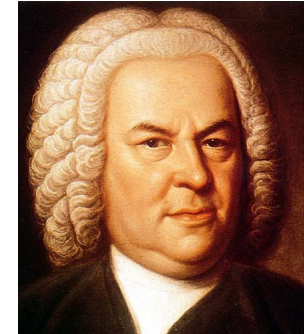
Piano Roll Representation

J.S. Bach, C-Major Fuge

(Well Tempered Piano, BWV 846)



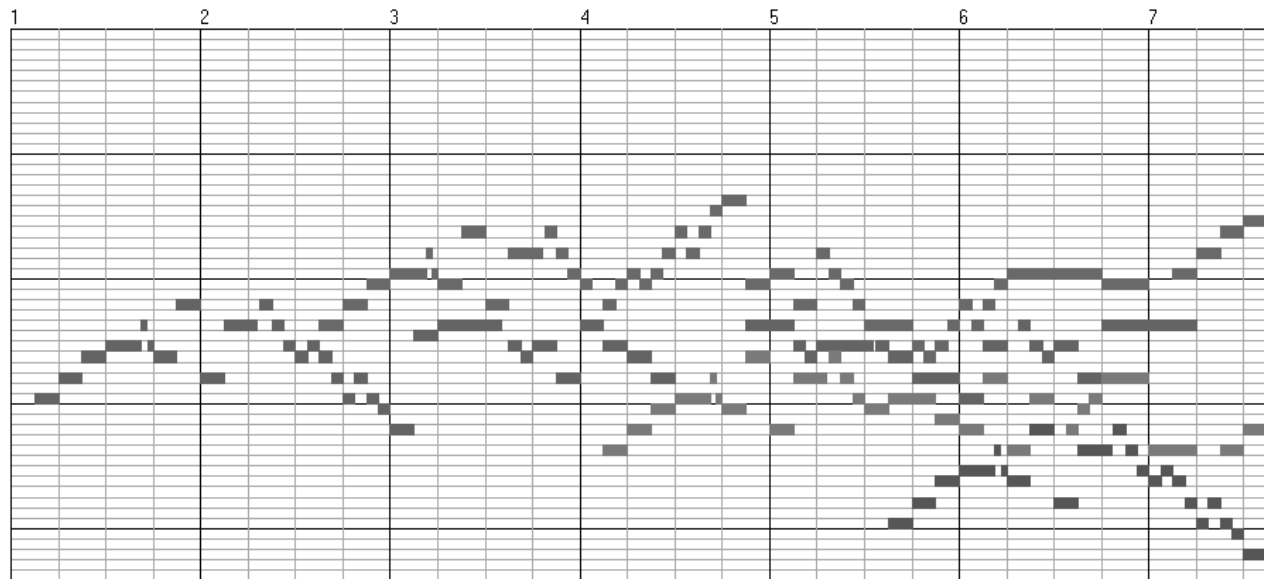
Piano Roll Representation



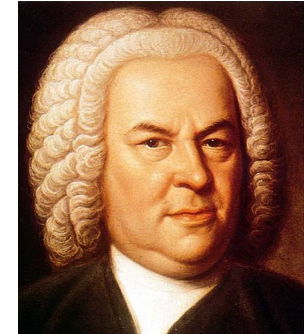
Query:



Goal: Find all occurrences of the query



Piano Roll Representation

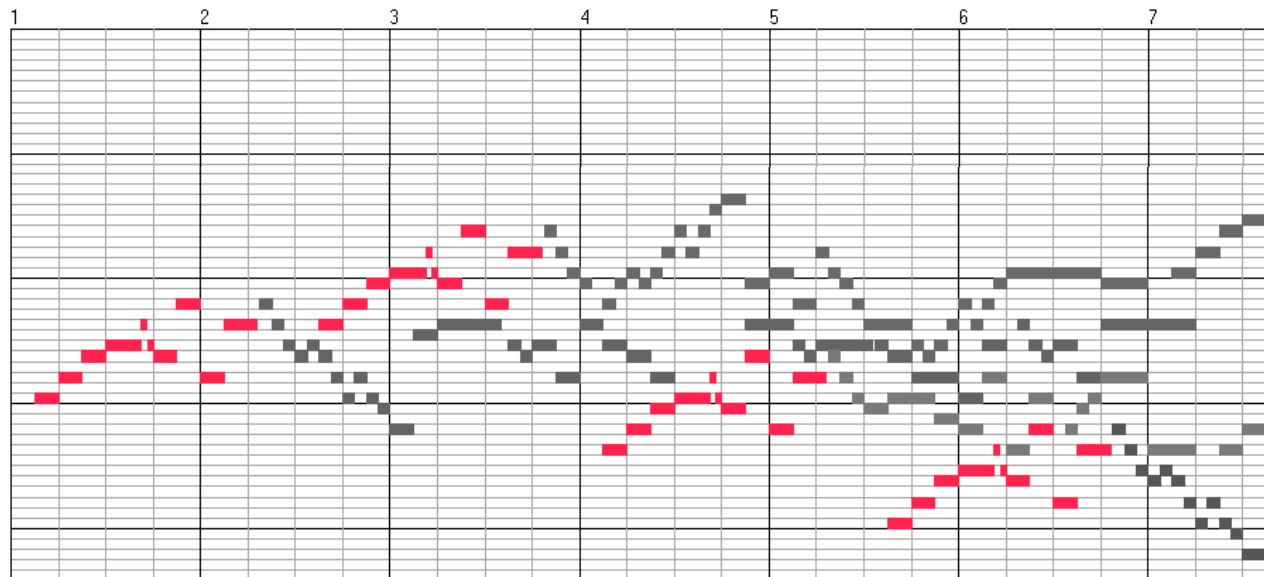


Query:

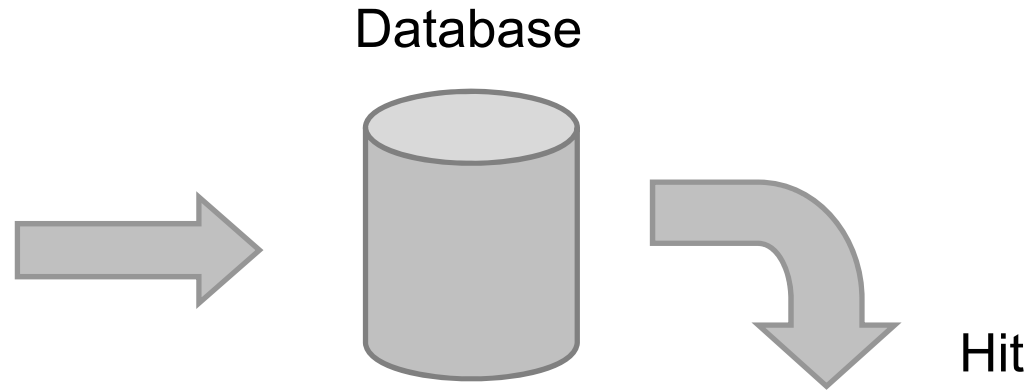


Goal: Find all occurrences of the query

Matches:



Music Retrieval



Audio ID

Bernstein (1962)
Beethoven, Symphony No. 5

Version ID

Beethoven, Symphony No. 5:

- Bernstein (1962)
- Karajan (1982)
- Gould (1992)



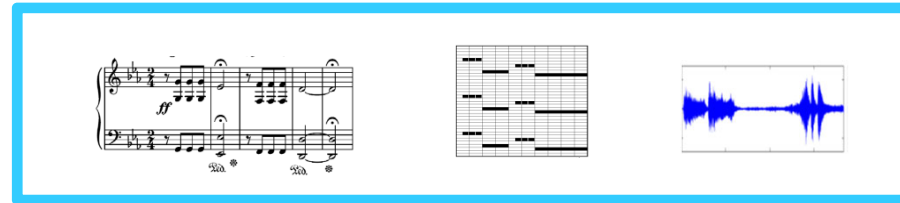
Category ID

- Beethoven, Symphony No. 9
- Beethoven, Symphony No. 3
- Haydn Symphony No. 94



Music Retrieval

Modalities



Retrieval tasks:

Specificity

Granularity

Audio ID

High
specificity

Fragment-based
retrieval

Version ID

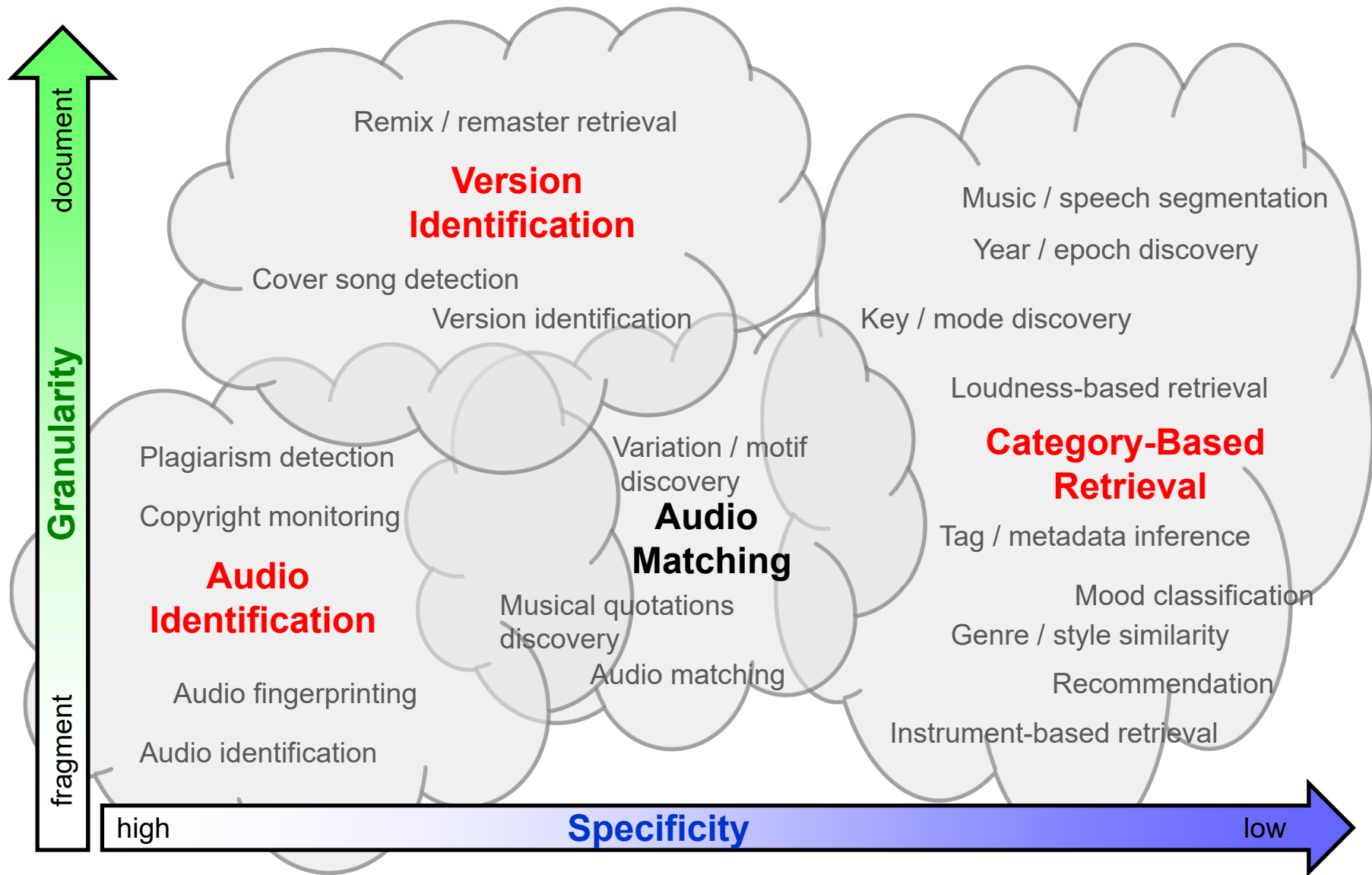


Category ID

Low
specificity

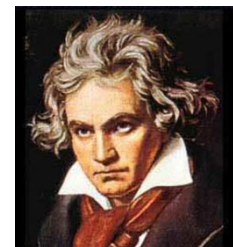
Document-based
retrieval

Music Retrieval



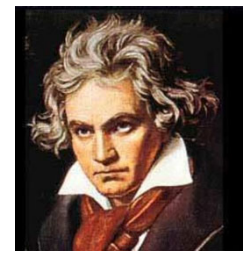
Music Synchronization: Audio-Audio

Beethoven's Fifth

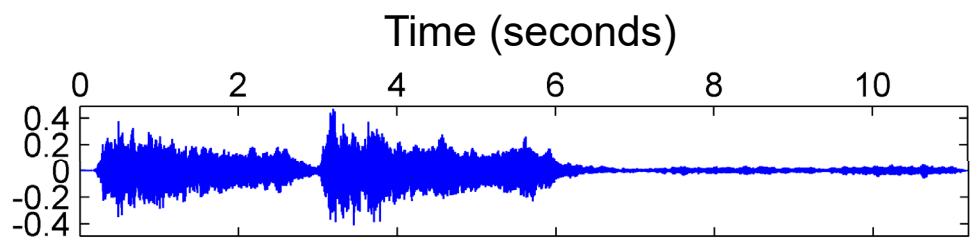


Music Synchronization: Audio-Audio

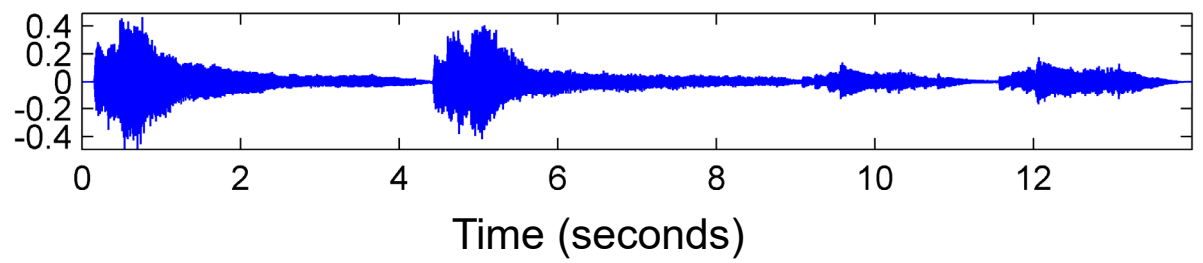
Beethoven's Fifth



Karajan
(Orchester)

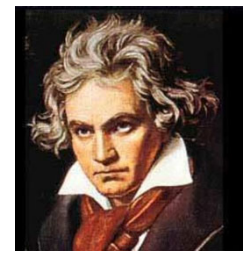


Gould
(Piano)

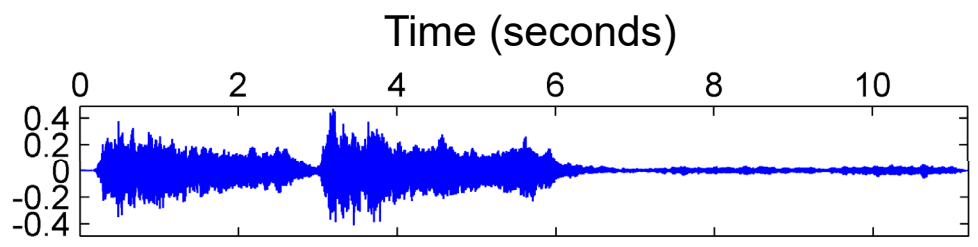


Music Synchronization: Audio-Audio

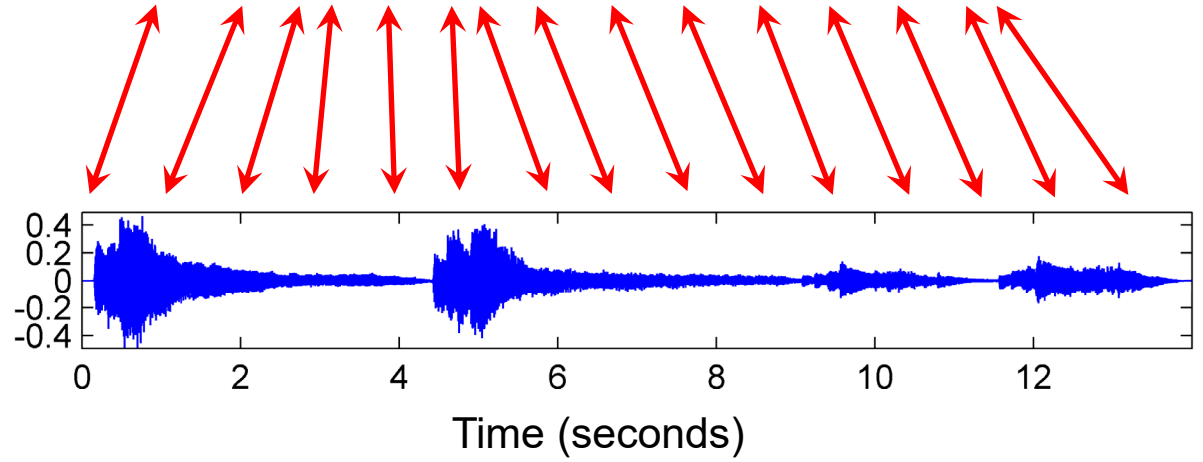
Beethoven's Fifth



Karajan
(Orchester)



Gould
(Piano)



Application: Interpretation Switcher



Music Synchronization: Audio-Audio

Task

Given: Two different audio recordings (two versions) of the same underlying piece of music.

Goal: Find for each position in one audio recording the **musically** corresponding position in the other audio recording.

Music Synchronization: Audio-Audio

Traditional Engineering Approach:

1.) Feature extraction

- Robust to variations (e.g., instrumentation, timbre, dynamics)
- Discriminative (e.g., capturing harmonic, melodic, tonal aspects)

➡ **Chroma features**

2.) Temporal alignment

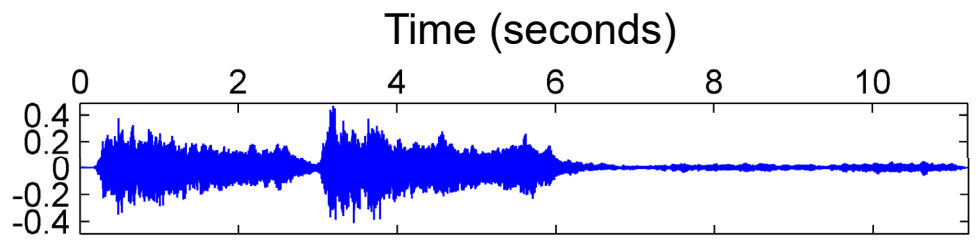
- Capturing local and global tempo variations
- Trade-off: Robustness vs. accuracy
- Efficiency

➡ **Dynamic time warping (DTW)**

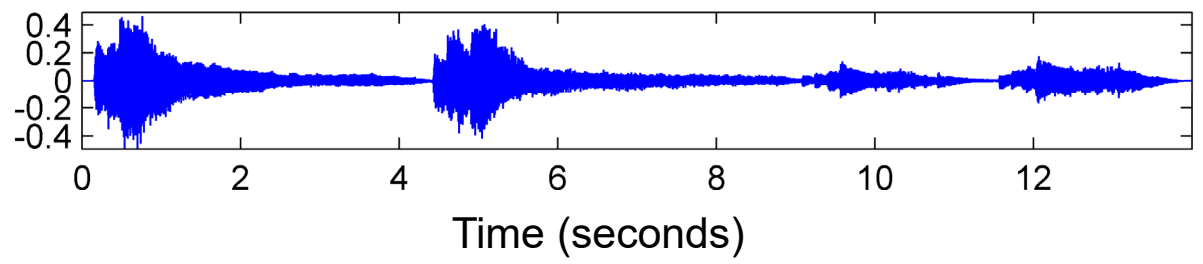
Music Synchronization: Audio-Audio

Beethoven's Fifth

Karajan
(Orchester)



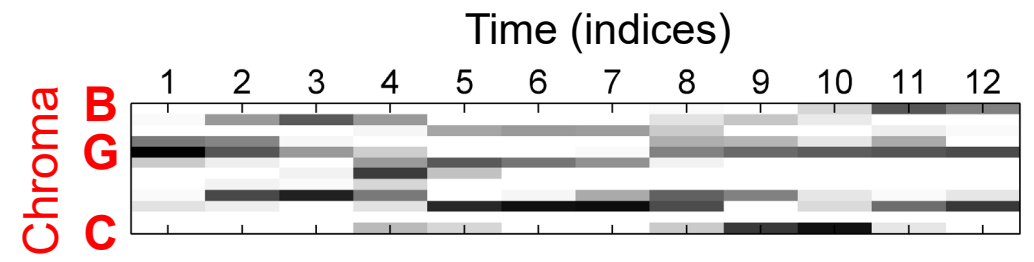
Gould
(Piano)



Music Synchronization: Audio-Audio

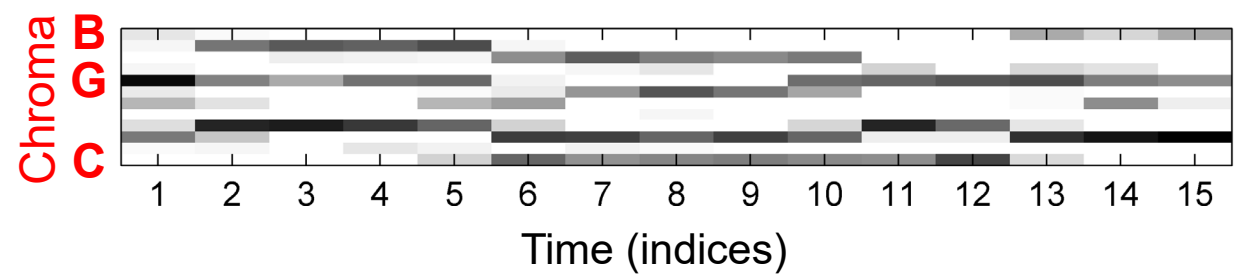
Beethoven's Fifth

Karajan
(Orchester)



Time–chroma representations

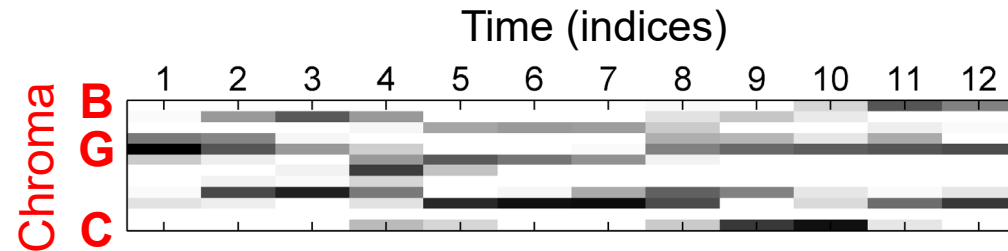
Gould
(Piano)



Music Synchronization: Audio-Audio

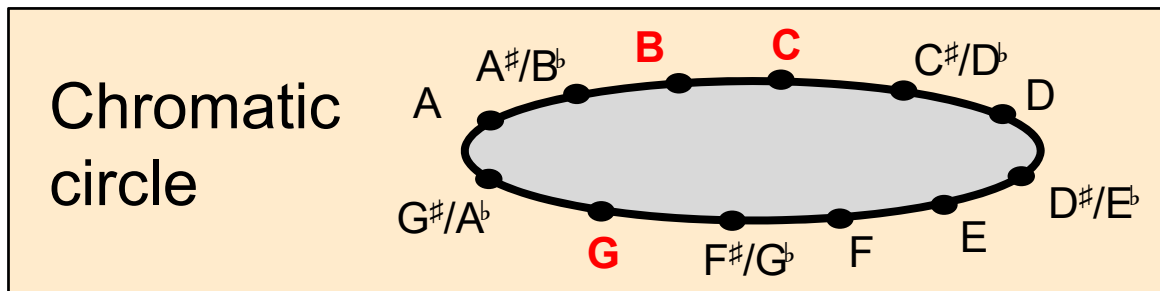
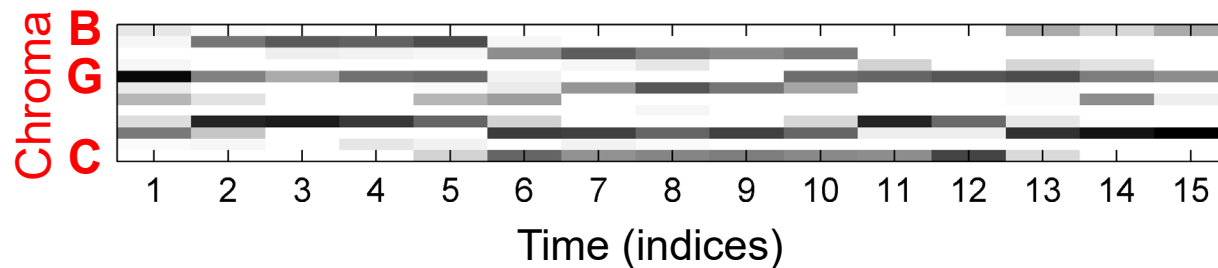
Beethoven's Fifth

Karajan
(Orchester)



Time–chroma representations

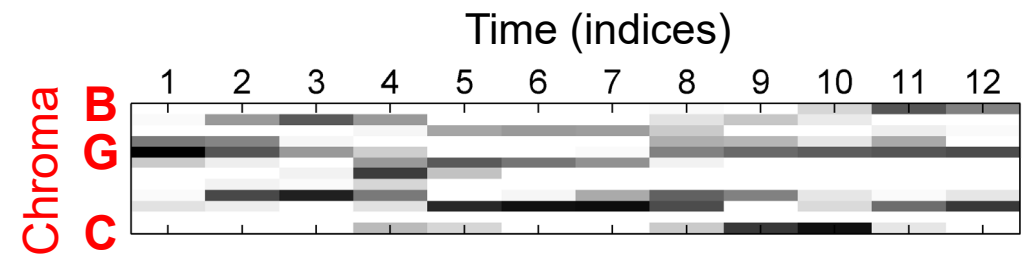
Gould
(Piano)



Music Synchronization: Audio-Audio

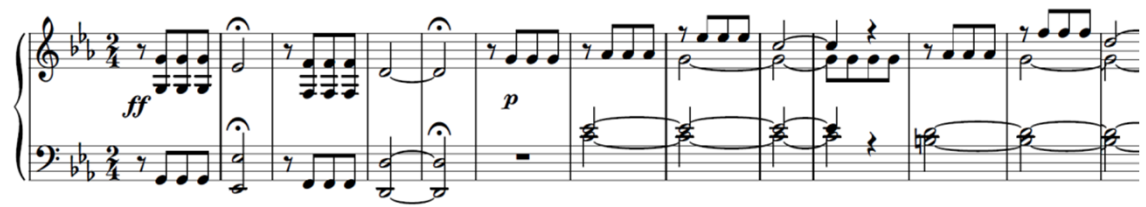
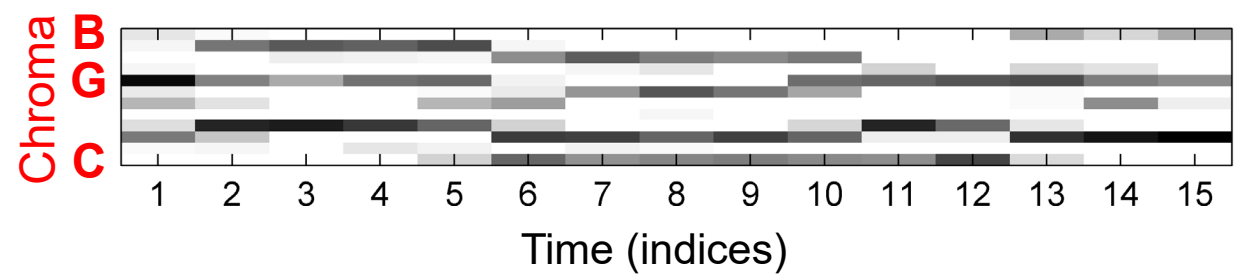
Beethoven's Fifth

Karajan
(Orchester)



Time–chroma representations

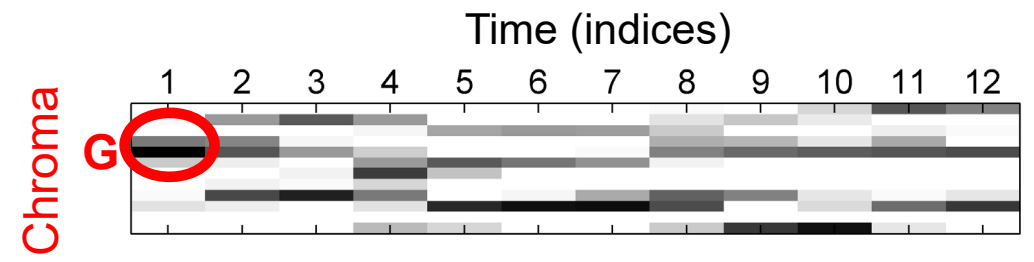
Gould
(Piano)



Music Synchronization: Audio-Audio

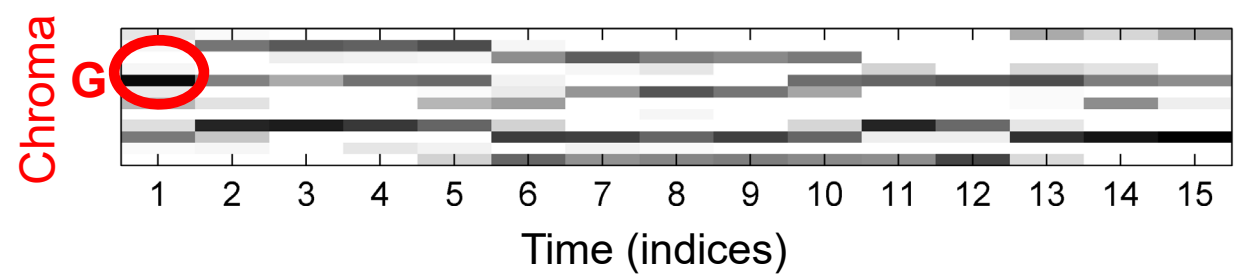
Beethoven's Fifth

Karajan
(Orchester)



Time–chroma representations

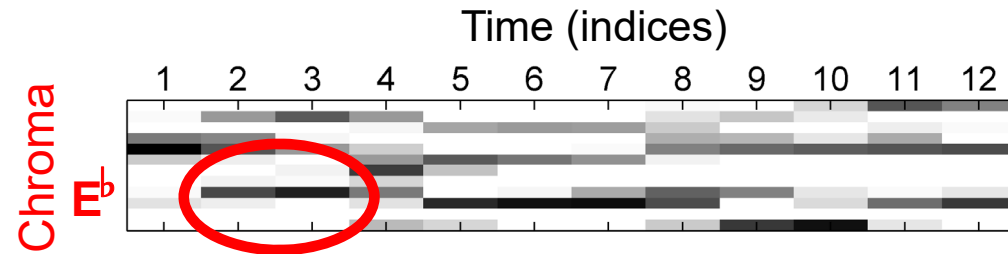
Gould
(Piano)



Music Synchronization: Audio-Audio

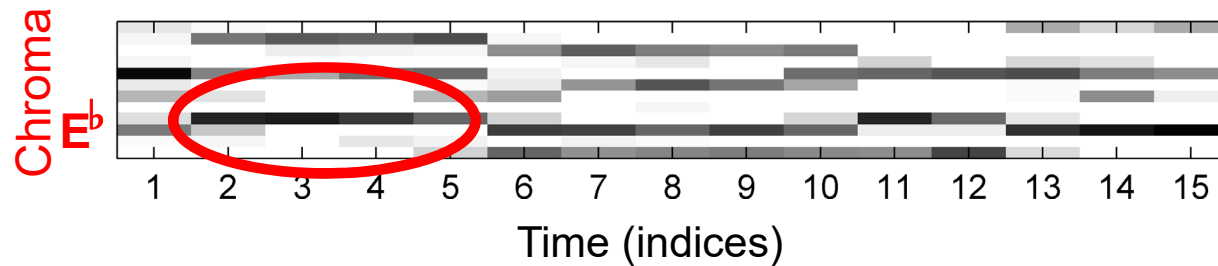
Beethoven's Fifth

Karajan
(Orchester)

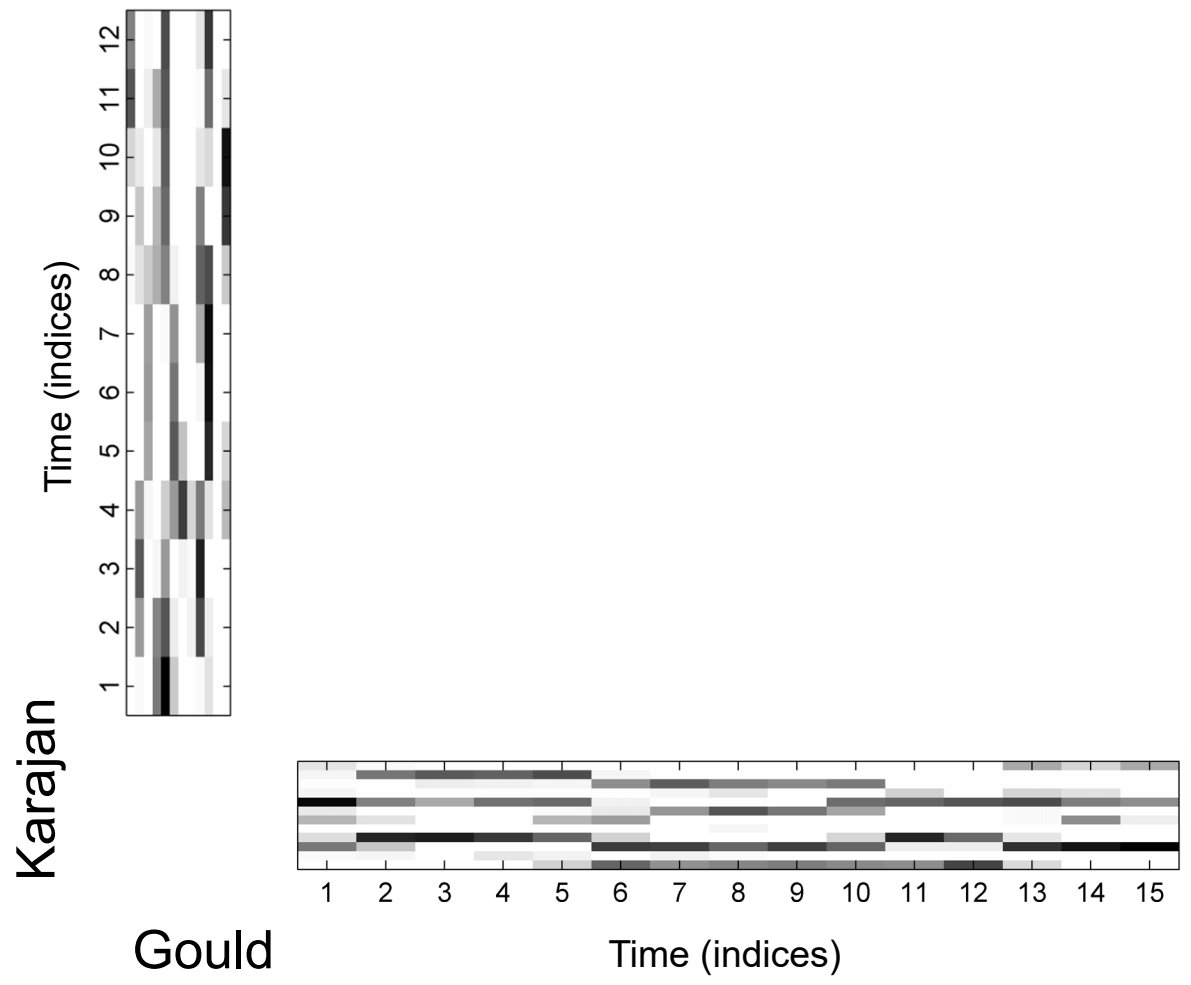


Time–chroma representations

Gould
(Piano)

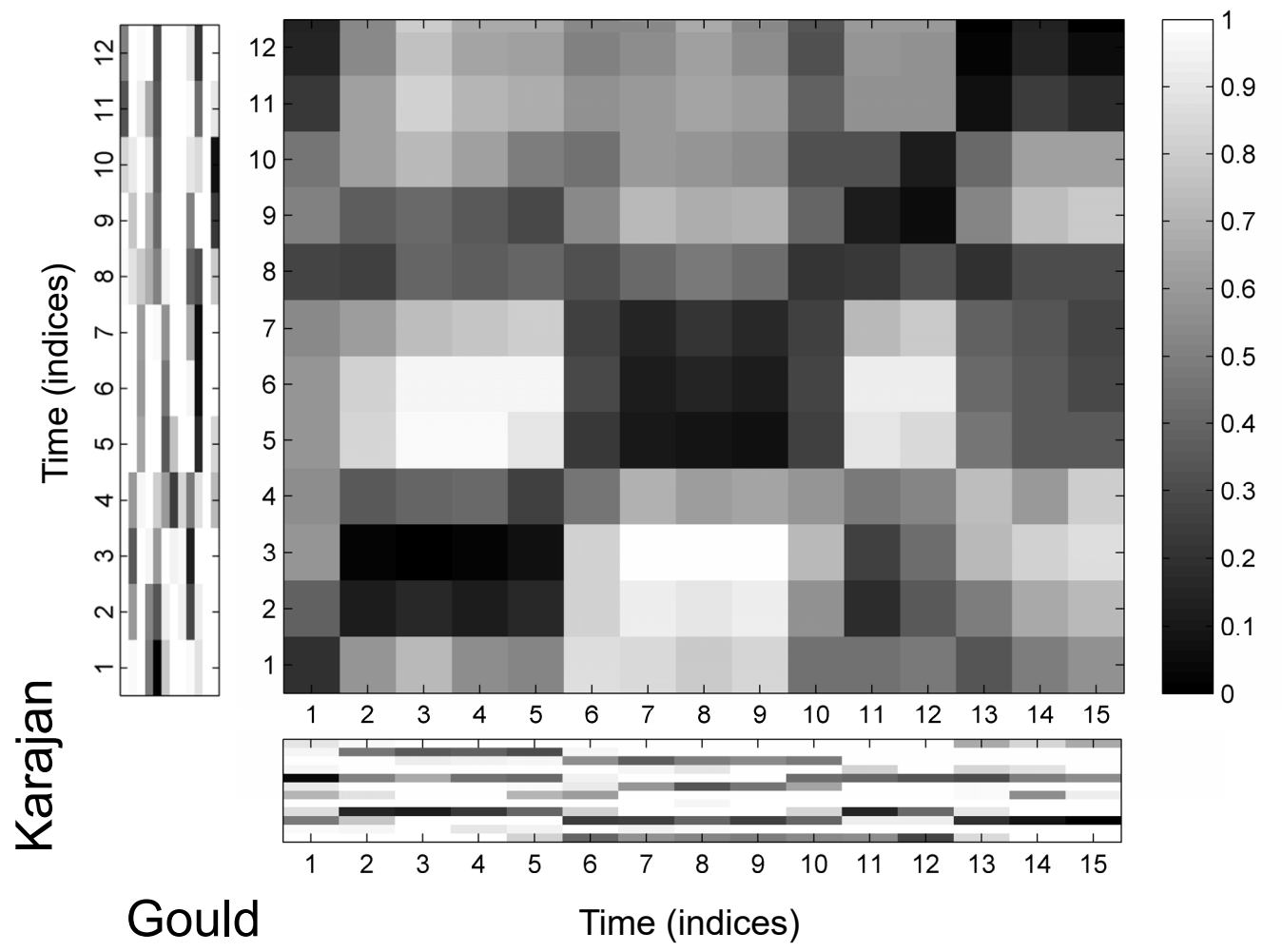


Music Synchronization: Audio-Audio



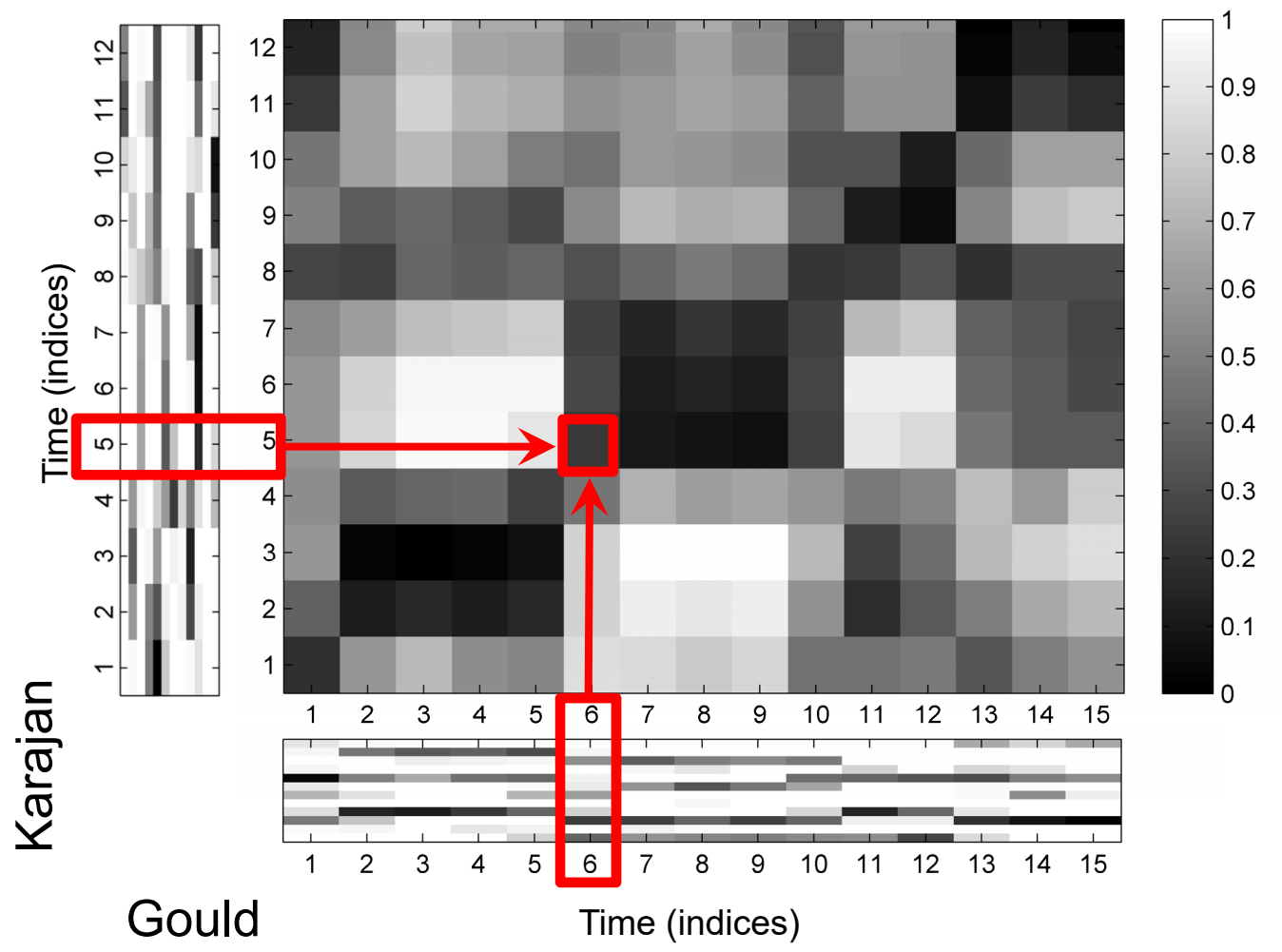
Music Synchronization: Audio-Audio

Cost matrix



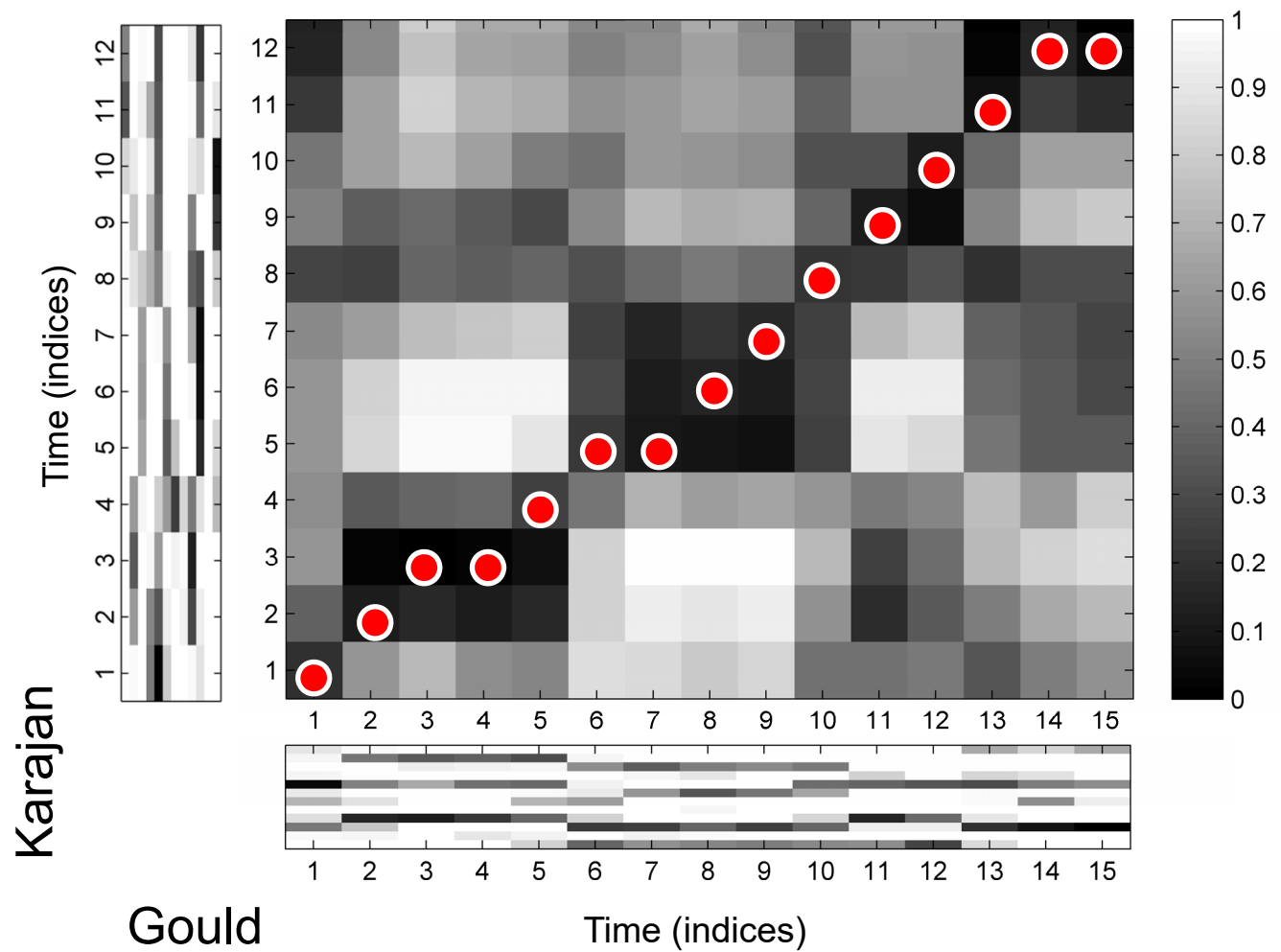
Music Synchronization: Audio-Audio

Cost matrix



Music Synchronization: Audio-Audio

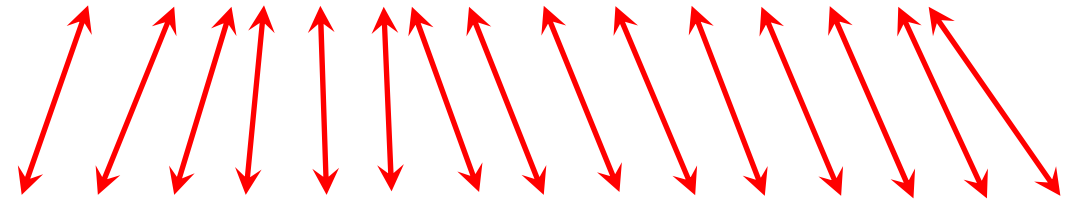
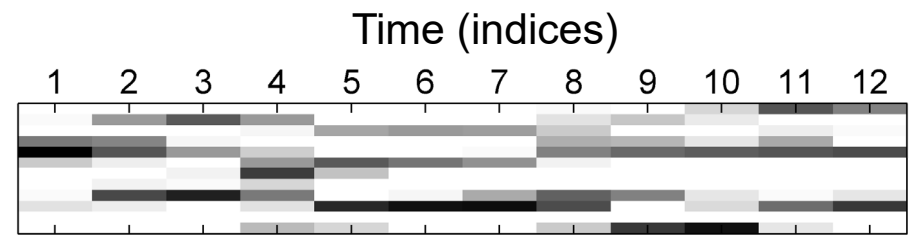
Cost-minimizing warping path



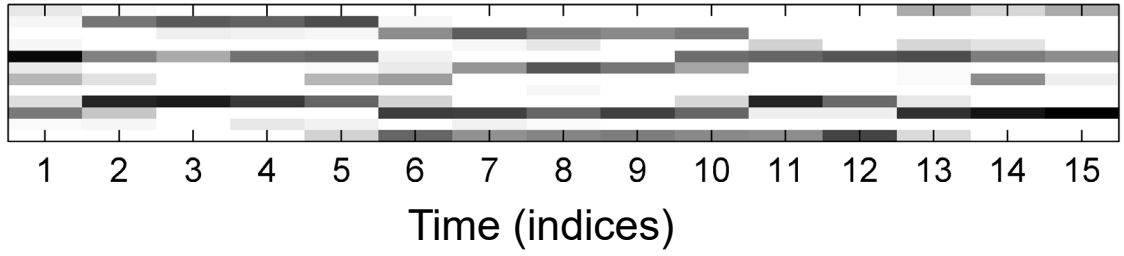
Music Synchronization: Audio-Audio

Cost-minimizing warping path = Optimal alignment

Karajan
(Orchester)



Gould
(Piano)



Music Synchronization: Audio-Audio

Deep Learning Approaches

- Learn audio features from data
 - Should be able to achieve high alignment accuracy
 - Should be robust to performance variations
 - Musical relevance?
- Alignment problem
 - Pre-aligned data for training
 - Part of loss function → differentiability?

CTC-Loss

Graves et al.: Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. ICML, 2006

Soft-DTW

Cuturi, Blondel: Soft-DTW: A Differentiable Loss Function for Time-Series. ICML, 2017

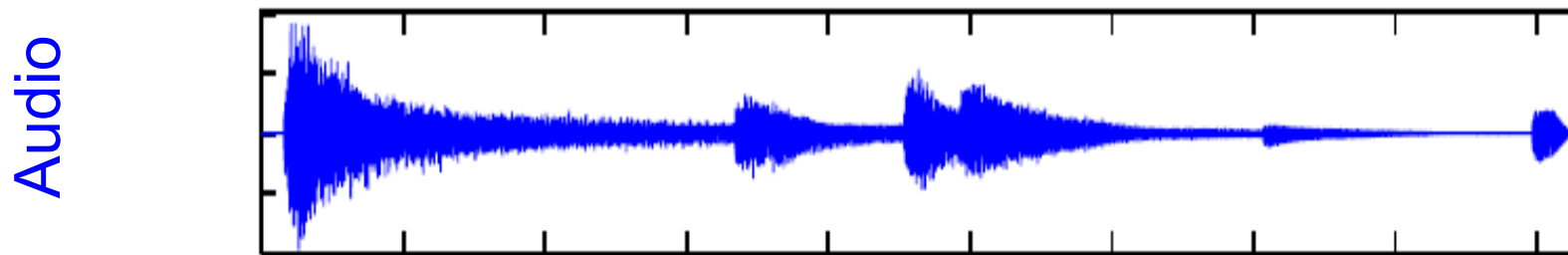
Music Synchronization: Image-Audio

Image

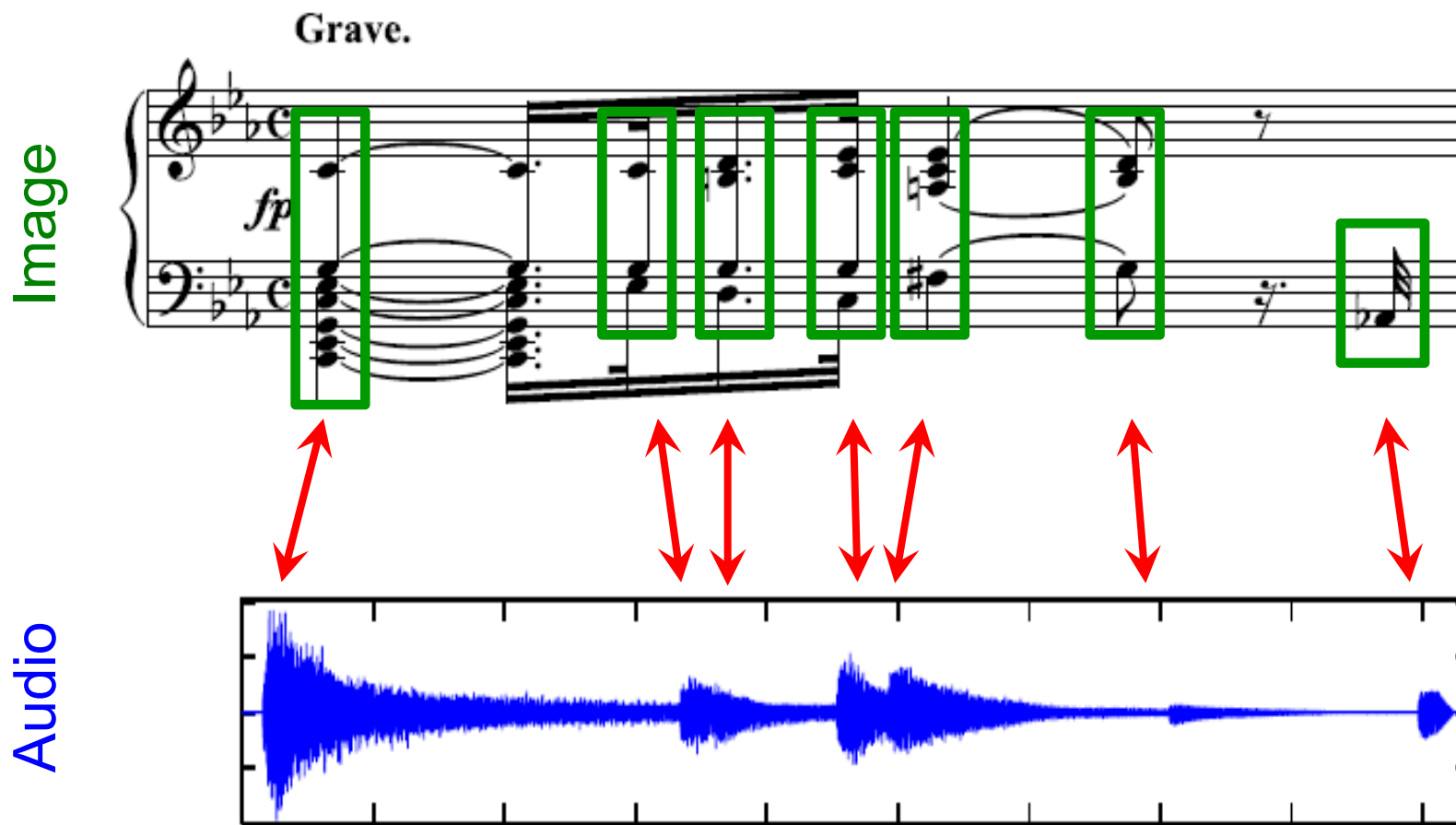
Grave.



A musical score for piano, marked "Grave." and "fp". The score is written for a grand piano, with a treble clef on the upper staff and a bass clef on the lower staff. The key signature is three flats (B-flat, E-flat, A-flat) and the time signature is common time (C). The music features a slow, somber mood with a focus on sustained chords and a few melodic lines.



Music Synchronization: Image-Audio



Application: Score Viewer

The screenshot displays two windows from a music application. The top window, titled "ScoreViewer", shows a musical score for "Beethoven - Klaviersonaten Band 1 - Henle". The score is for "Sonata no.8 in C minor, op.13 'Pathétique' / Rondo (Allegro)". The score is displayed in a scrollable view with a yellow highlight on the "Rondo Allegro" section. Below the score, there are navigation controls: "Track: 29 / 54", "Bar: 1 / 211", and "Page: 159 / 285". There are also "Score Following On", "Play", and "Stop" buttons.

The bottom window, titled "AudioViewer", shows a playlist for "Beethoven - Piano Sonatas-Alfred Brendel". The playlist is for "Disc 1" and contains 11 tracks. The current track is "11 Sonata no.8 in C minor, op.13 'Pathétique' / Rondo (Allegro)" with a duration of 4:20. Below the playlist, there is a waveform visualization and a progress bar. At the bottom, there are navigation controls: "Disc: 1 / 11", "Track: 11 / 11", and "Time: 00:00.00 / 4:30.35". There are also "Play" and "Stop" buttons.



Music Synchronization: Image-Audio

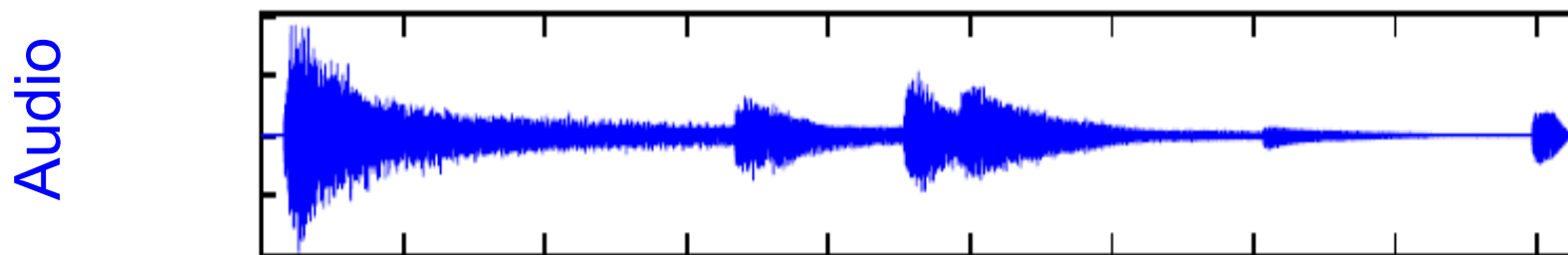
Image

Grave.

fp



A musical score for piano in a slow tempo, marked 'Grave.' and 'fp' (fortissimo). The score is written for a grand piano, with a treble clef on the upper staff and a bass clef on the lower staff. The key signature has two flats (B-flat and E-flat), and the time signature is common time (C). The music features a series of chords and melodic lines, with a prominent bass line in the lower register.



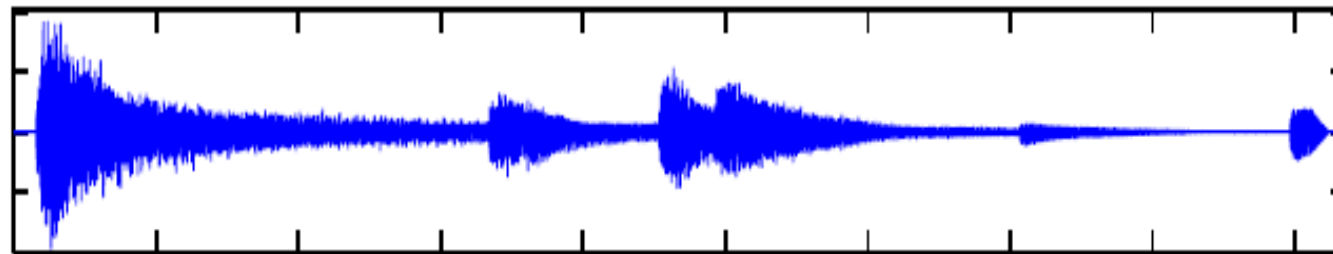
Music Synchronization: Image-Audio

Image Processing: Optical Music Recognition

Image



Audio



Music Synchronization: Image-Audio

Image Processing: Optical Music Recognition

Image



Audio



Audio Processing: Fourier Analysis

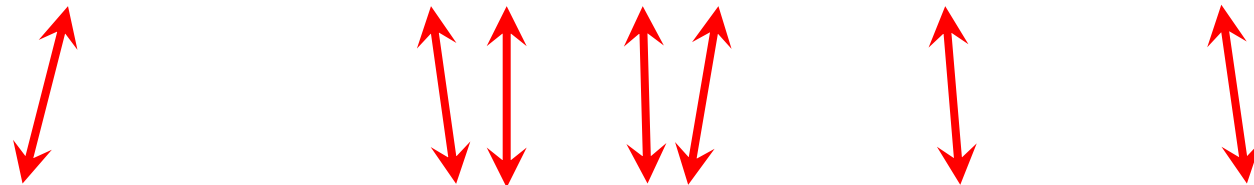
Music Synchronization: Image-Audio

Image Processing: Optical Music Recognition

Image



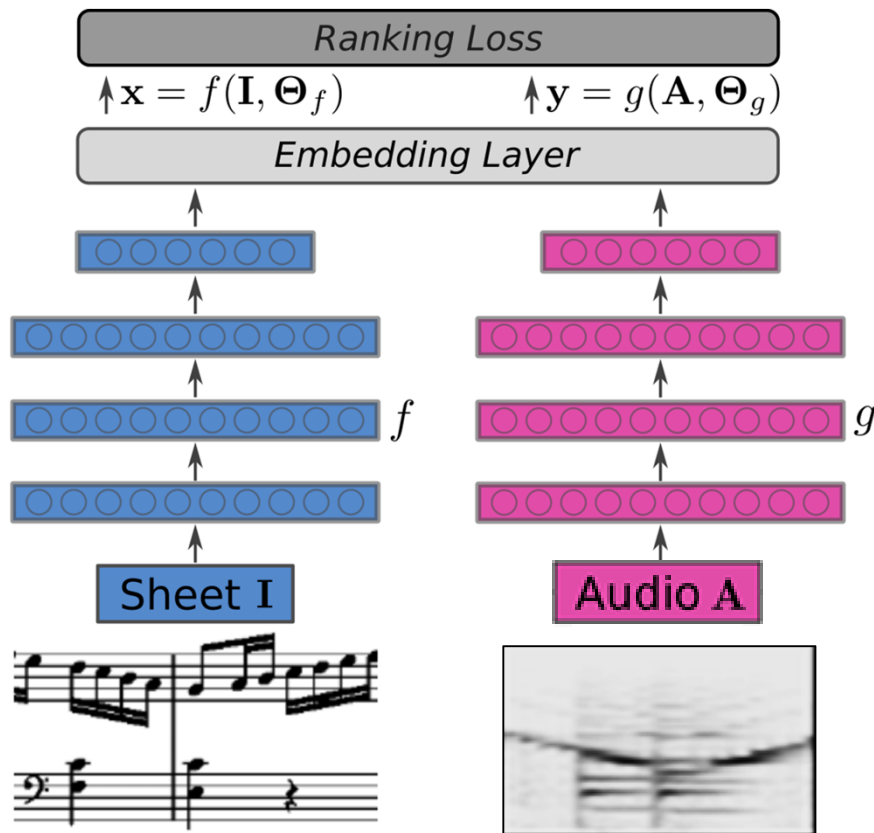
Audio



Audio Processing: Fourier Analysis

Music Synchronization: Image-Audio

Deep Learning Approach



- Deep learning
- Embedding techniques
- Triplet loss
- ...

Cross-Modal Retrieval

Dorfer et al.: End-to-End Cross-Modality Retrieval with CCA Projections and Pairwise Ranking Loss. International Journal of Multimedia Information Retrieval, 2018.

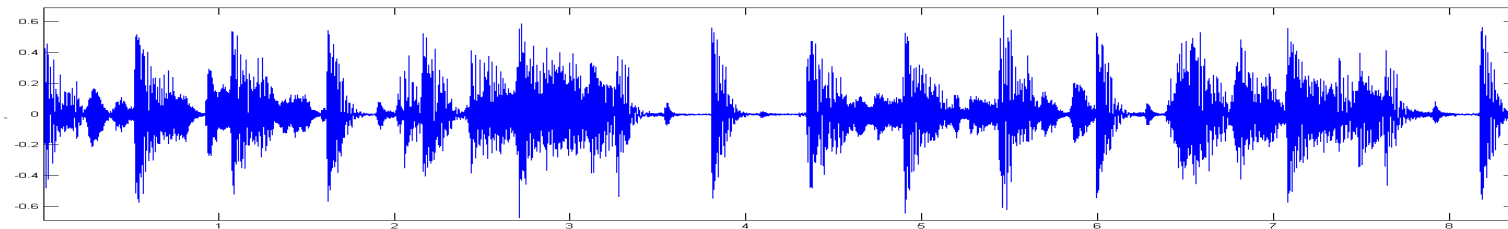
Music Processing

Coarse/Relative Level	Fine/Absolute Level
What do different versions or instances have in common?	What are the characteristics of a specific version or instance?
Provide coarse description: What makes up a piece of music?	Capture nuances and subtleties: What makes music come alive?
Identify despite of differences	Identify the differences
Example tasks: Music Retrieval Genre Classification Global Tempo Estimation	Example tasks: Music Transcription Performance Analysis Local Tempo Estimation

Tempo Estimation and Beat Tracking

Basic task: “Tapping the foot when listening to music”

Example: Queen – Another One Bites The Dust

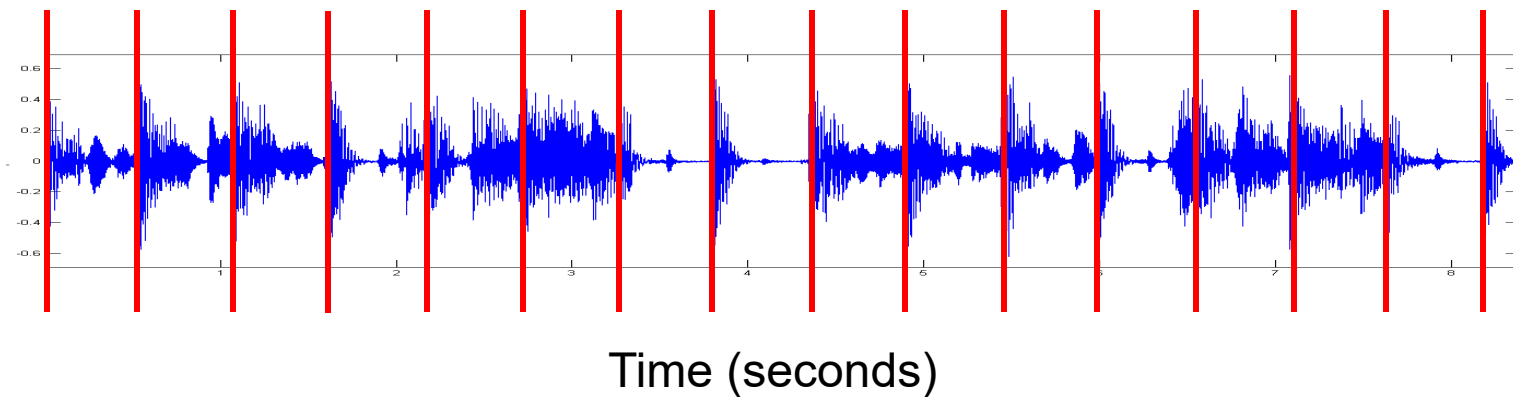


Time (seconds)

Tempo Estimation and Beat Tracking

Basic task: “Tapping the foot when listening to music”

Example: Queen – Another One Bites The Dust



Tempo Estimation and Beat Tracking

Light effects

Music recommendation

DJ

Audio editing

Tempo Estimation and Beat Tracking

Example: Chopin – Mazurka Op. 68-3

Pulse level: Quarter note

Tempo: ???



Tempo Estimation and Beat Tracking

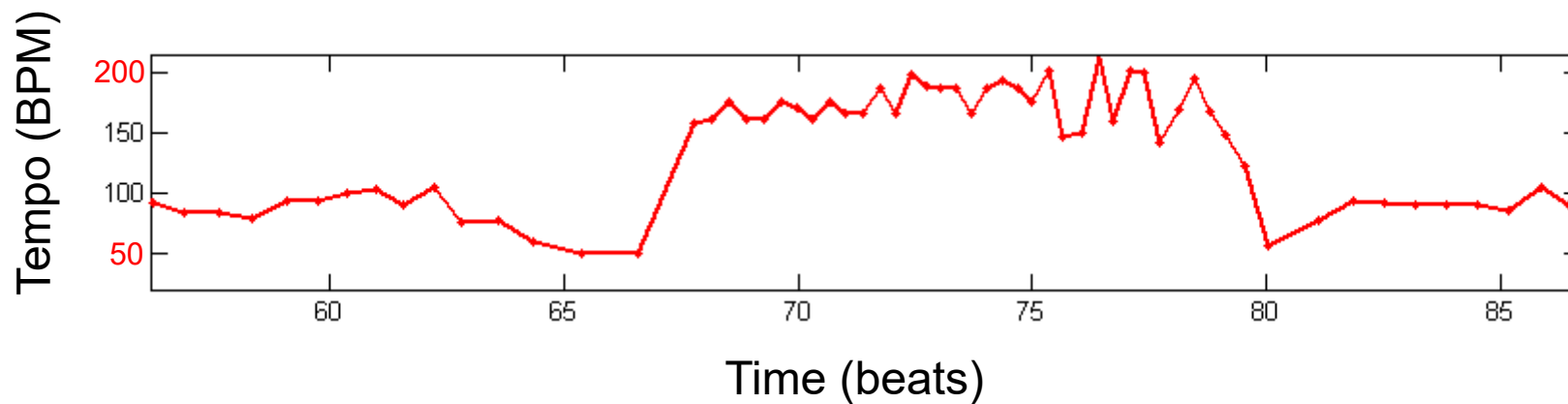
Example: Chopin – Mazurka Op. 68-3

Pulse level: Quarter note

Tempo: **50-200 BPM**



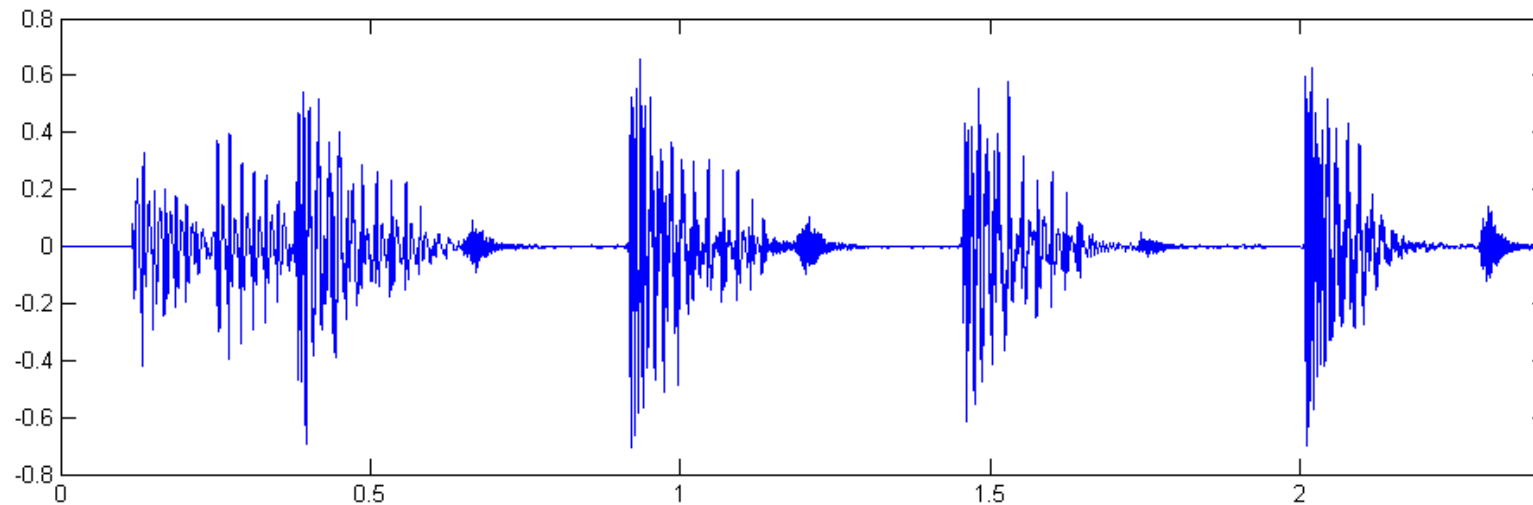
Tempo curve



Tempo Estimation and Beat Tracking

Tasks

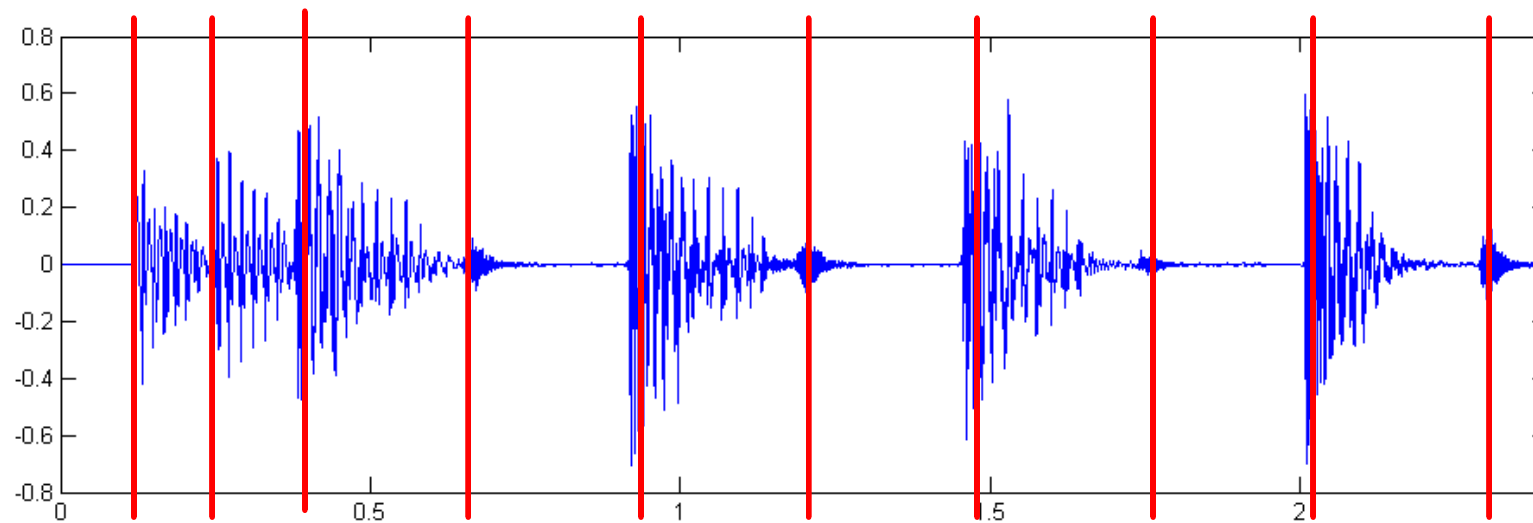
- Onset detection
- Beat tracking
- Tempo estimation



Tempo Estimation and Beat Tracking

Tasks

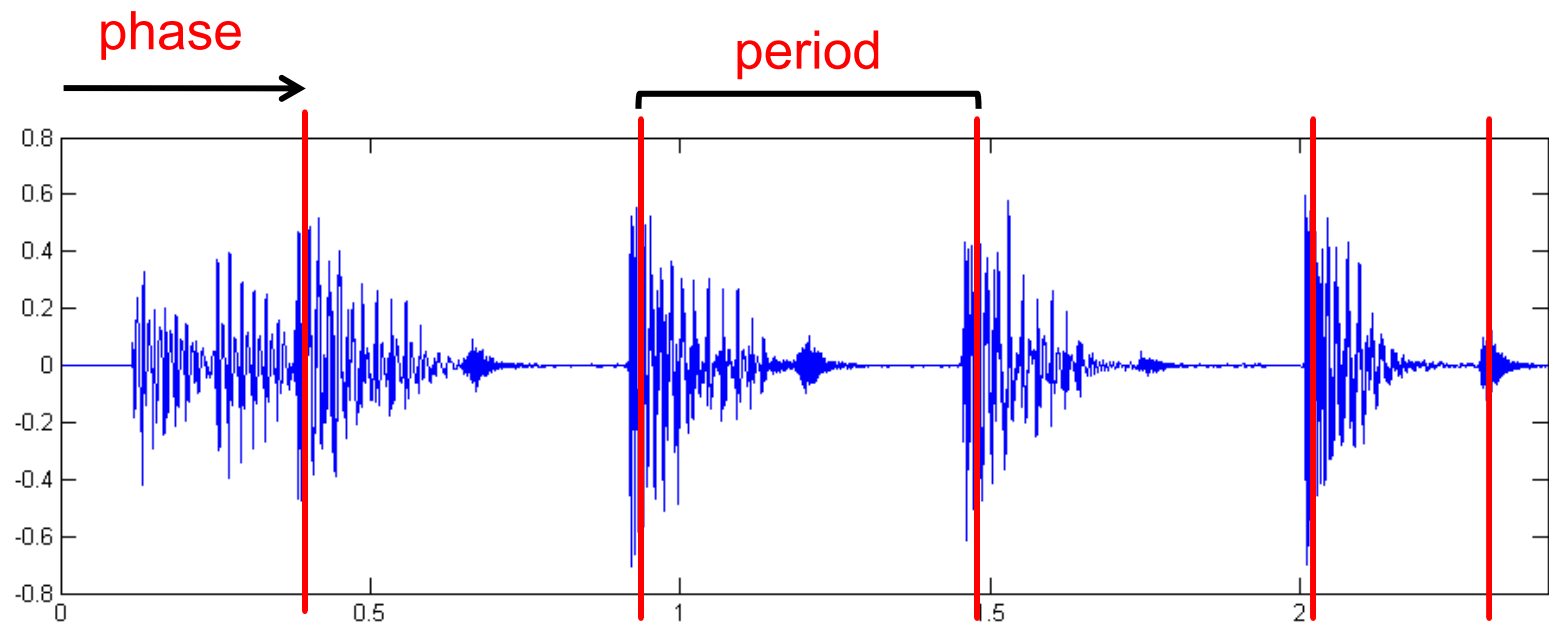
- Onset detection
- Beat tracking
- Tempo estimation



Tempo Estimation and Beat Tracking

Tasks

- Onset detection
- **Beat tracking**
- Tempo estimation



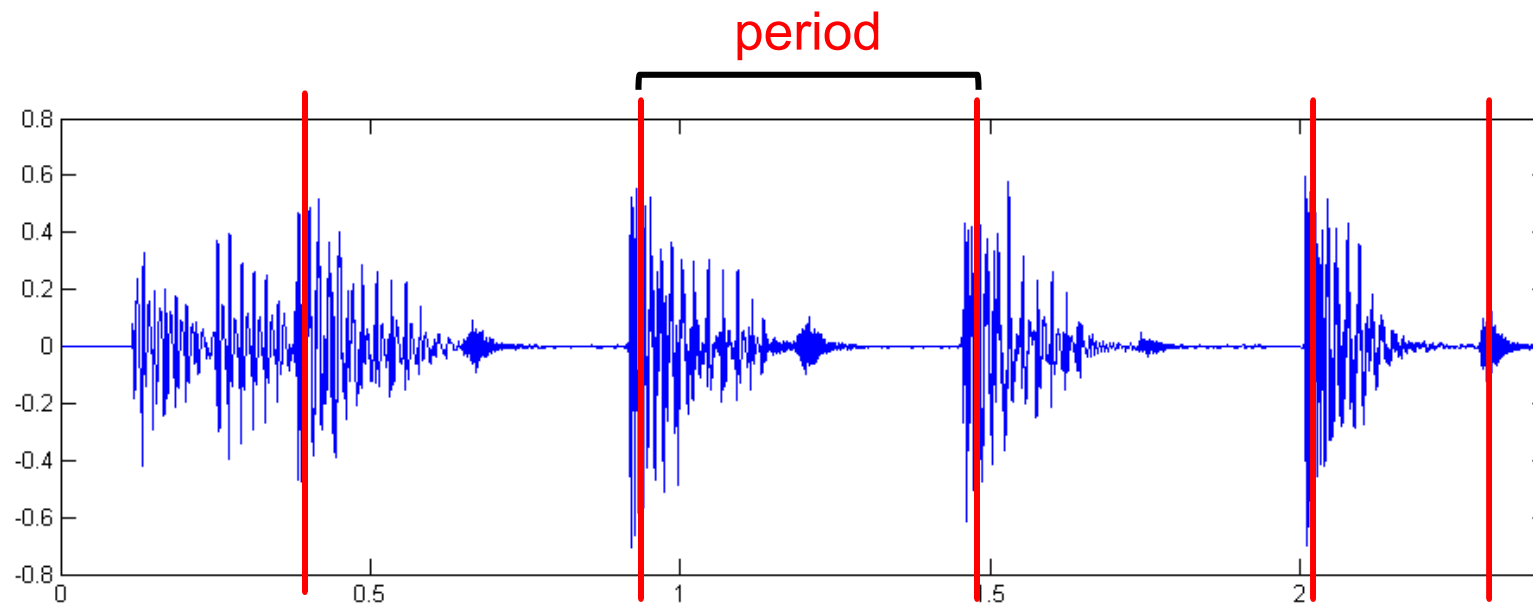
Tempo Estimation and Beat Tracking

Tasks

- Onset detection
- Beat tracking
- Tempo estimation

Tempo := $60 / \text{period}$

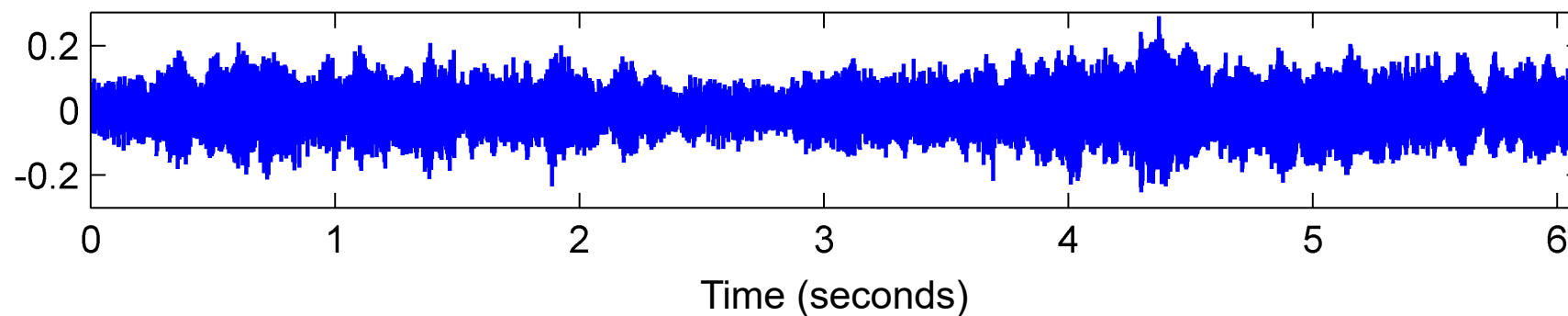
Beats per minute (BPM)



Onset Detection (Spectral Flux)

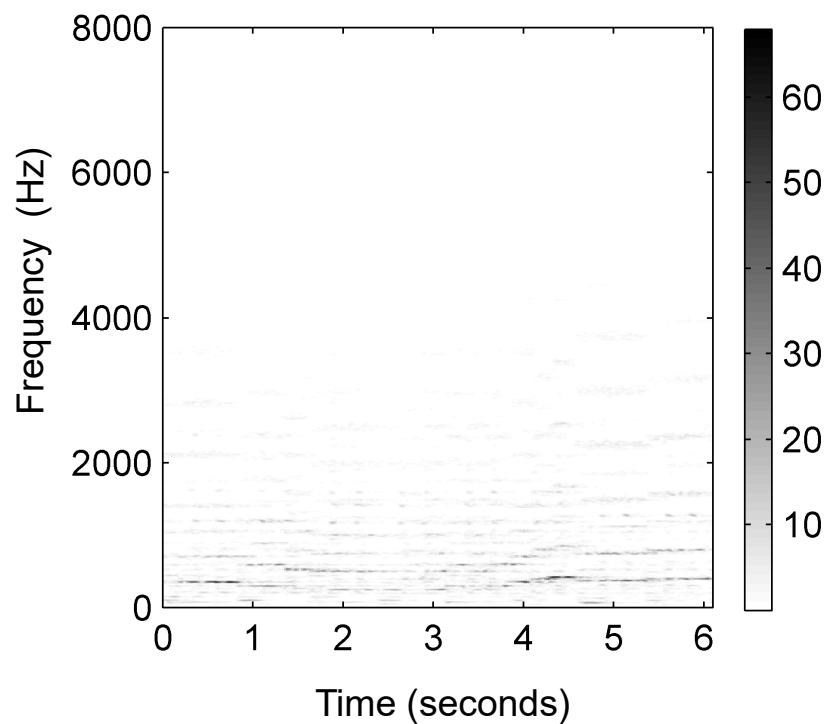


Audio recording



Onset Detection (Spectral Flux)

Magnitude spectrogram $|X|$

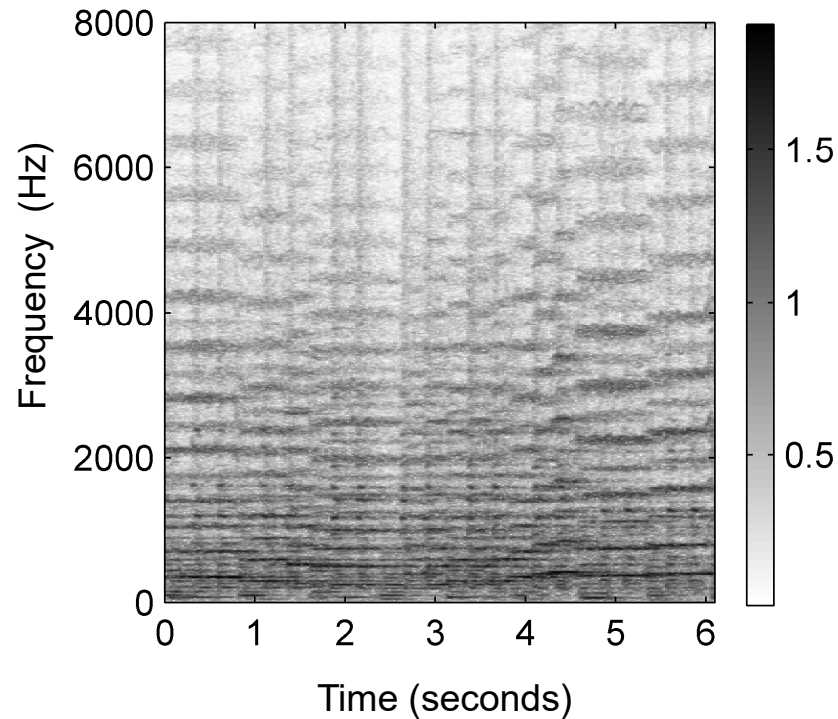


Steps:

1. Spectrogram

Onset Detection (Spectral Flux)

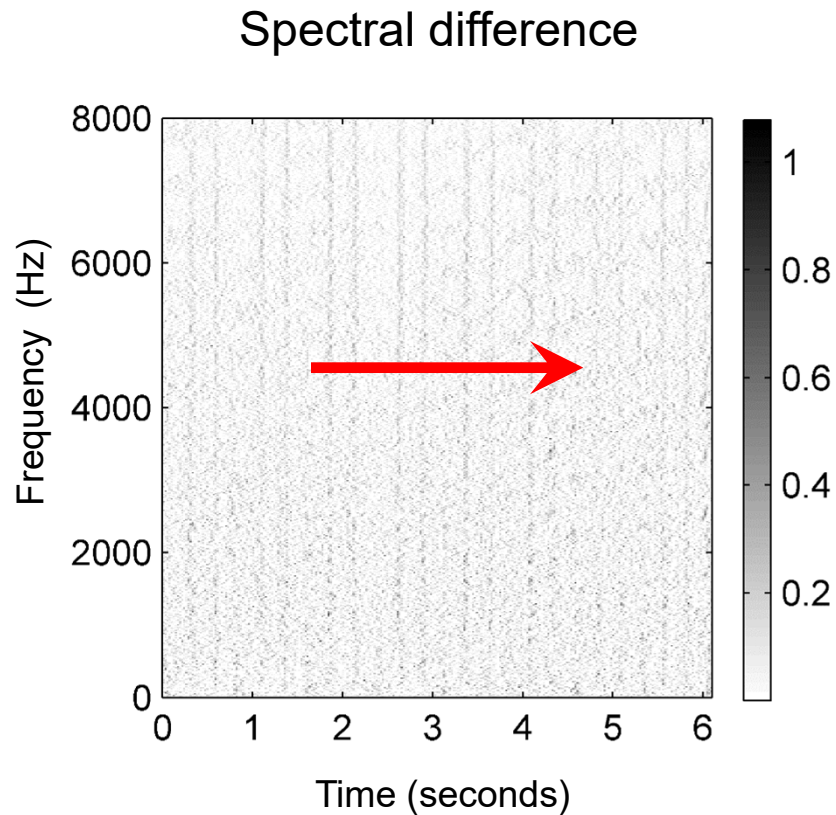
Compressed spectrogram Y



Steps:

1. Spectrogram
2. Logarithmic compression

Onset Detection (Spectral Flux)

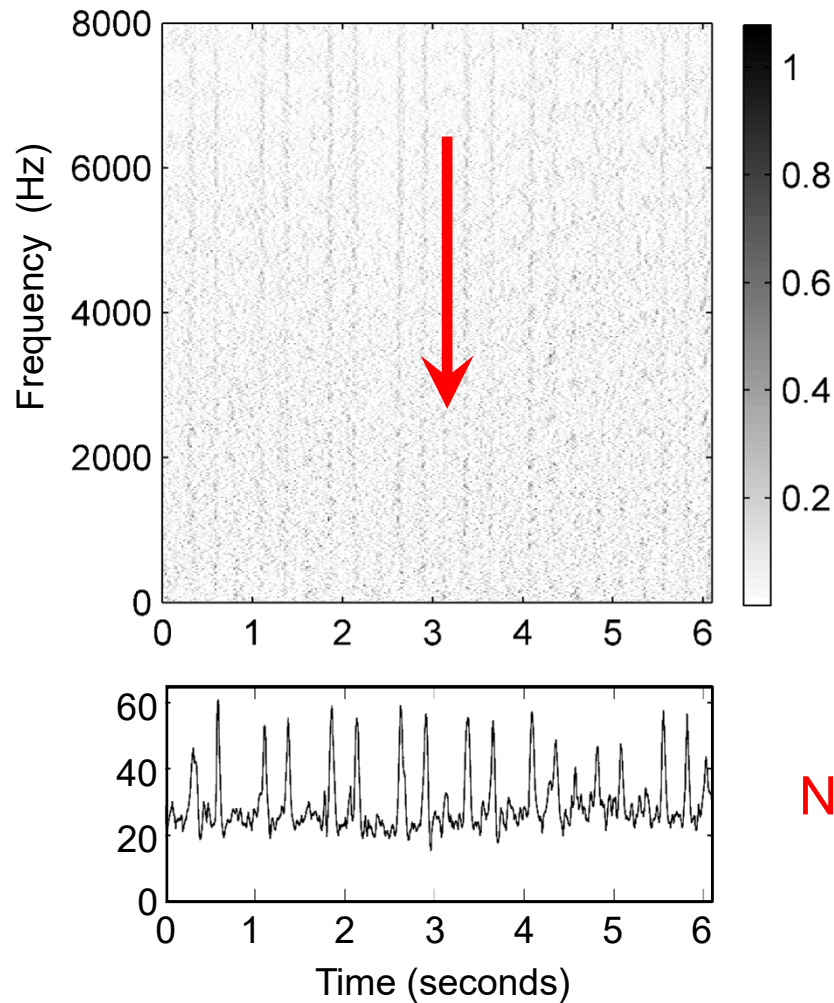


Steps:

1. Spectrogram
2. Logarithmic compression
3. Differentiation & half wave rectification

Onset Detection (Spectral Flux)

Spectral difference



Steps:

1. Spectrogram
2. Logarithmic compression
3. Differentiation & half wave rectification
4. Accumulation

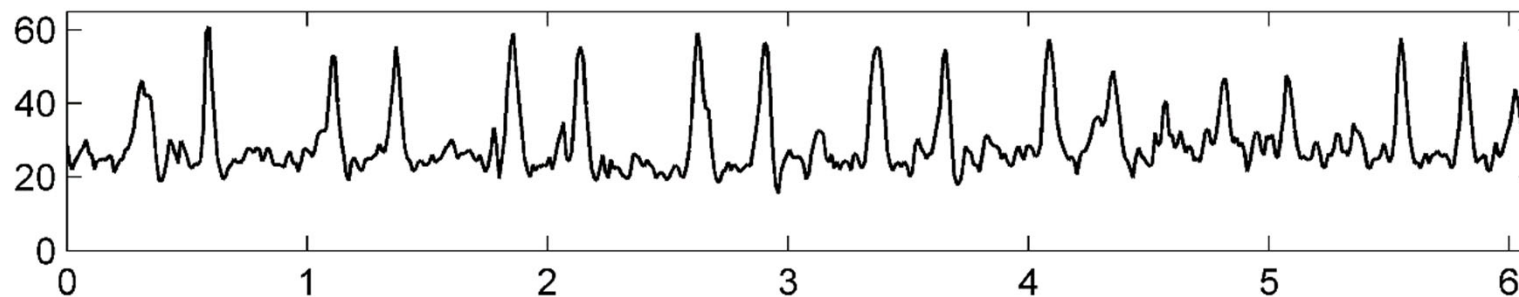
Novelty curve

Onset Detection (Spectral Flux)

Steps:

1. Spectrogram
2. Logarithmic compression
3. Differentiation & half wave rectification
4. Accumulation

Novelty function



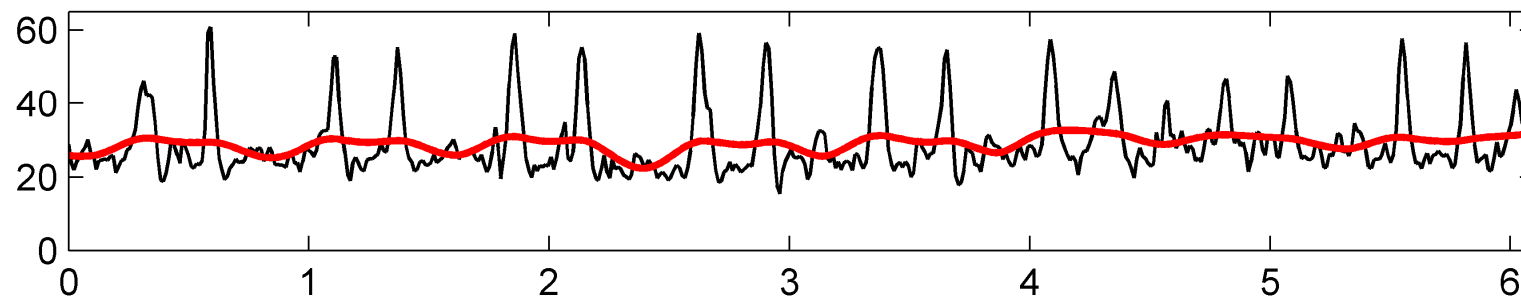
Onset Detection (Spectral Flux)

Steps:

1. Spectrogram
2. Logarithmic compression
3. Differentiation & half wave rectification
4. Accumulation
5. Normalization

Novelty function

Substraction of local average

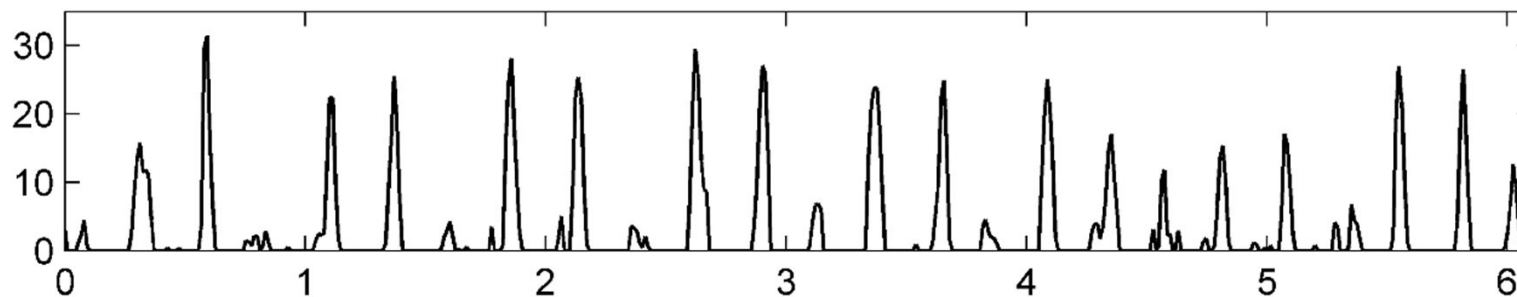


Onset Detection (Spectral Flux)

Steps:

1. Spectrogram
2. Logarithmic compression
3. Differentiation & half wave rectification
4. Accumulation
5. Normalization

Normalized novelty function



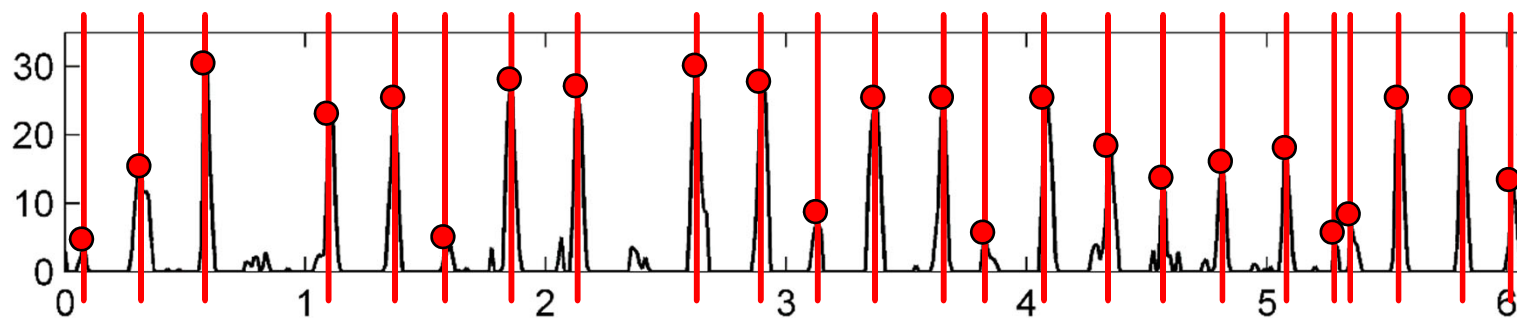
Onset Detection (Spectral Flux)

Steps:

1. Spectrogram
2. Logarithmic compression
3. Differentiation & half wave rectification
4. Accumulation
5. Normalization

Normalized novelty function

Peak positions indicate beat candidates



Onset Detection (Spectral Flux)

Deep Learning

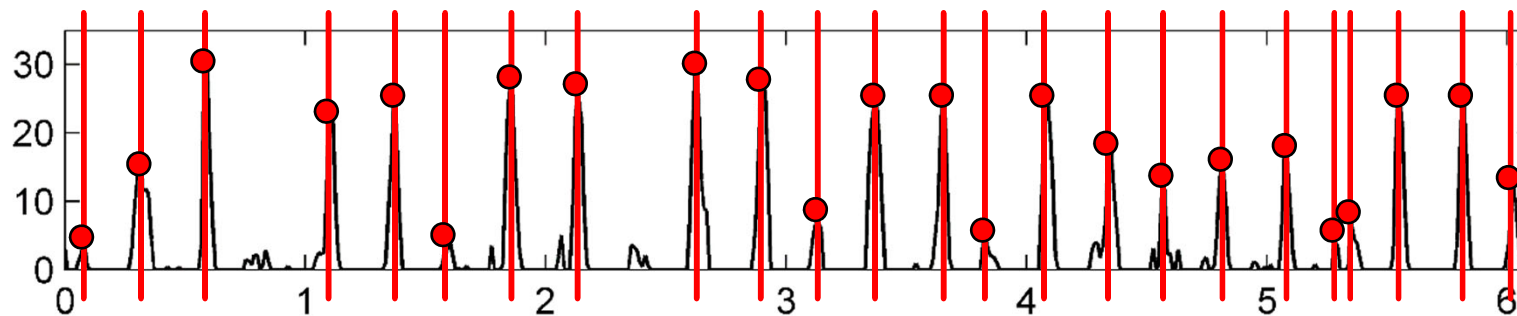
1. Input representation
2. Sigmoid activation
3. Convolution & rectified linear unit (ReLU)
4. Pooling
5. Convolution & ReLU

Steps:

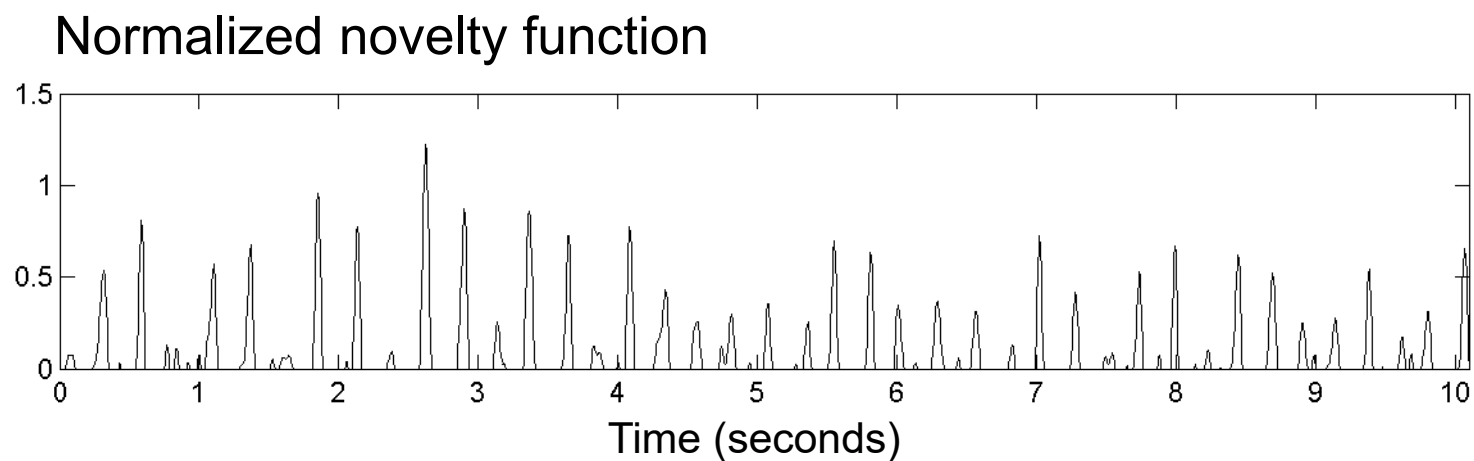
1. Spectrogram
2. Logarithmic compression
3. Differentiation & half wave rectification
4. Accumulation
5. Normalization

Normalized novelty function

Peak positions indicate beat candidates

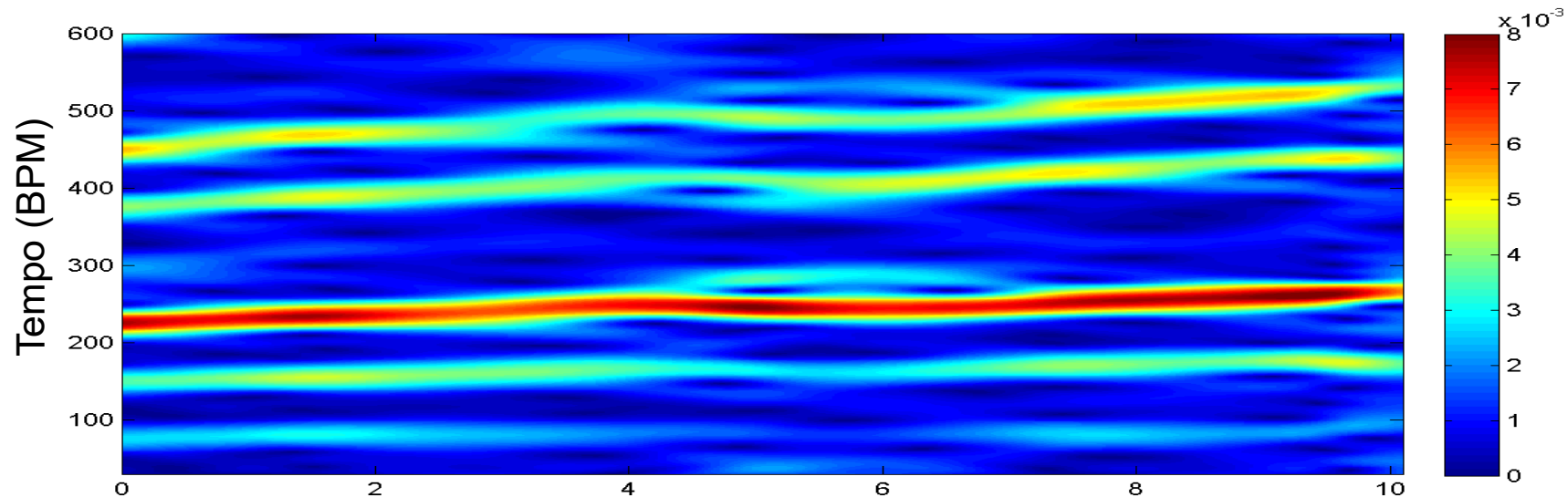


Local Pulse and Tempo Tracking

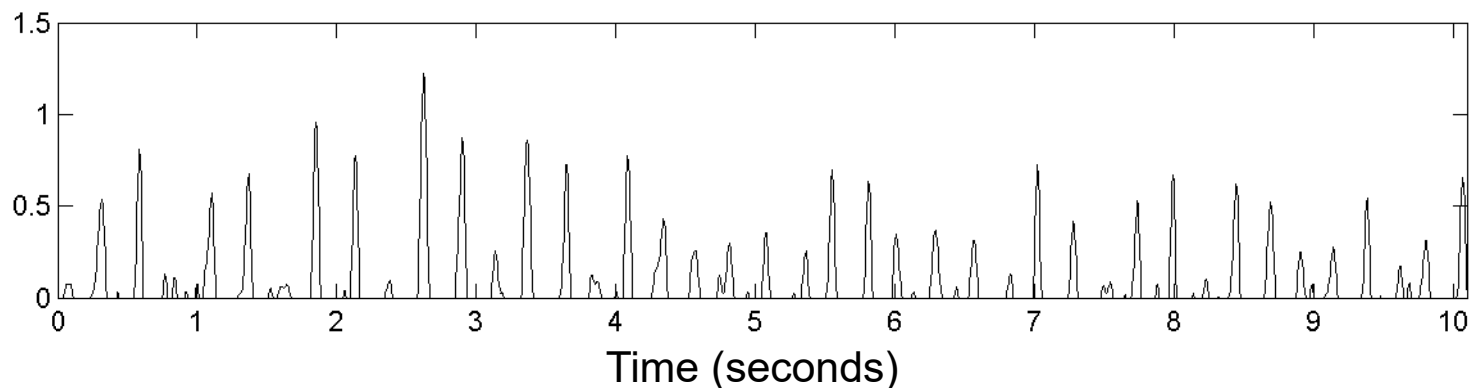


Local Pulse and Tempo Tracking

Fourier temogram (STFT of novelty function)

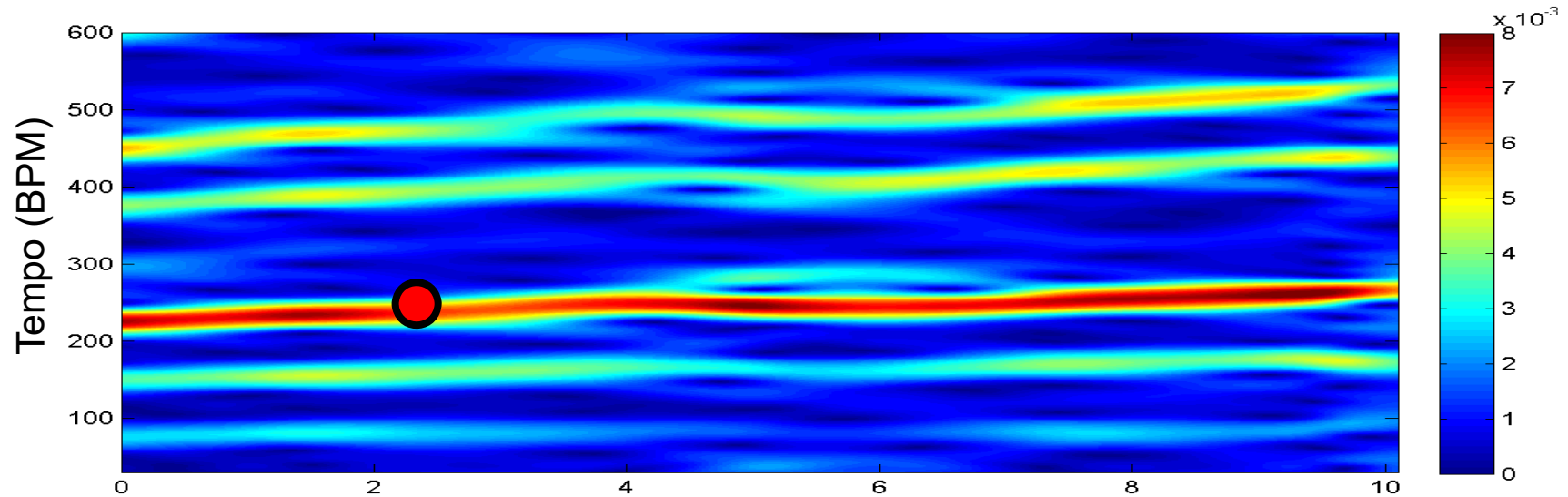


Normalized novelty function

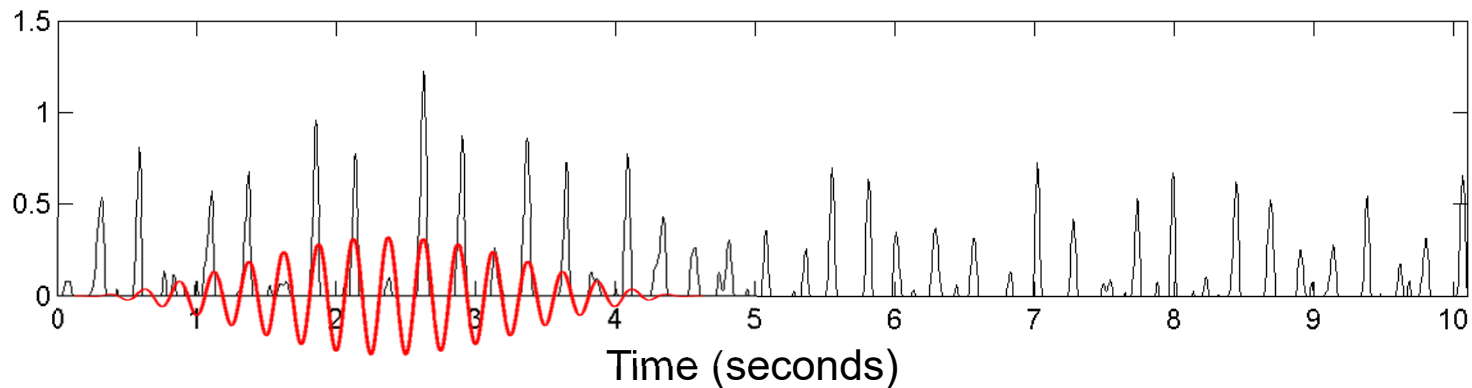


Local Pulse and Tempo Tracking

Fourier temogram (STFT of novelty function)

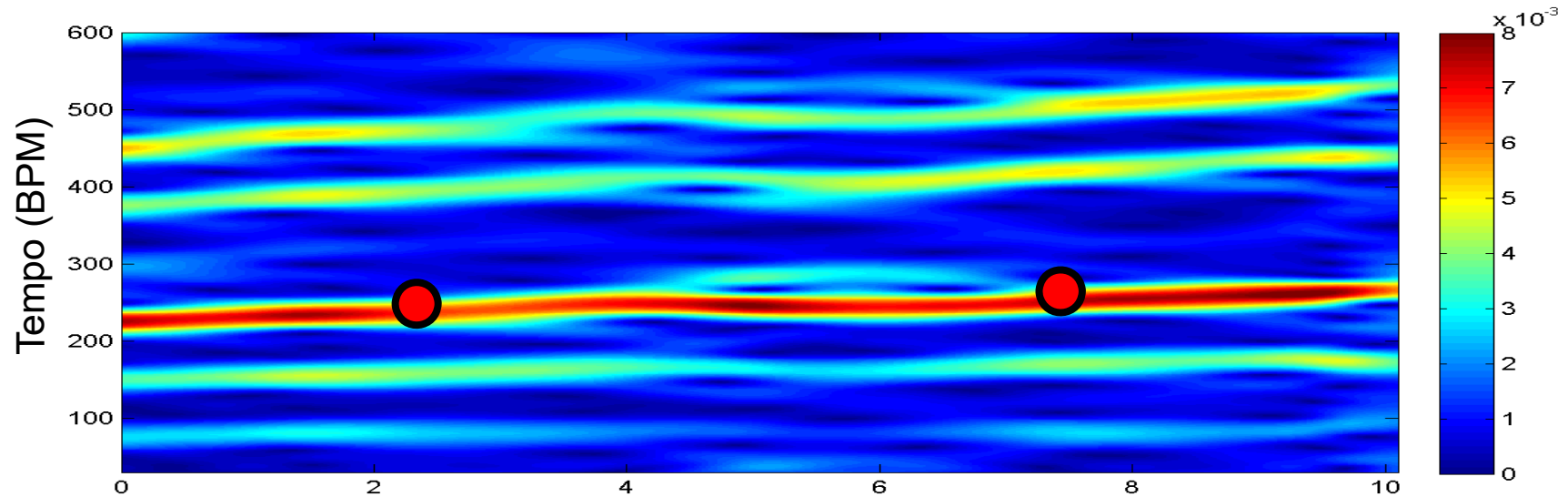


Optimizing local periodicity kernel

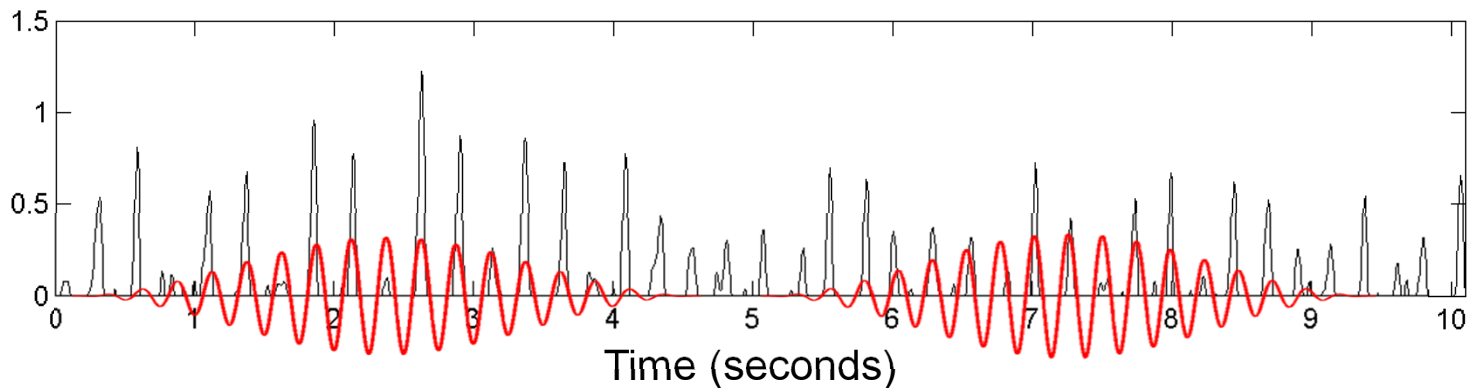


Local Pulse and Tempo Tracking

Fourier temogram (STFT of novelty function)

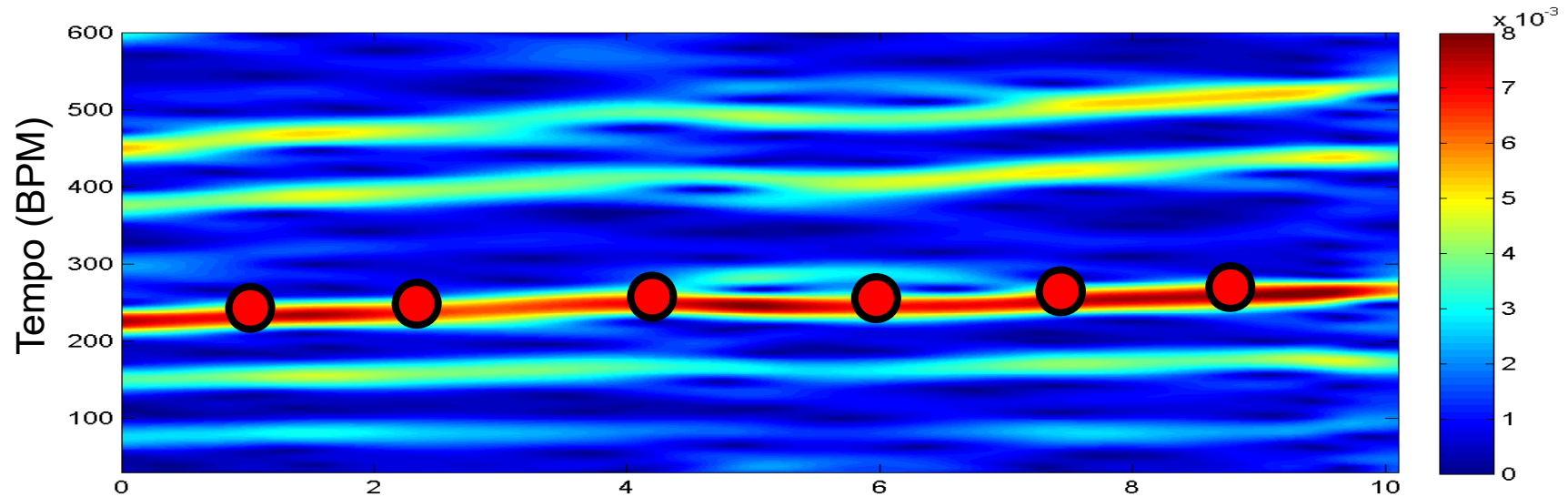


Optimizing local periodicity kernel

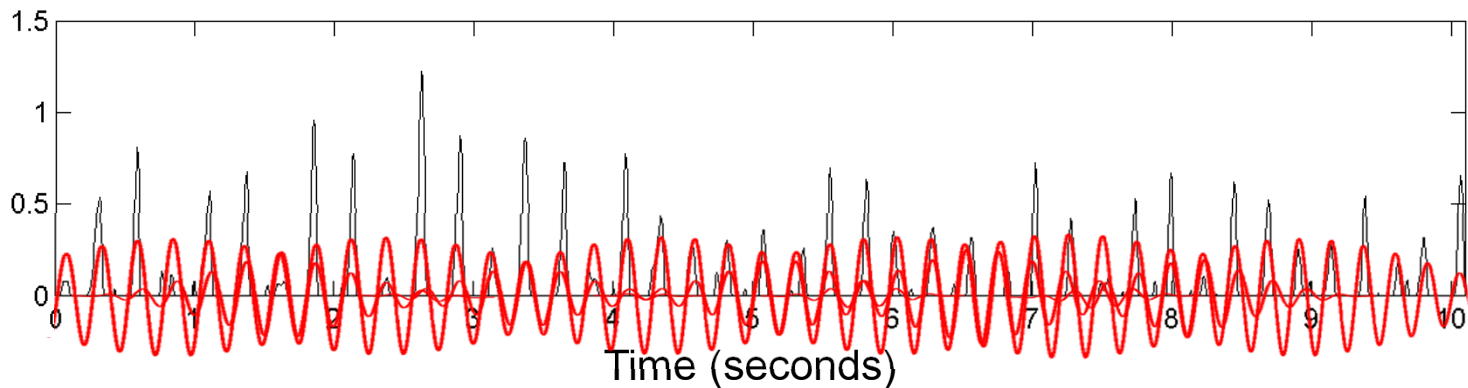


Local Pulse and Tempo Tracking

Fourier temogram (STFT of novelty function)

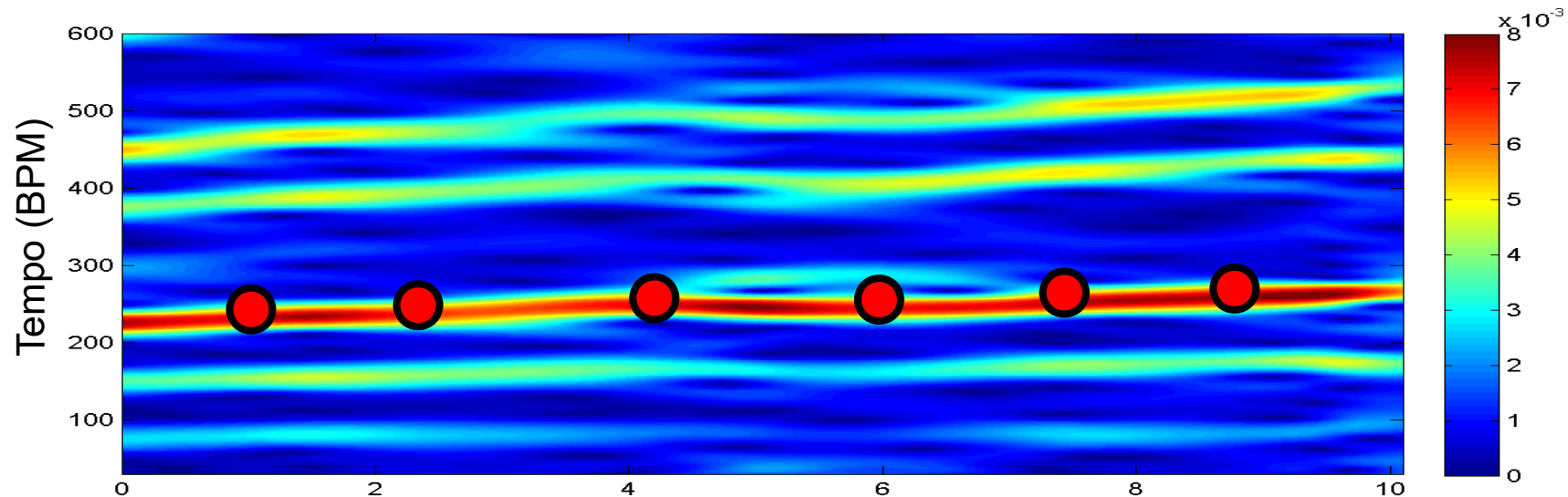


Optimizing local periodicity kernel

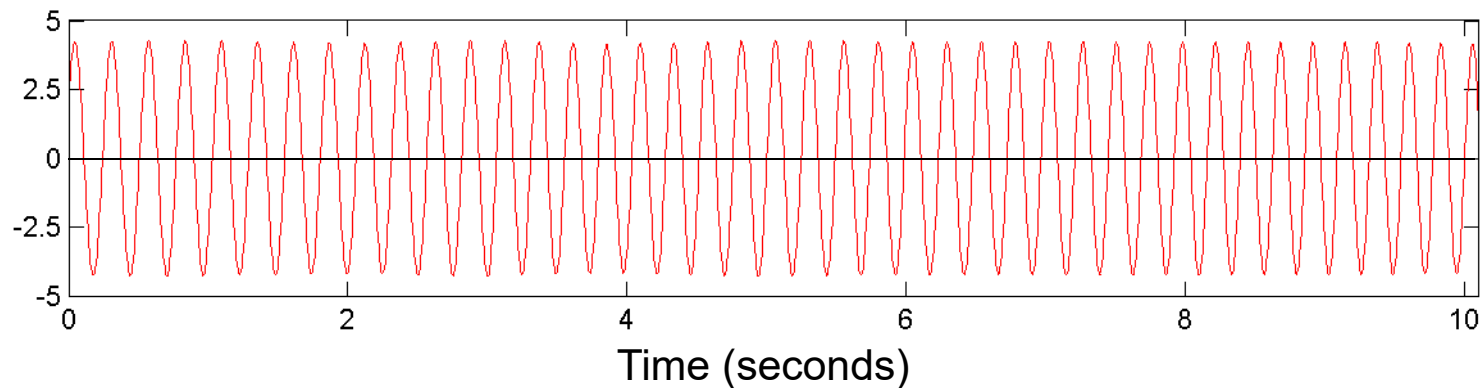


Local Pulse and Tempo Tracking

Fourier temogram (STFT of novelty function)

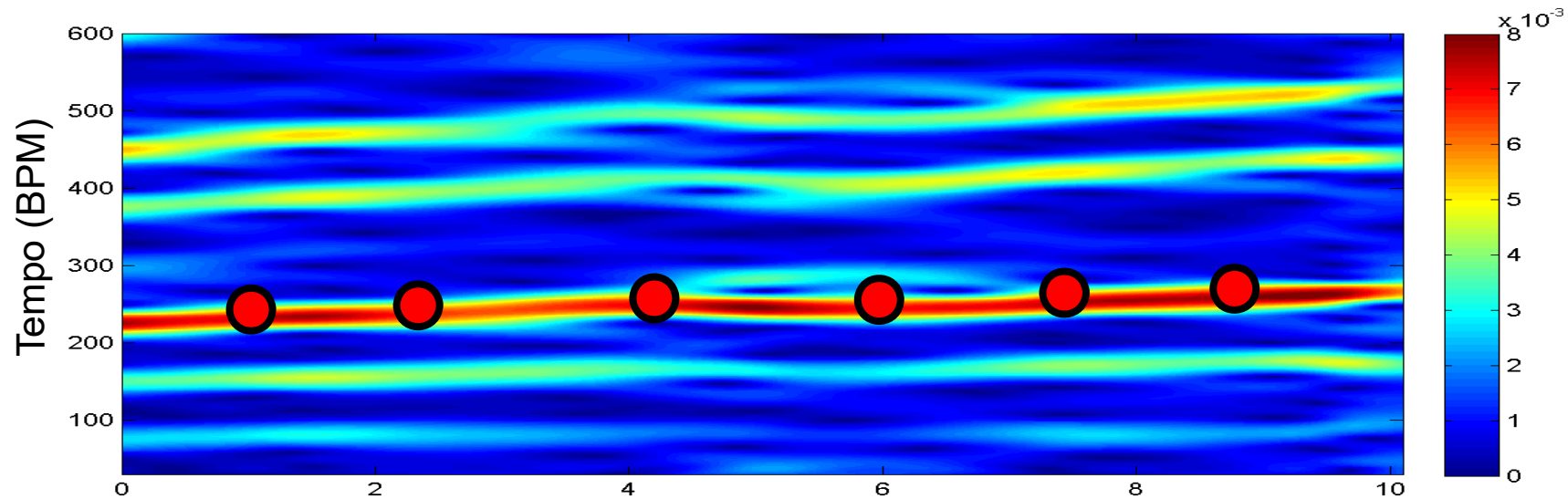


Accumulation of kernels

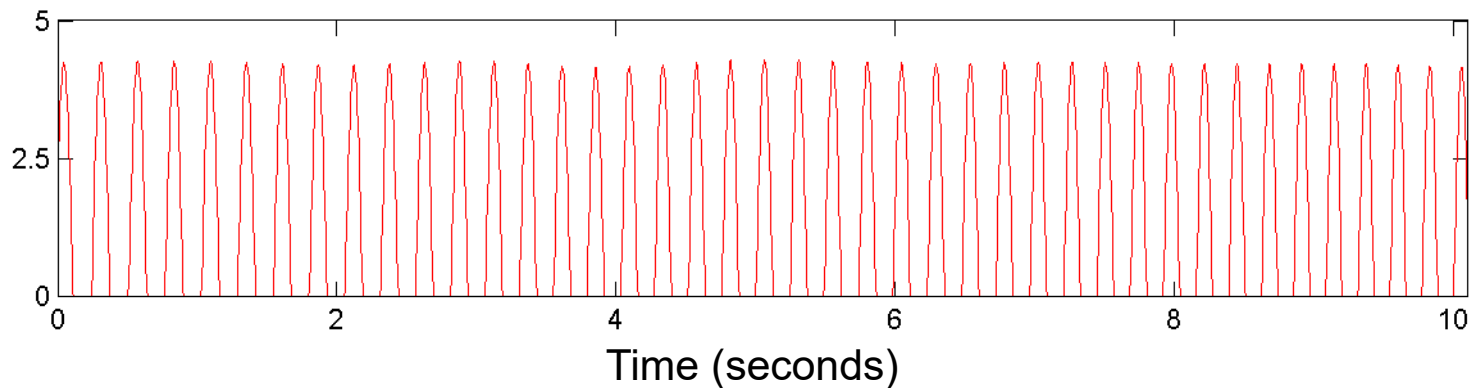


Local Pulse and Tempo Tracking

Fourier temogram (STFT of novelty function)



Halfwave rectification

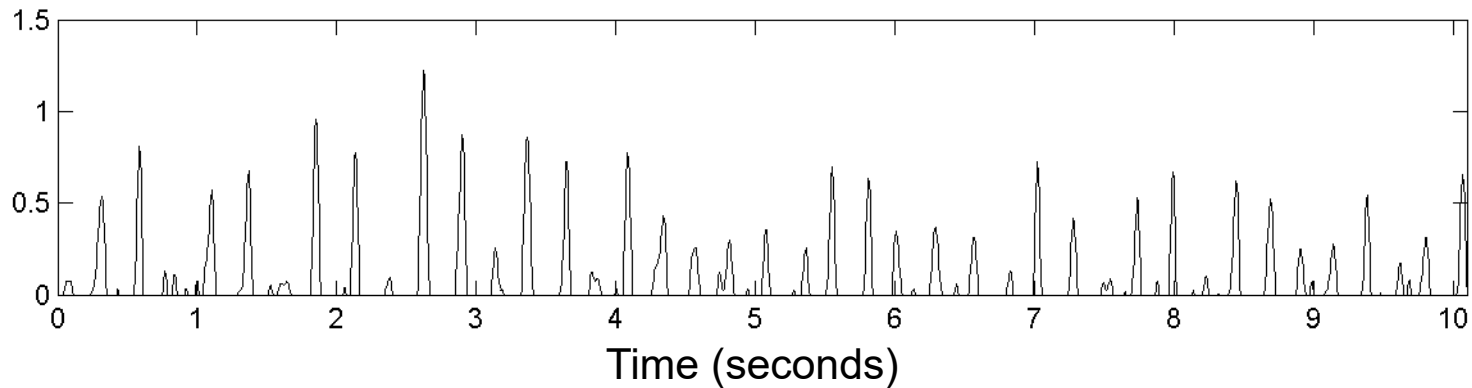


Local Pulse and Tempo Tracking

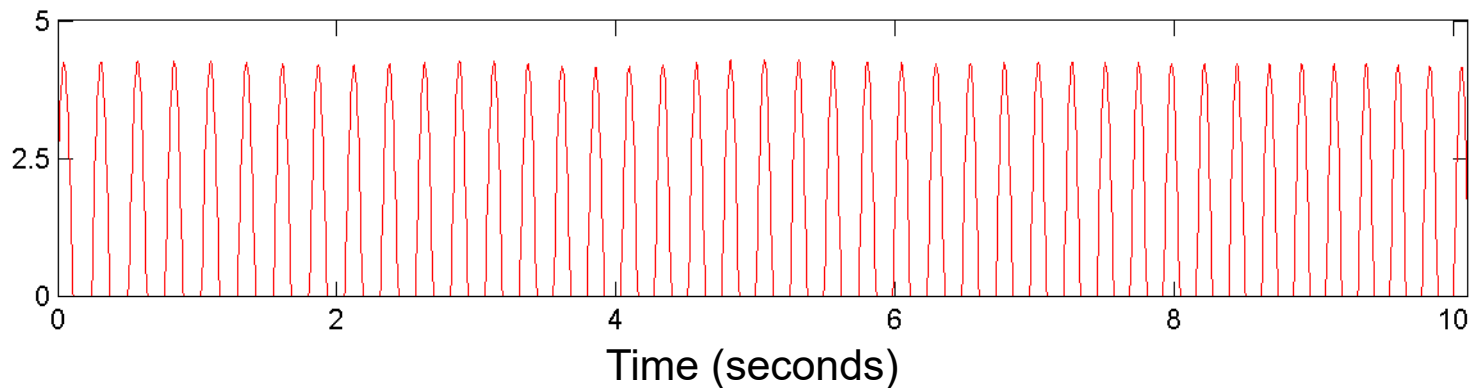
Pulse Tracking

Grosche, Müller: Extracting Predominant Local Pulse Information from Music Recordings. IEEE TASLP 19(6), 2011.

Novelty Curve



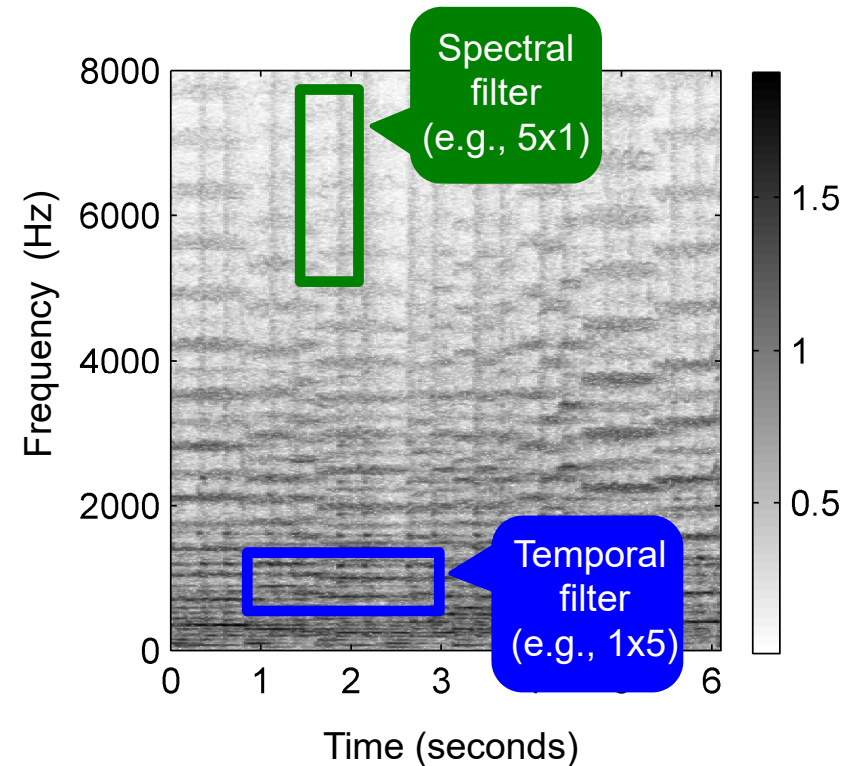
Predominant Local Pulse (PLP)



Local Pulse and Tempo Tracking

Deep Learning Approach

- End-to-end approach
 - Input: Short audio snippets
 - Output: Tempo value
- DL architecture inspired by traditional engineering
 - Layers and activation functions
 - Shape of convolutional kernels

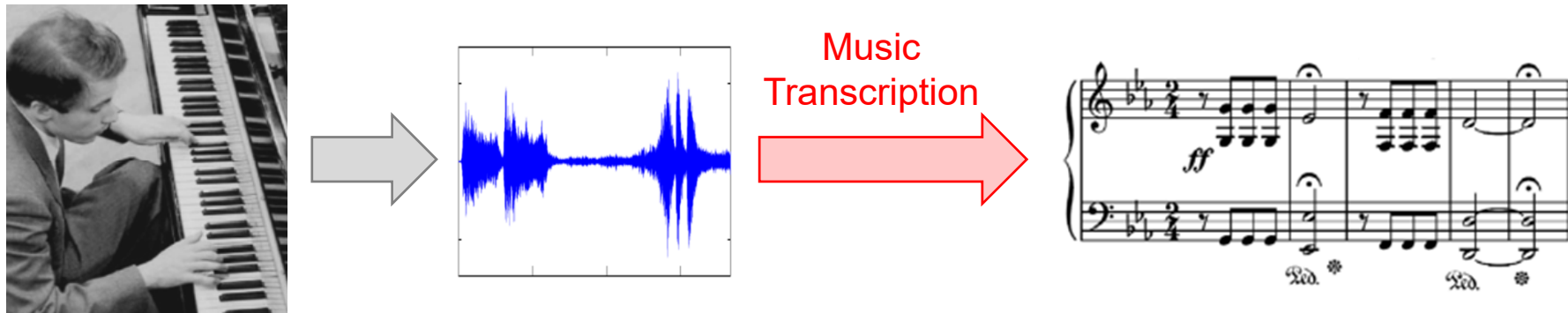


Tempo Estimation

Schreiber, Müller: A Single-Step Approach to Musical Tempo Estimation Using a Convolutional Neural Network, ISMIR, 2018.

Automatic Music Transcription

Task: Convert a music recording into sheet music

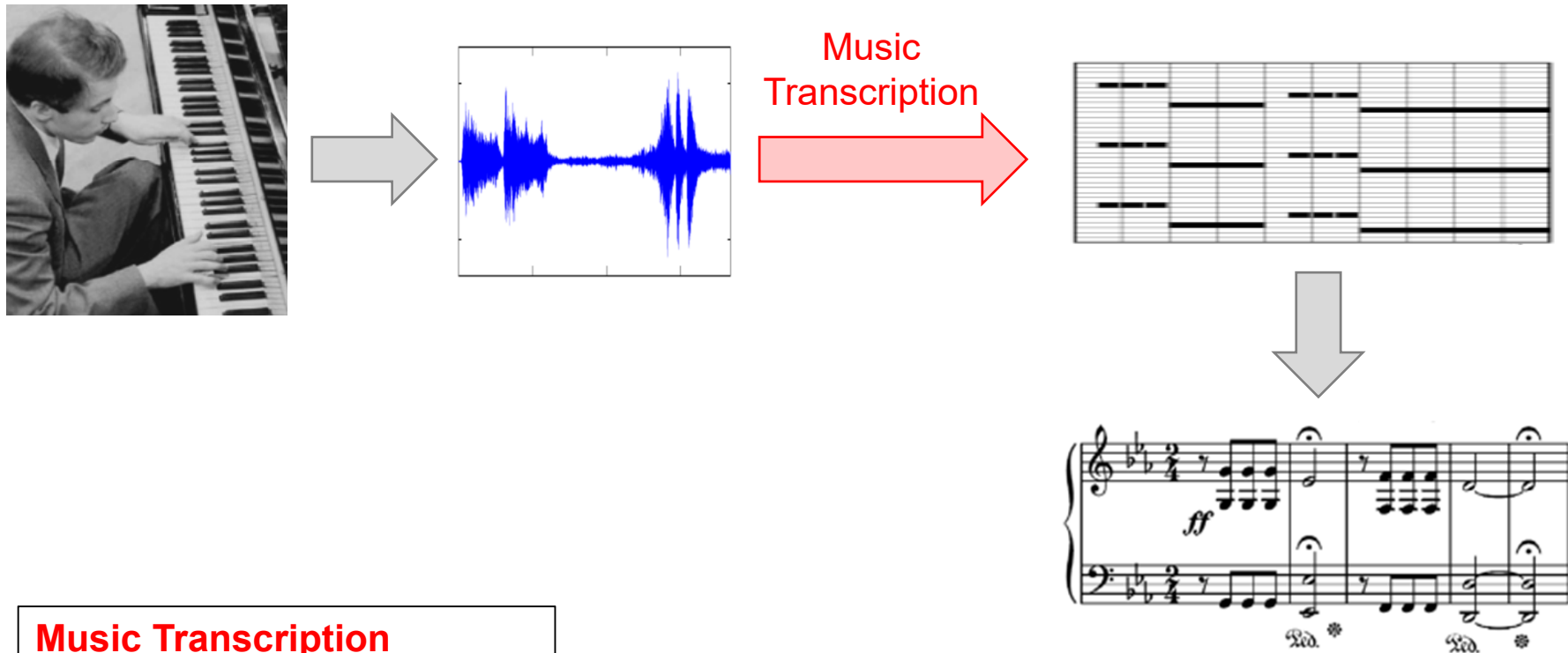


Music Transcription

Benetos et al.: Automatic Music Transcription: An Overview. IEEE Signal Processing Magazine 36(1), 2019.

Automatic Music Transcription

Task: Convert a music recording into sheet music
(or another symbolic music representation)

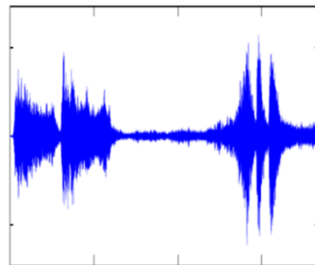
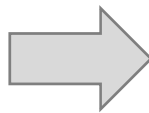


Music Transcription

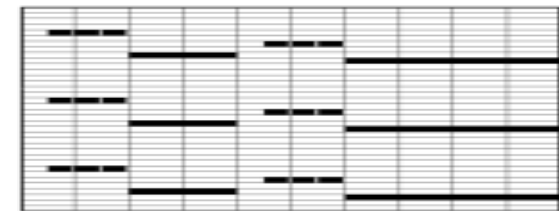
Benetos et al.: Automatic Music Transcription: An Overview. IEEE Signal Processing Magazine 36(1), 2019.

Automatic Music Transcription

Task: Convert a music recording into sheet music
(or another symbolic music representation)



Music
Transcription



Multitask Learning for estimating

- pitches,
- note onsets & offsets,
- beat & measure positions,
- musical voices & instrumentation,
- pedalling, dynamics, ...

Why is Music Processing Challenging?

Example: Chopin, Mazurka Op. 63 No. 3



Mazurka.

F. CHOPIN. Op. 63, No. 3.

Allegretto.

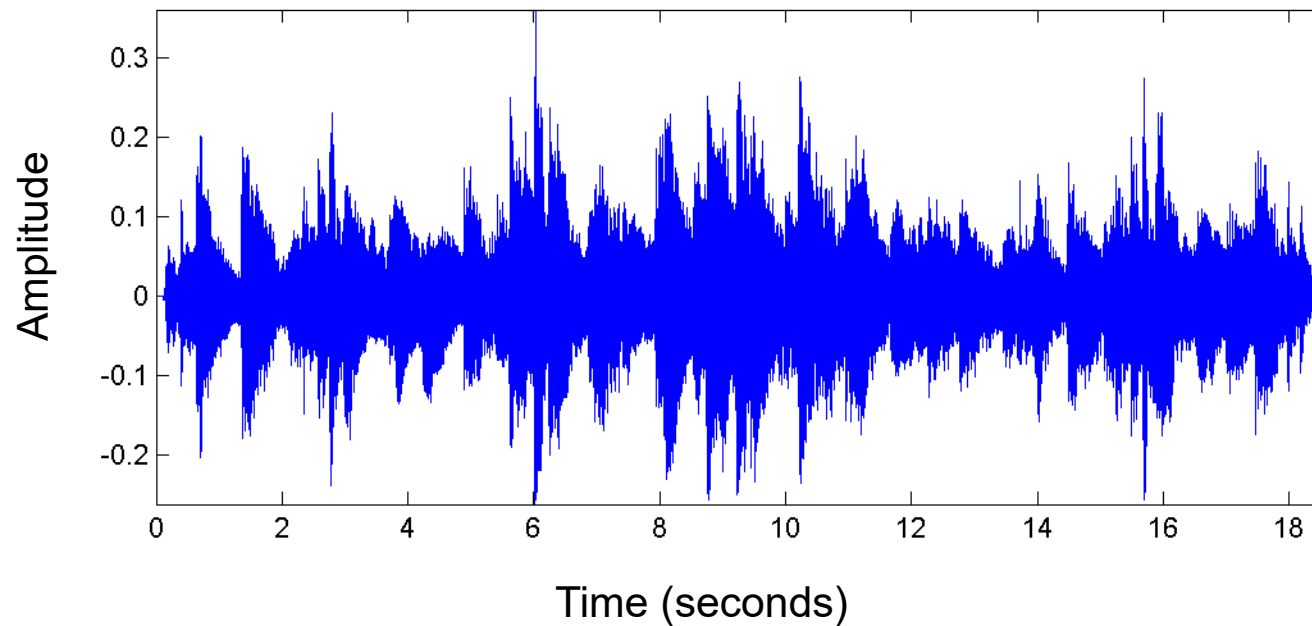
41. *p*

The image shows a musical score for Chopin's Mazurka Op. 63 No. 3, measures 41-50. The score is in 3/4 time, key of D major, and marked 'Allegretto'. It features a treble and bass clef. The right hand (treble clef) has a melody with triplets and slurs. The left hand (bass clef) has a rhythmic accompaniment with chords and triplets. The score includes dynamic markings like 'p' and 'p_{ed.}' (pedal), and fingering numbers (1, 2, 3, 4). There are asterisks under some notes in the bass line, possibly indicating specific processing points or annotations.

Why is Music Processing Challenging?

Example: Chopin, Mazurka Op. 63 No. 3

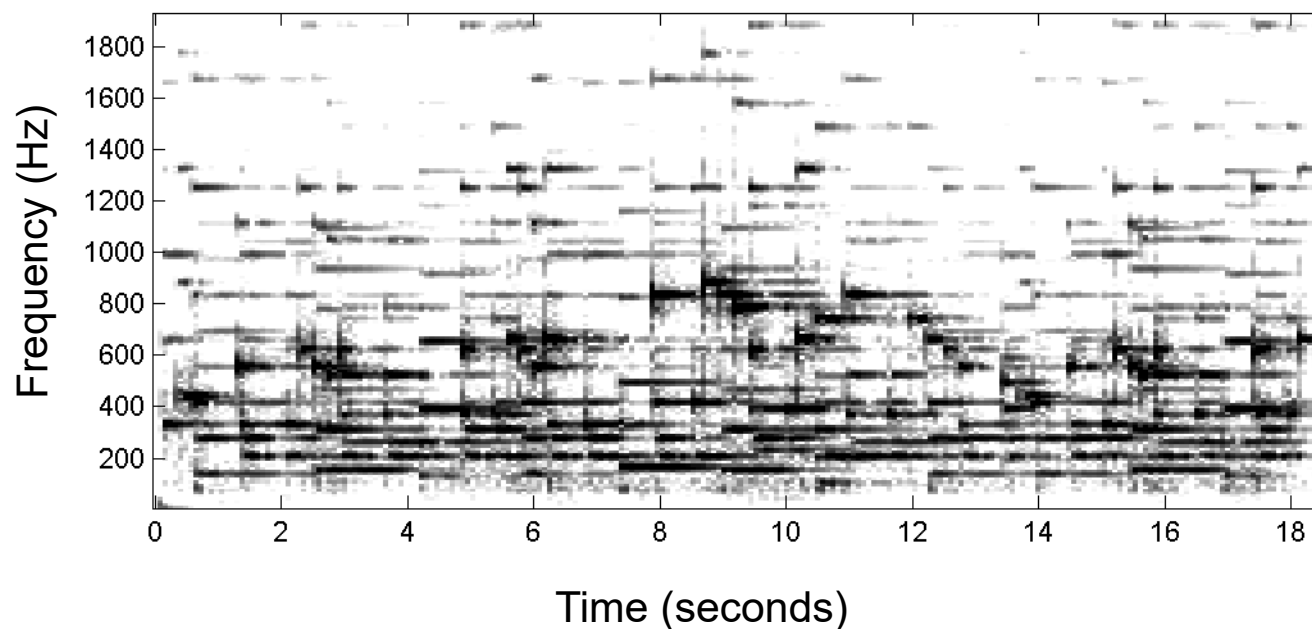
- Waveform



Why is Music Processing Challenging?

Example: Chopin, Mazurka Op. 63 No. 3

- Waveform / Spectrogram



Why is Music Processing Challenging?

Example: Chopin, Mazurka Op. 63 No. 3

- Waveform / Spectrogram
- Performance
 - Tempo
 - Dynamics
 - Note deviations
 - Sustain pedal

Why is Music Processing Challenging?

Example: Chopin, Mazurka Op. 63 No. 3



- Waveform / Spectrogram
- Performance
 - Tempo
 - Dynamics
 - Note deviations
 - Sustain pedal
- Polyphony

A musical score for Chopin's Mazurka Op. 63 No. 3, showing two systems of piano music. The score is annotated with performance information: blue highlights on the upper staff indicate the main melody, red highlights on the lower staff indicate an additional melody line, and yellow highlights on the lower staff indicate the accompaniment. Fingerings and dynamics like 'p' and 'f' are also visible.

 **Main Melody**
 **Additional melody line**
 **Accompaniment**

Source Separation

- Decomposition of audio stream into different sound sources
- Central task in digital signal processing
- “Cocktail party effect”

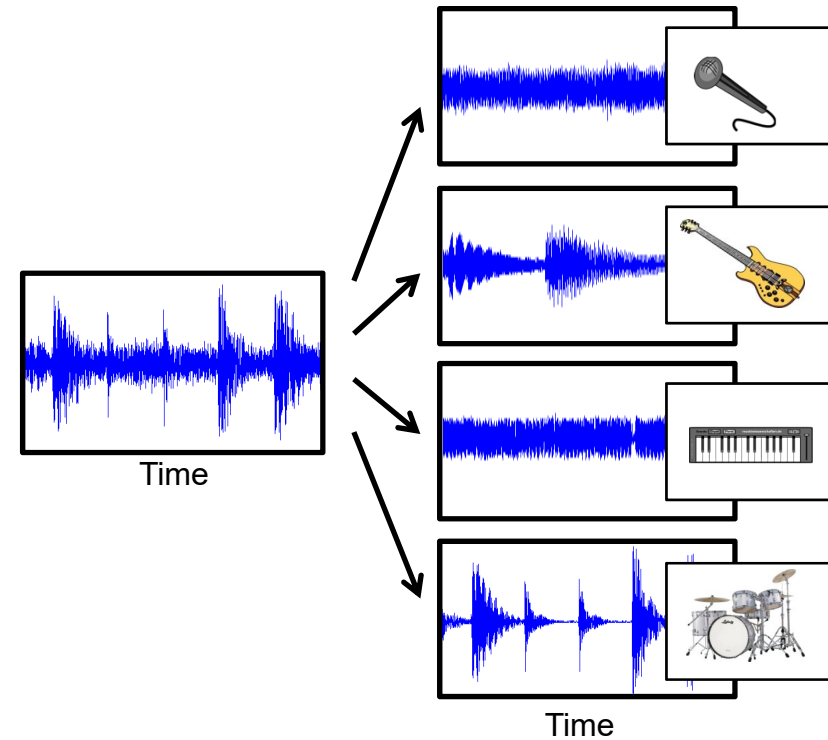


Source Separation

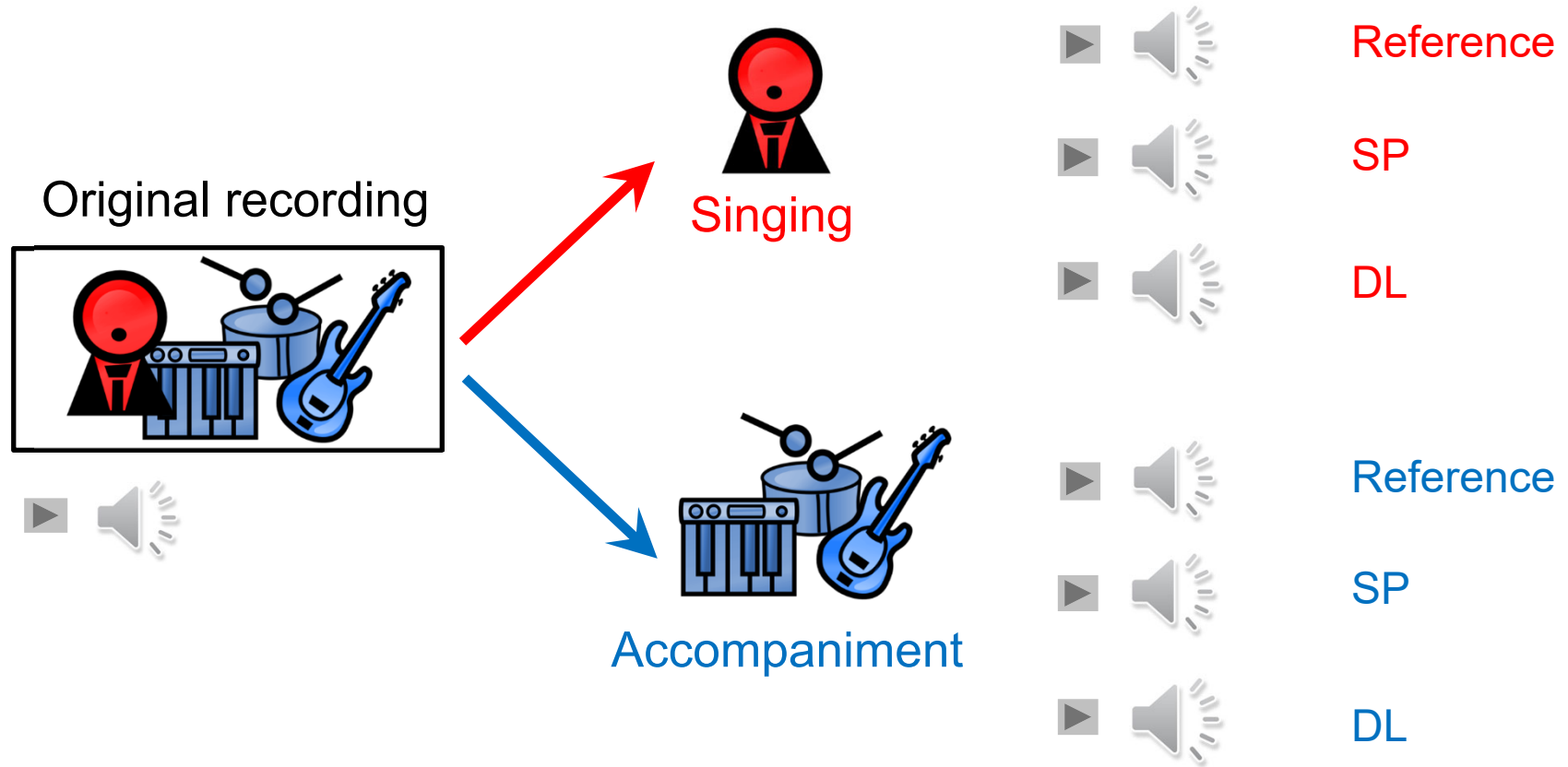
- Decomposition of audio stream into different sound sources
- Central task in digital signal processing
- “Cocktail party effect”
- Several input signals
- Sources are assumed to be statistically independent

Source Separation (Music)

- Main melody, accompaniment, drum track
- Instrumental voices
- Individual note events
- Only mono or stereo
- Sources are often highly dependent



Source Separation (Singing Voice)



DL-Based Source Separation

Stöter, Uhlich Luitkus, Mitsufuji: Open-Unmix – A Reference Implementation for Music Source Separation. JOSS, 2019.

- Reference: Best possible result
- SP: Traditional signal processing
- DL: Deep Learning

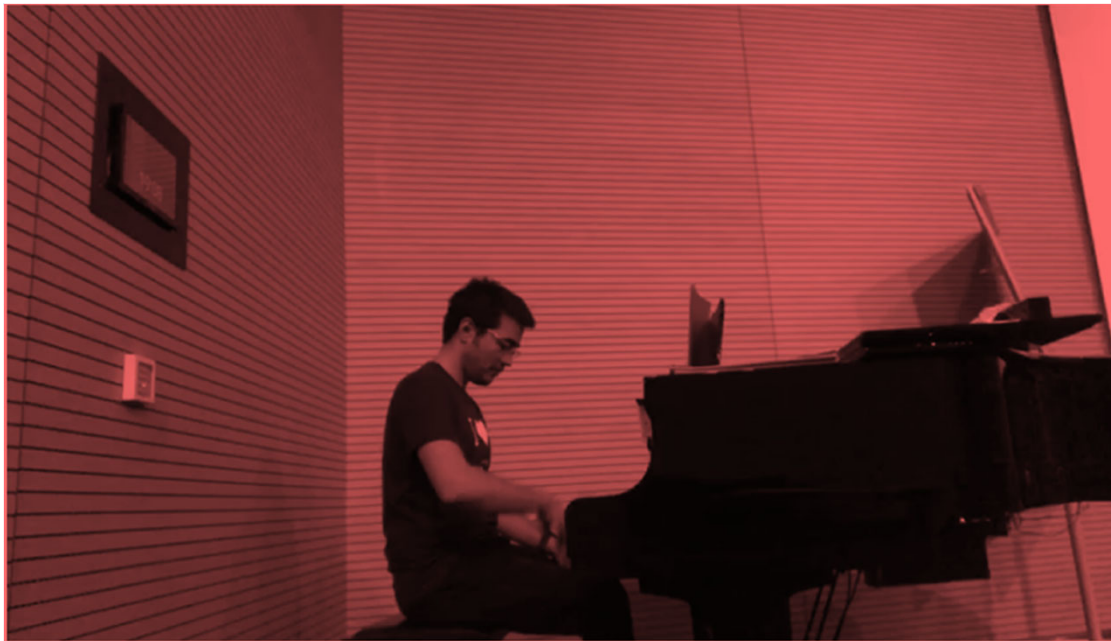
Source Separation (Piano Concerto)

- Yigitcan Özer
- PhD student in engineering
- Pianist



Source Separation (Piano Concerto)

- Yigitcan Özer
- PhD student in engineering
- Pianist



Only Piano!



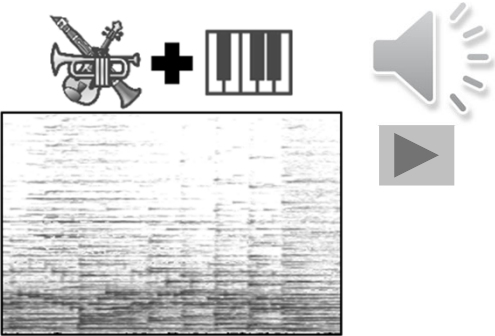
**Where is the
orchestra?**



Source Separation (Piano Concerto)



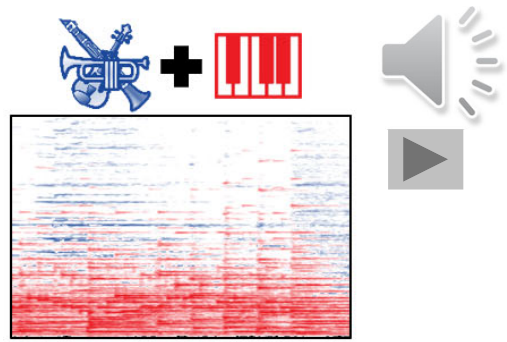
A musical score for a piano concerto, starting at measure 89. The score is written for a full orchestra and piano. The instruments listed on the left are: Flute, Clarinet, Trumpet, Trombone, Horn, Violin I, Violin II, Viola, and Cello. The piano part is shown in the bottom two staves. The score includes various musical notations such as notes, rests, and dynamic markings like 'p' (piano).



A diagram illustrating the source separation process. It shows a plus sign between a trumpet icon and a piano keyboard icon. Below this, a spectrogram displays the frequency spectrum of the audio. To the right of the spectrogram are a speaker icon and a play button icon, indicating the output of the process.

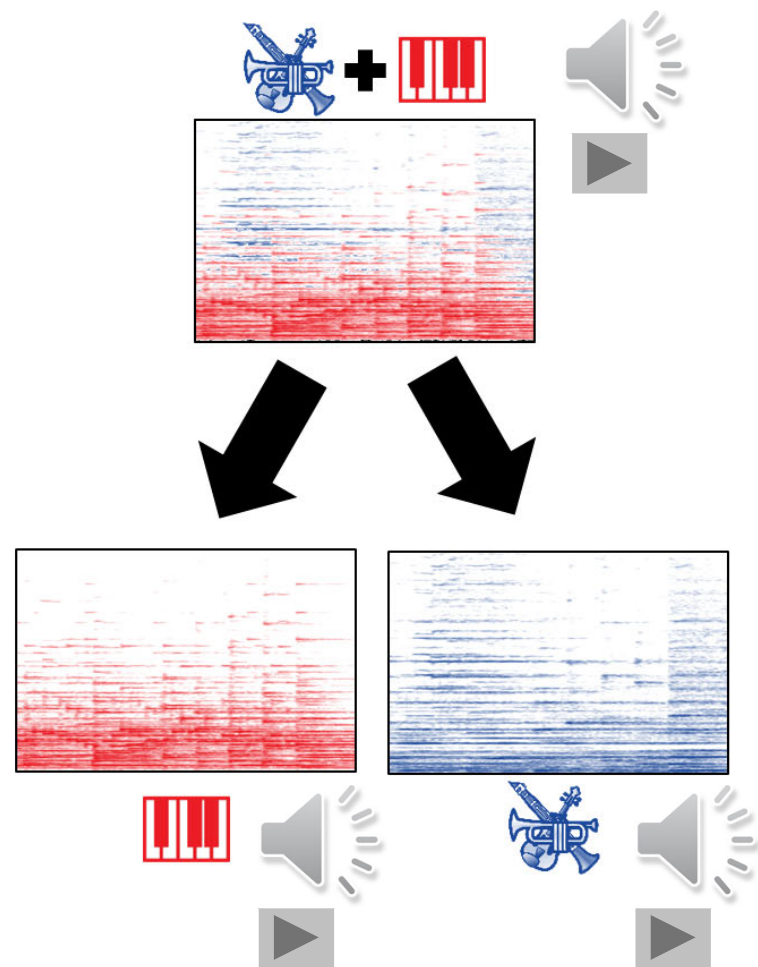
Source Separation (Piano Concerto)

A musical score for a piano concerto, starting at measure 89. The score is divided into two main sections. The upper section, in blue, includes staves for woodwinds (flute, oboe, clarinet, bassoon), brass (trumpet, trombone, horn, tuba), and strings (violin, viola, cello, double bass). The lower section, in red, is the piano part, featuring a grand staff with treble and bass clefs. A red piano keyboard icon is positioned to the left of the piano part. The piano part shows a complex melodic line in the right hand and a supporting bass line in the left hand.

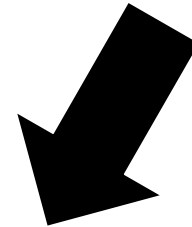
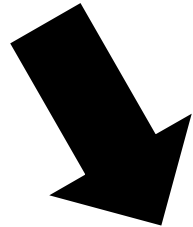
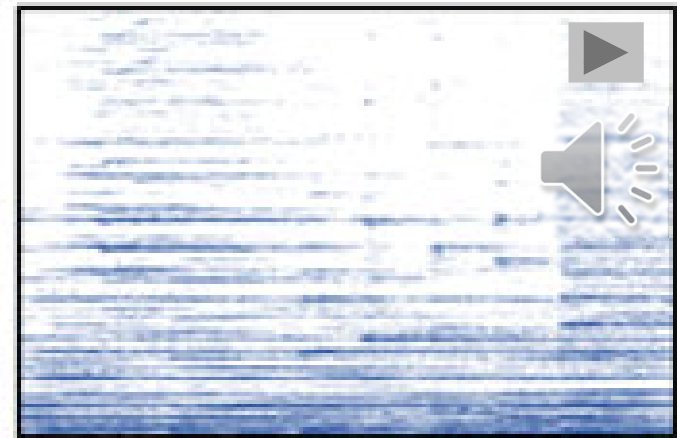
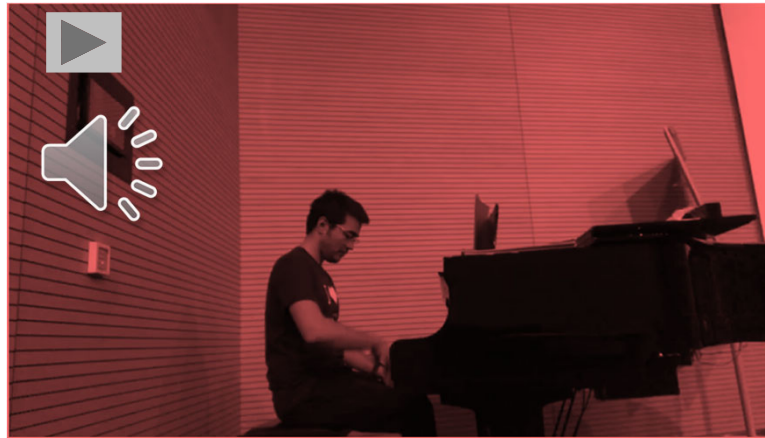


Source Separation (Piano Concerto)

A musical score for a piano concerto, starting at measure 89. The score is divided into two main sections. The upper section, from measure 89 to the end of the page, is written in blue ink and includes staves for woodwinds (flute, oboe, clarinet, bassoon), brass (trumpet, trombone, horn), and strings (violin, viola, cello, double bass). The lower section, from measure 89 to the end of the page, is written in red ink and is specifically for the piano. A red piano keyboard icon is placed to the left of the piano staves. The piano part features a complex, fast-moving melodic line in the right hand and a supporting bass line in the left hand.

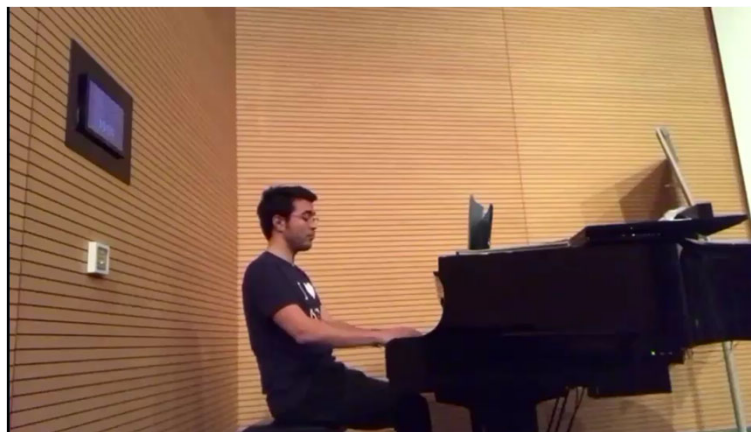


Source Separation (Piano Concerto)



Piano Source Separation

Özer, Müller: Source Separation of Piano Concertos with Test-Time Adaptation, ISMIR, 2022.



Score-Informed Source Separation

Exploit musical score to support decomposition process

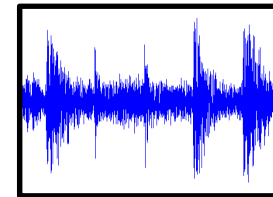
Prior Knowledge

Ewert, Pardo, Müller, Plumbley:
Score-Informed Source Separation
for Musical Audio Recordings.
IEEE SPM 31(3), 2014.

Musical
Information



Audio
Signal

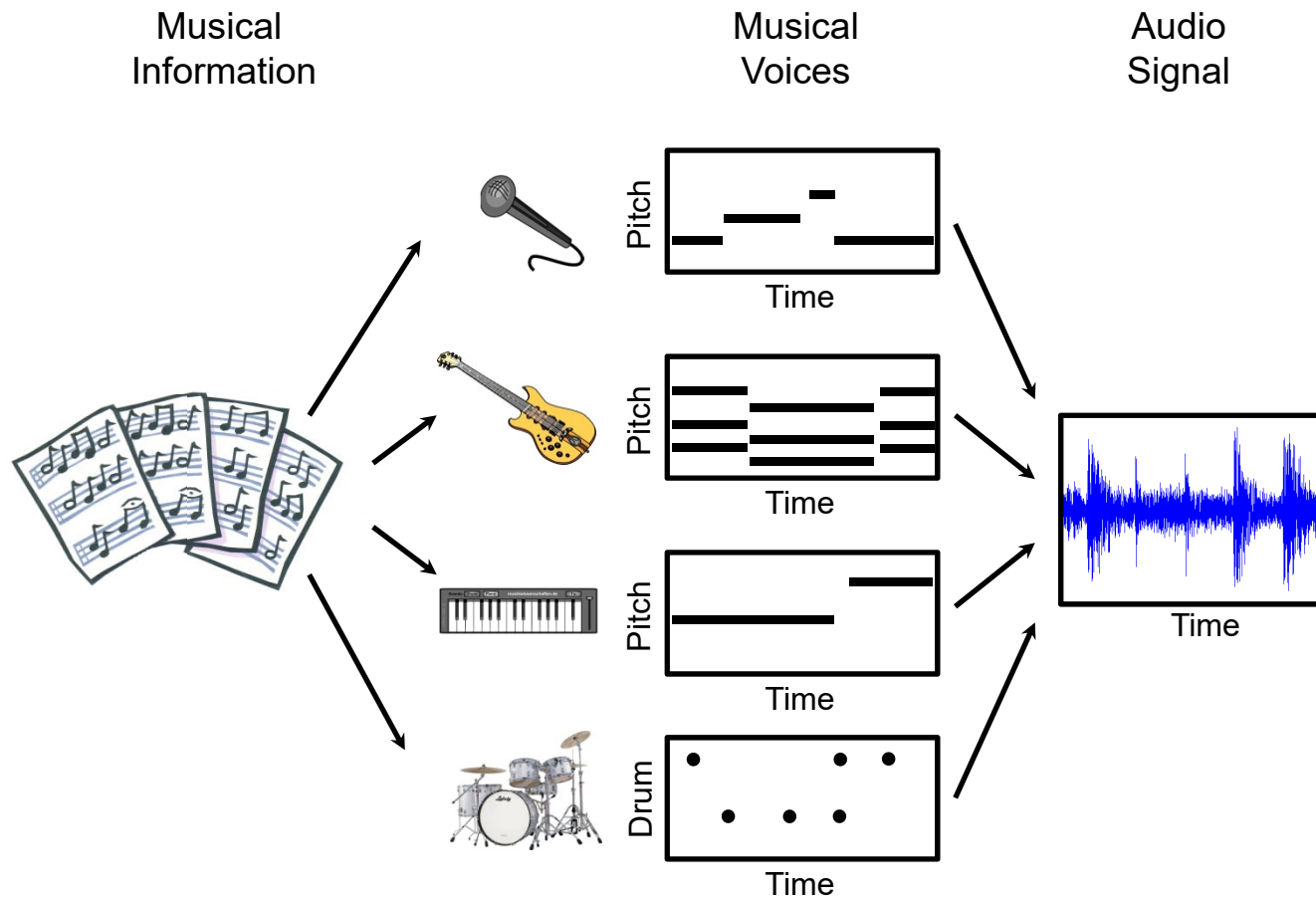


Time

Score-Informed Source Separation

Exploit musical score to support decomposition process

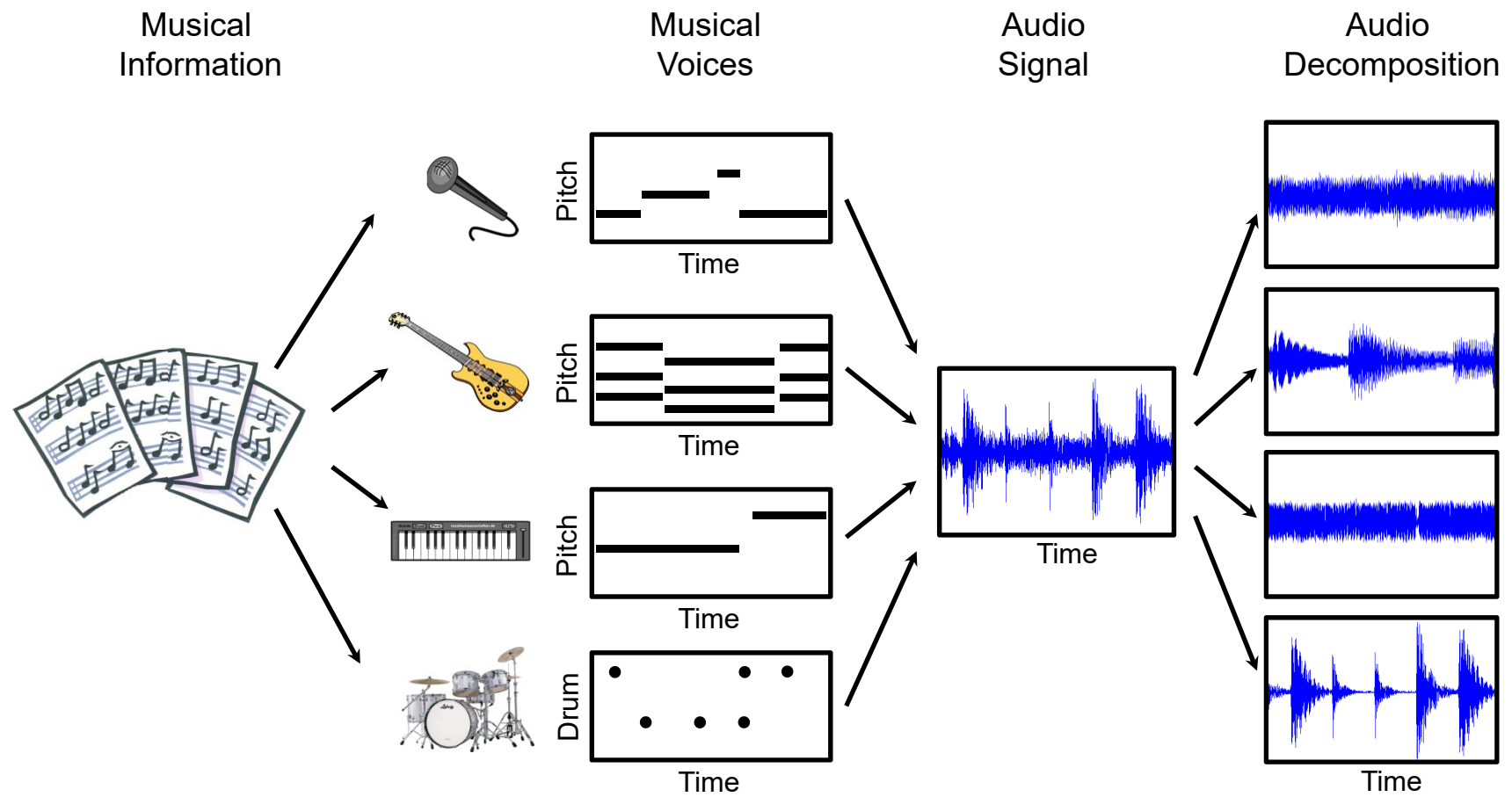
Prior Knowledge
Ewert, Pardo, Müller, Plumbley:
Score-Informed Source Separation
for Musical Audio Recordings.
IEEE SPM 31(3), 2014.



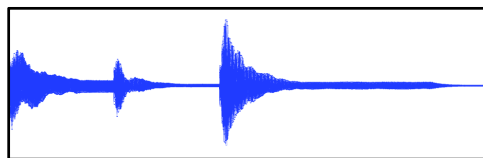
Score-Informed Source Separation

Exploit musical score to support decomposition process

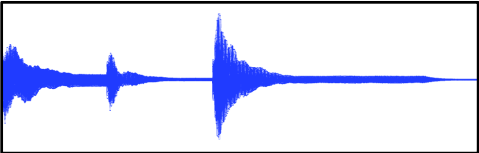
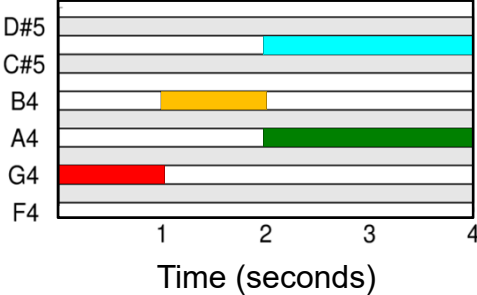
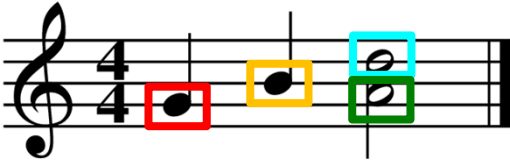
Prior Knowledge
Ewert, Pardo, Müller, Plumbley:
Score-Informed Source Separation
for Musical Audio Recordings.
IEEE SPM 31(3), 2014.



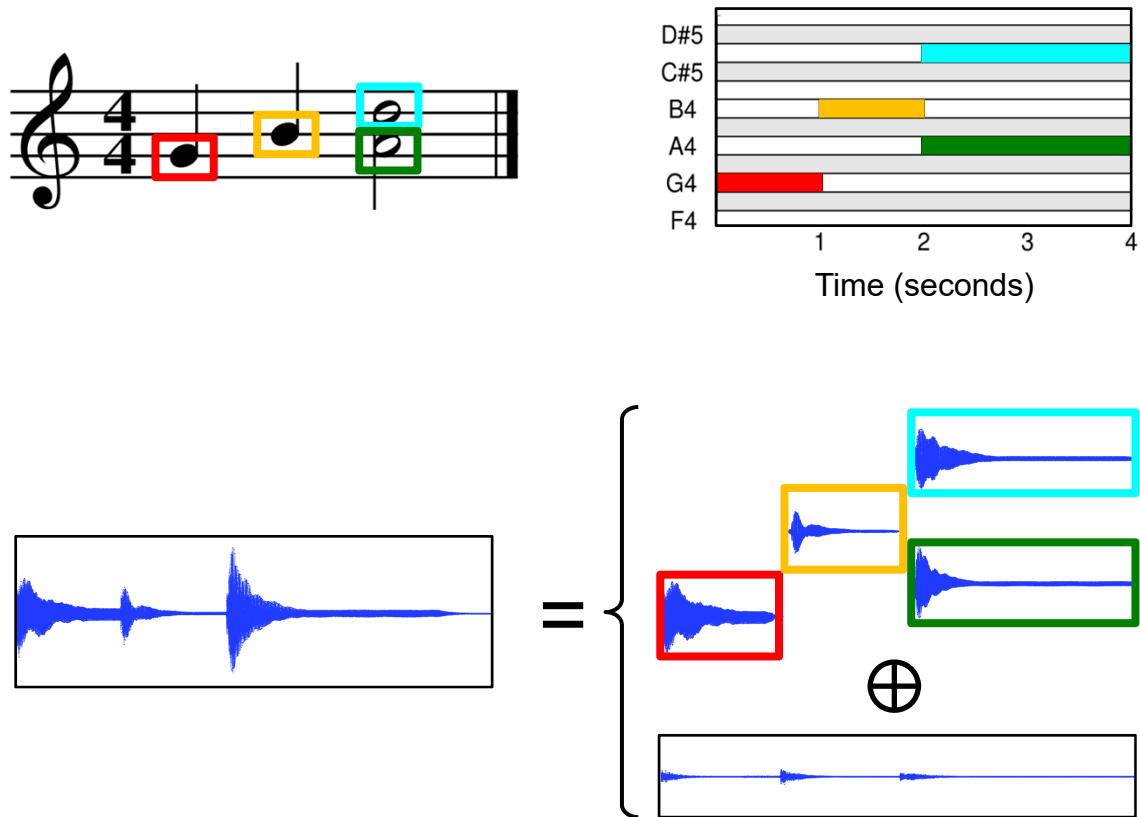
Score-Informed Audio Decomposition



Score-Informed Audio Decomposition

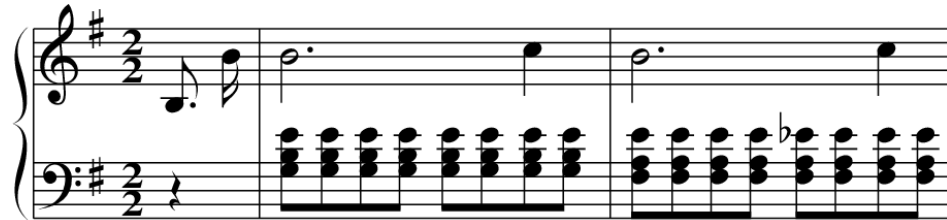


Score-Informed Audio Decomposition

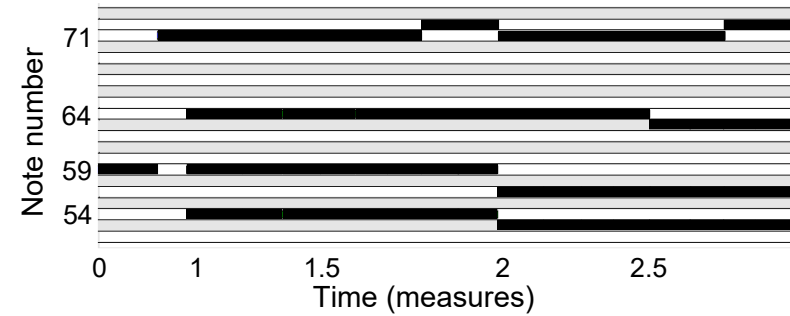


Score-Informed Audio Decomposition

Sheet music

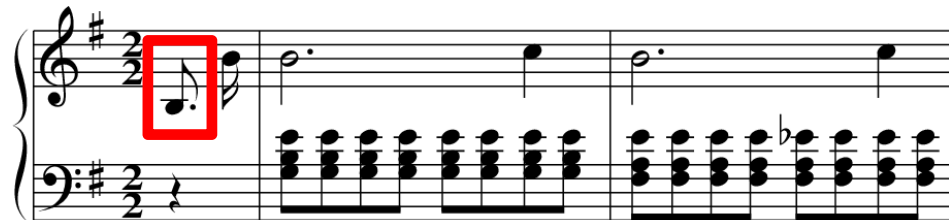


Piano roll



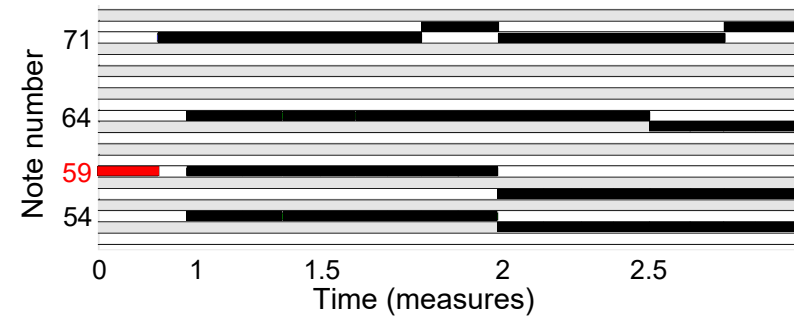
Score-Informed Audio Decomposition

Sheet music



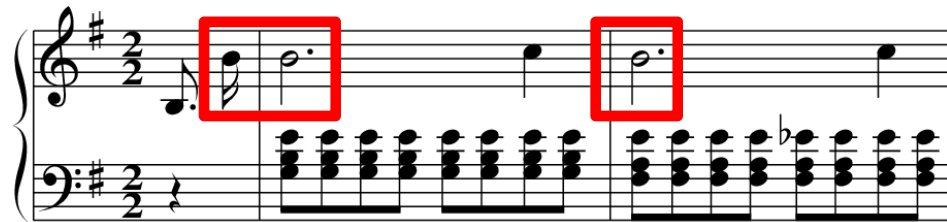
$p = 59$

Piano roll



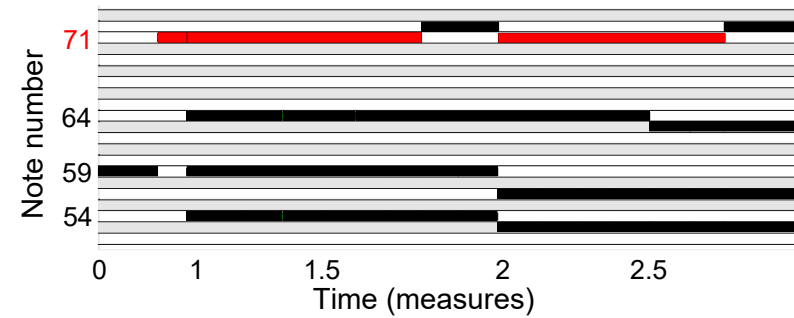
Score-Informed Audio Decomposition

Sheet music



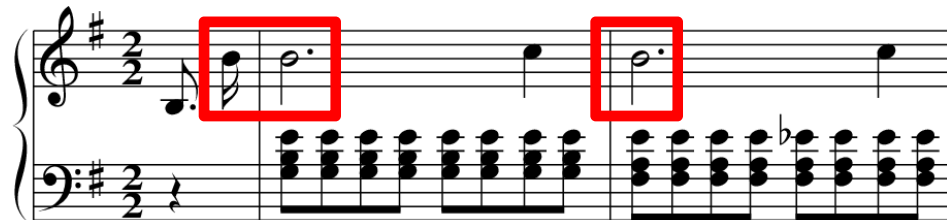
$p = 71$

Piano roll



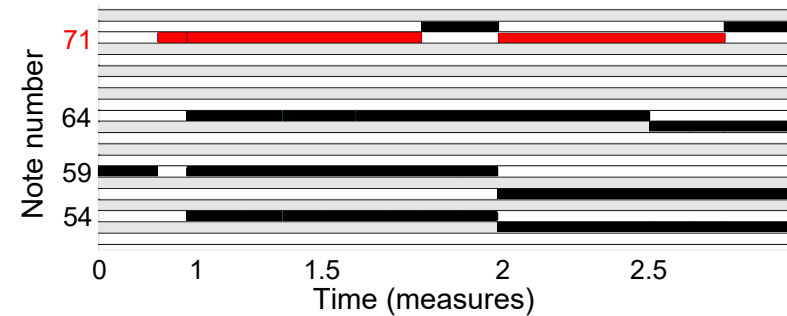
Score-Informed Audio Decomposition

Sheet music

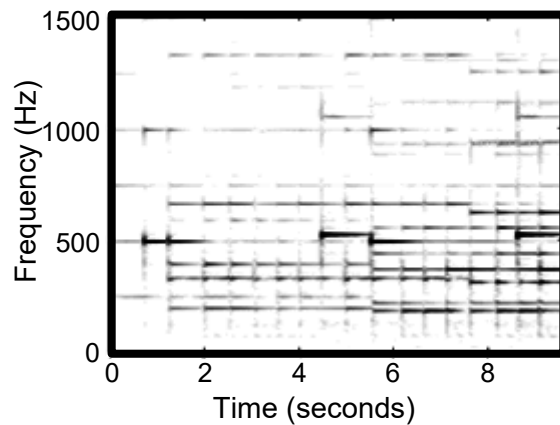


$p = 71$

Piano roll

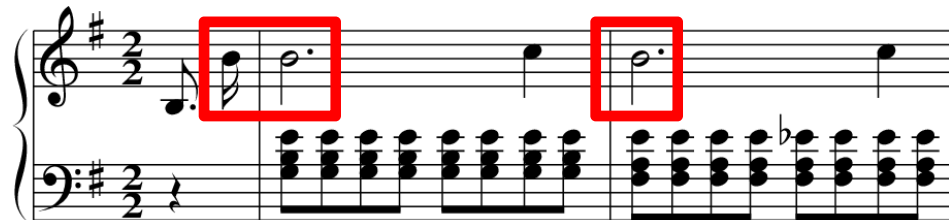


Spectrogram



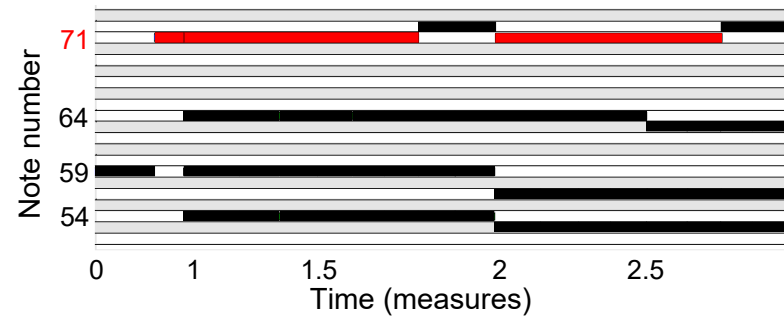
Score-Informed Audio Decomposition

Sheet music

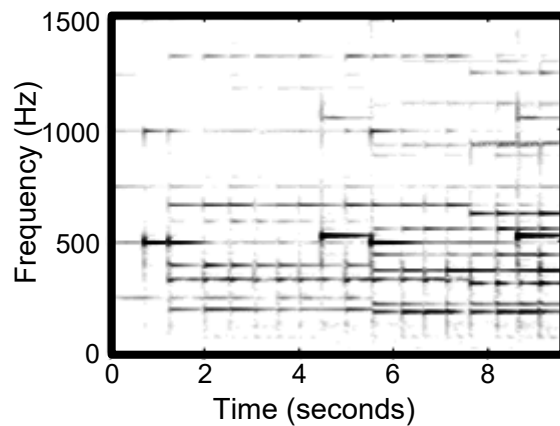


$p = 71$

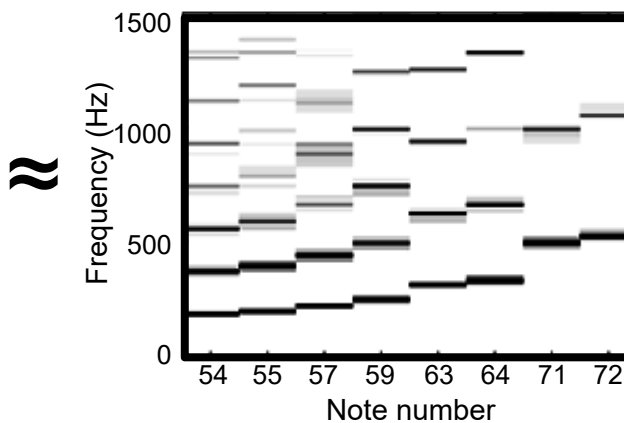
Piano roll



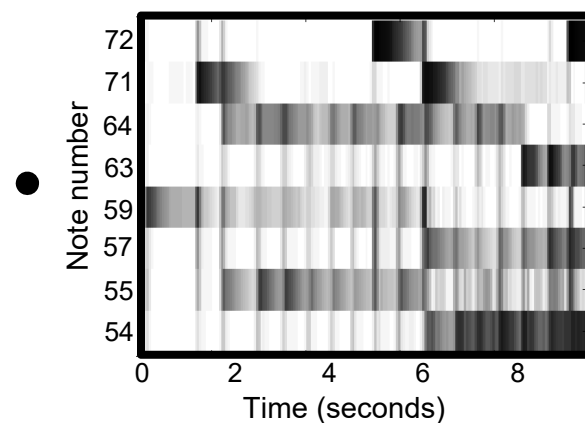
Spectrogram



Spectral patterns

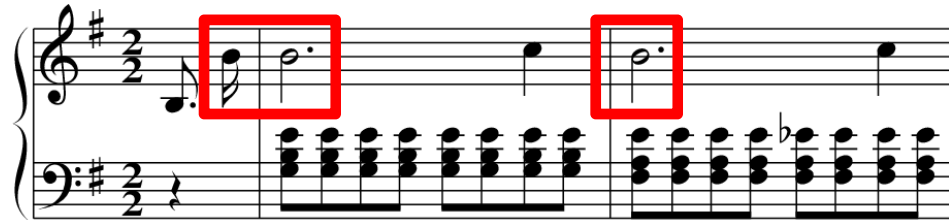


Activity patterns



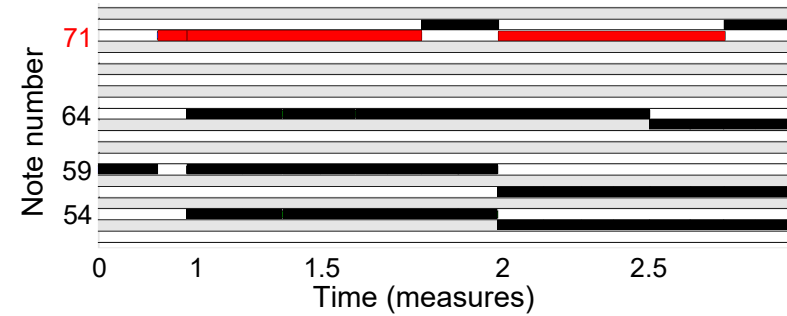
Score-Informed Audio Decomposition

Sheet music

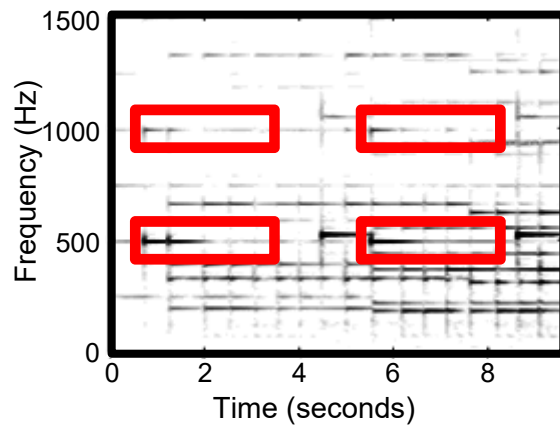


$p = 71$

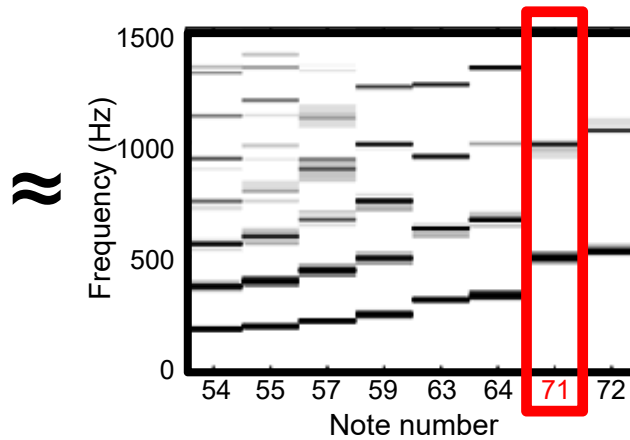
Piano roll



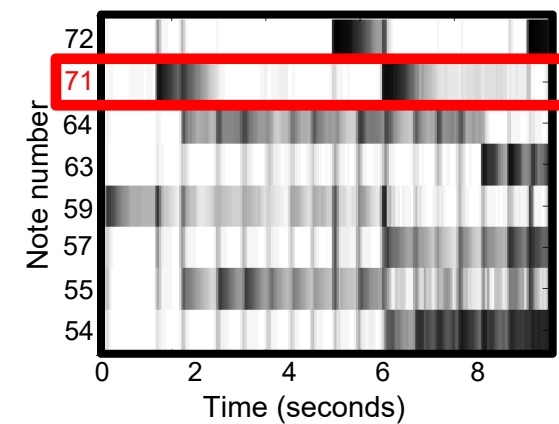
Spectrogram



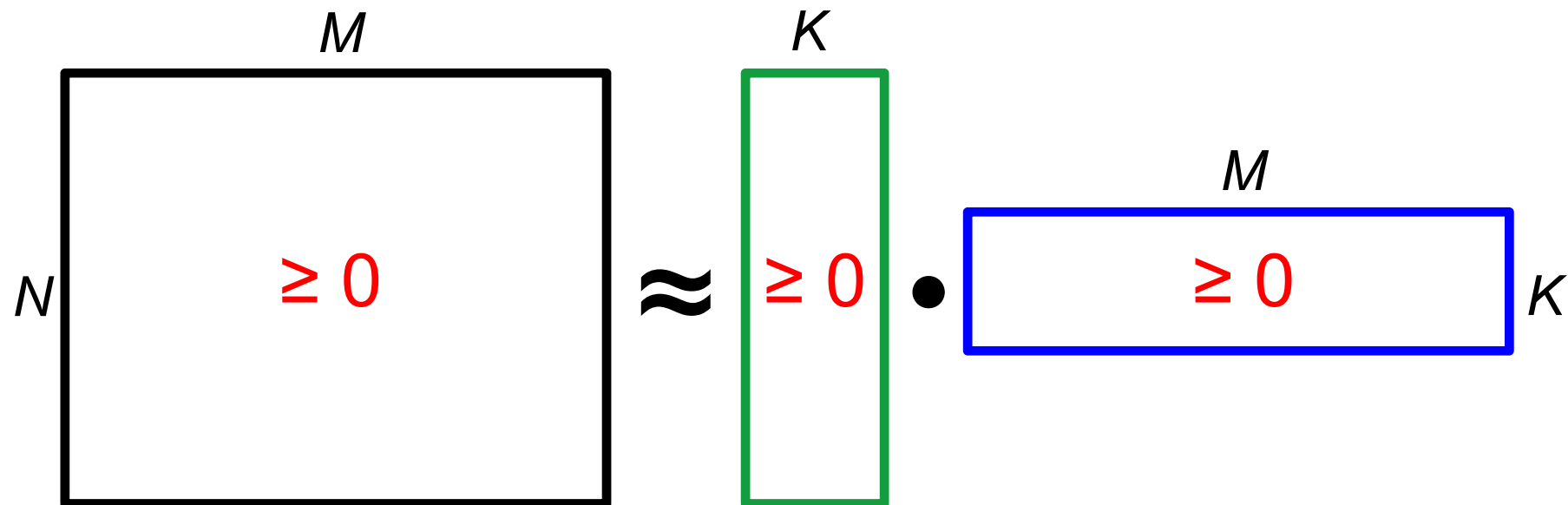
Spectral patterns



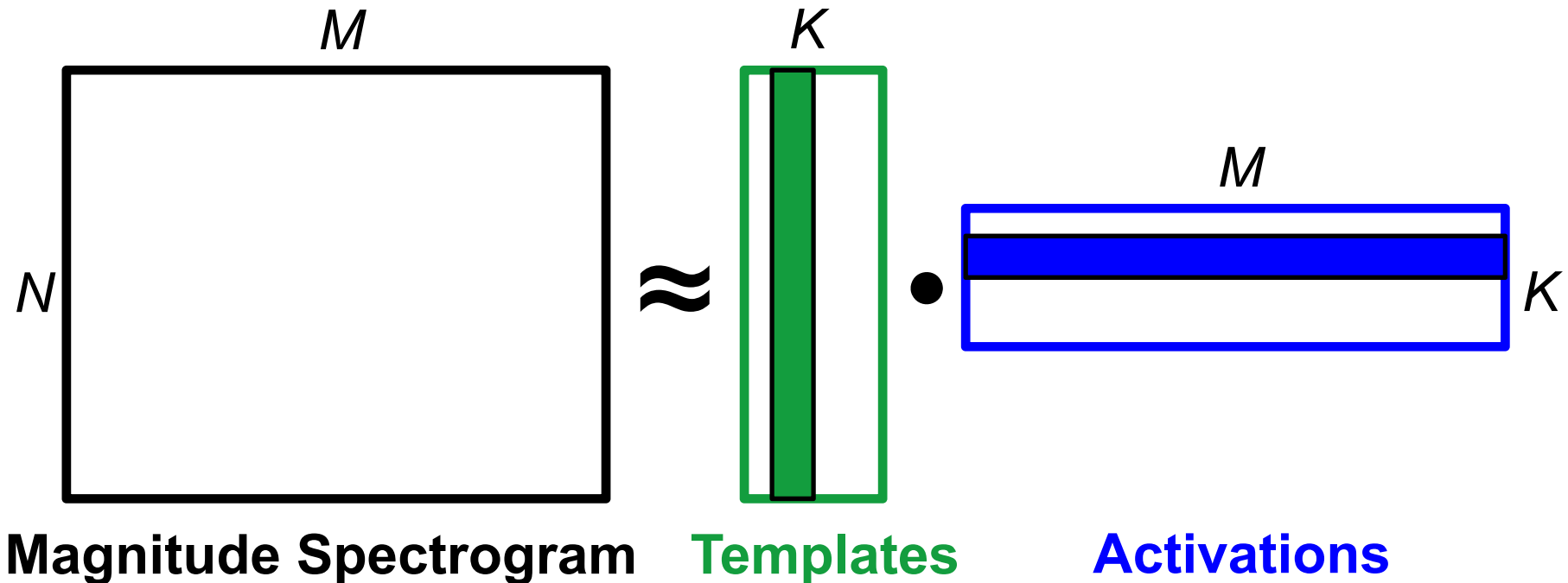
Activity patterns



NMF (Nonnegative Matrix Factorization)



NMF (Nonnegative Matrix Factorization)



Templates: Pitch + Timbre

“How does it sound”

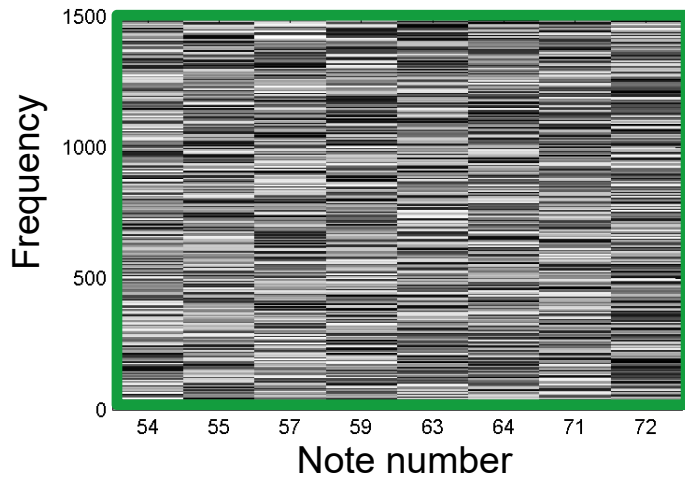
Activations: Onset time + Duration

“When does it sound”

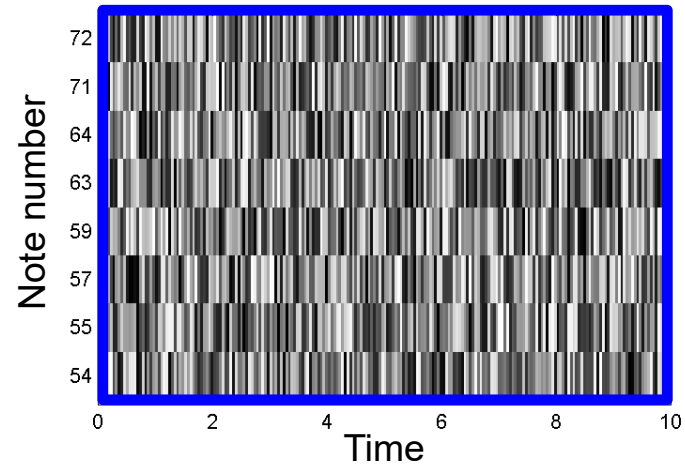
NMF-Decomposition

Random initialization

Initialized template



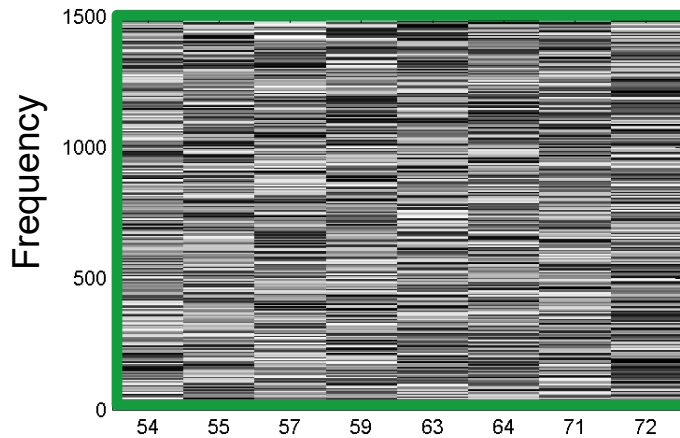
Initialized activations



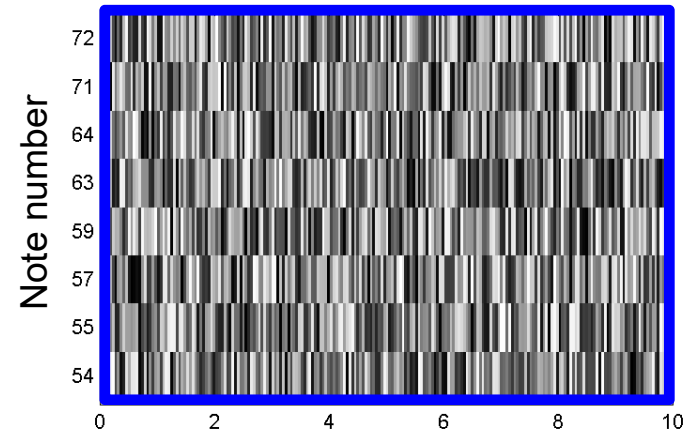
NMF-Decomposition

Random initialization → No semantic meaning

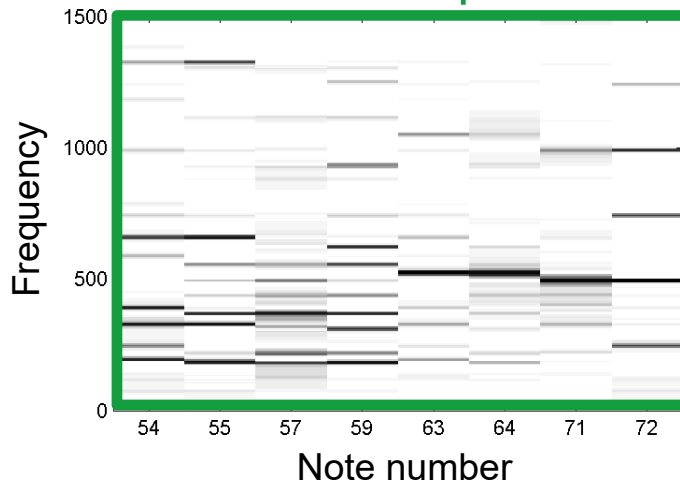
Initialized template



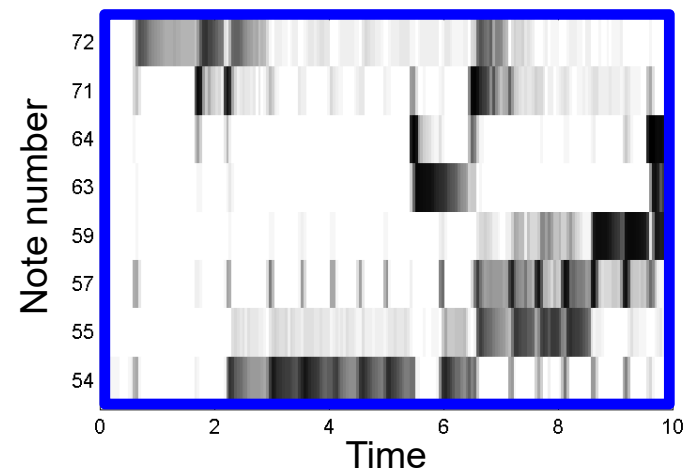
Initialized activations



Learnt templates



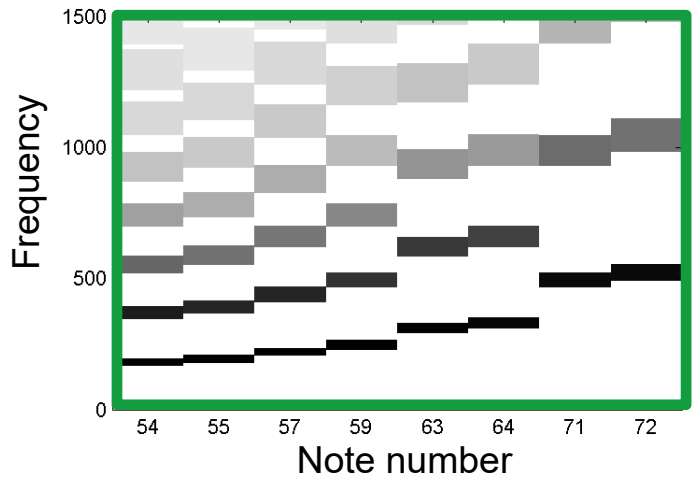
Learnt activations



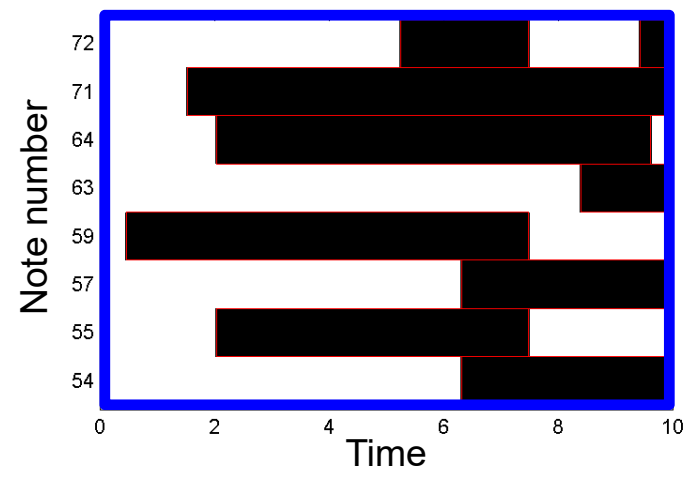
NMF-Decomposition

Constrained initialization

Initialized template

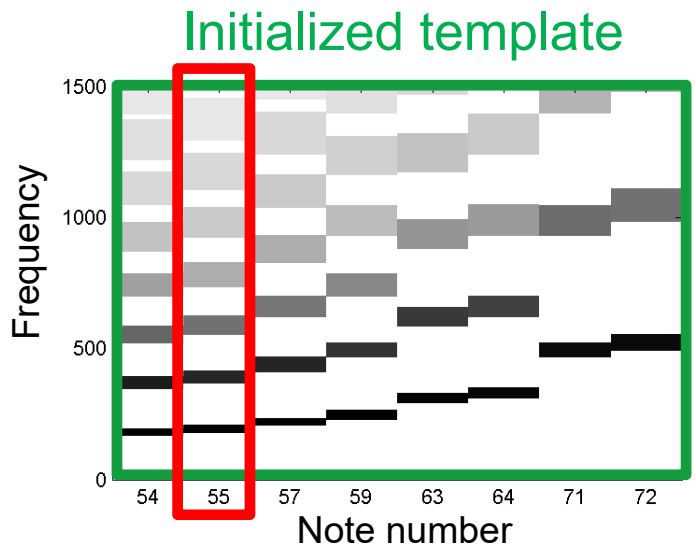


Initialized activations

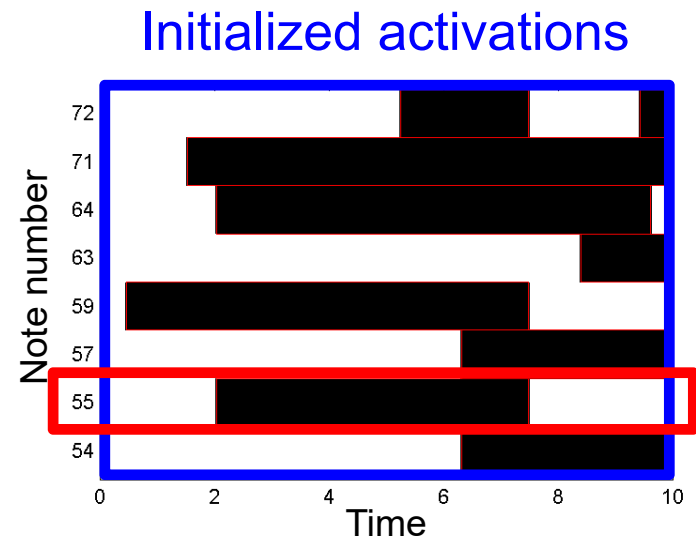


NMF-Decomposition

Constrained initialization



Template constraint for $p=55$

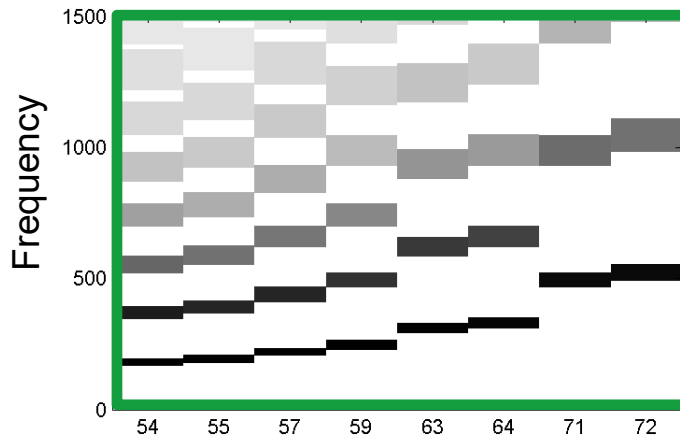


Activation constraint for $p=55$

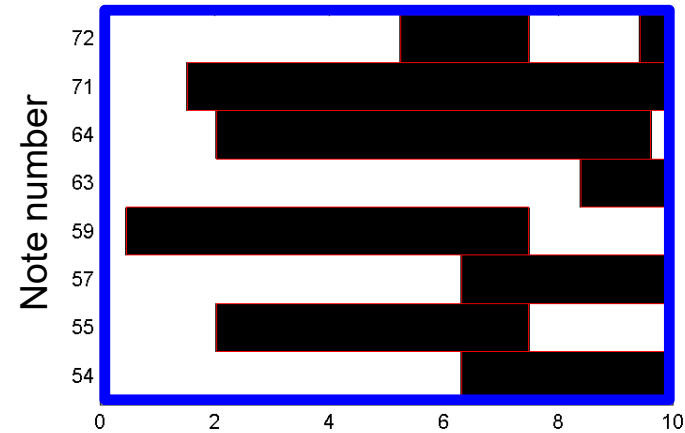
NMF-Decomposition

Constrained initialization → NMF as refinement

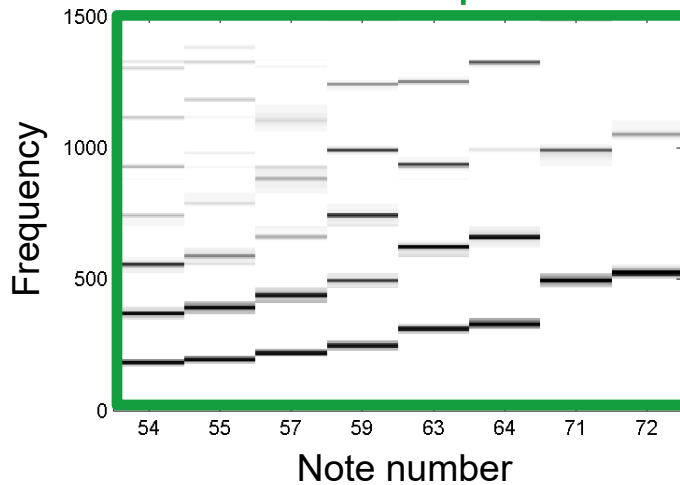
Initialized template



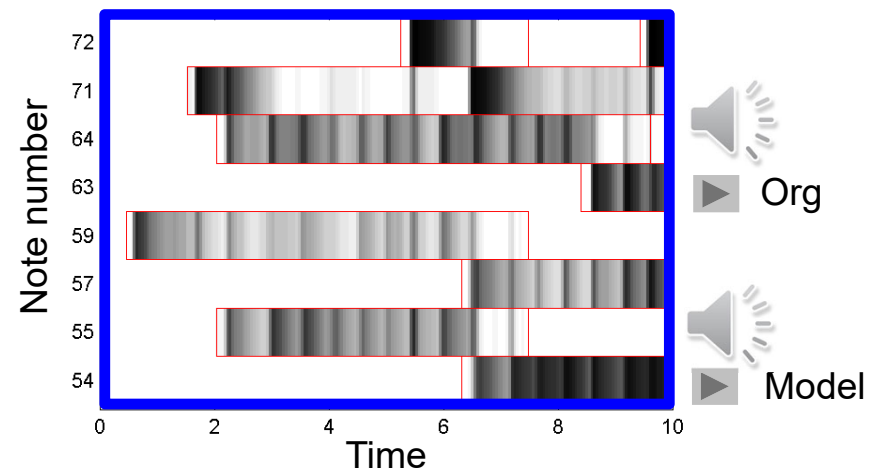
Initialized activations



Learnt templates

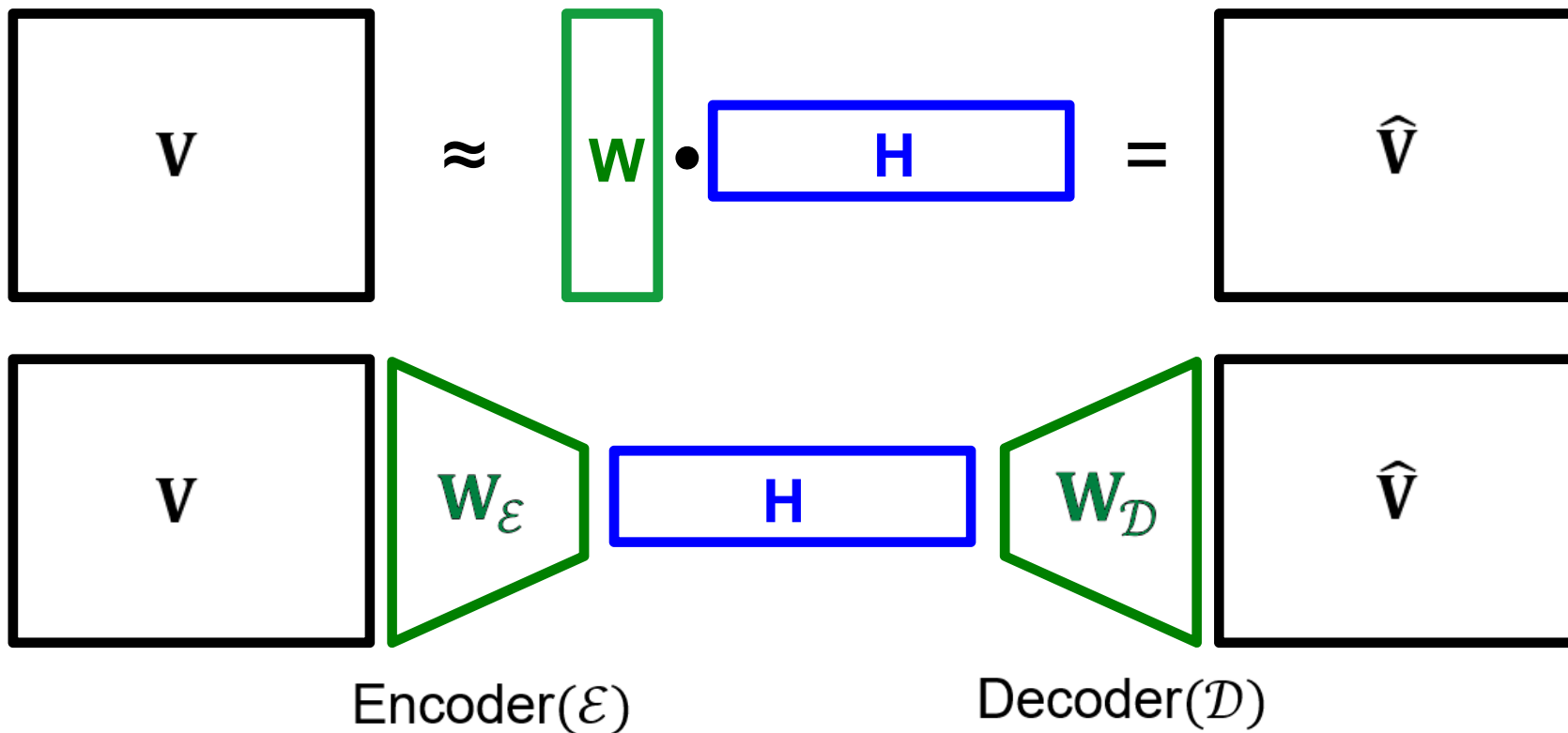


Learnt activations



NMF-Decomposition

Simulation of NMF via Autoencoder



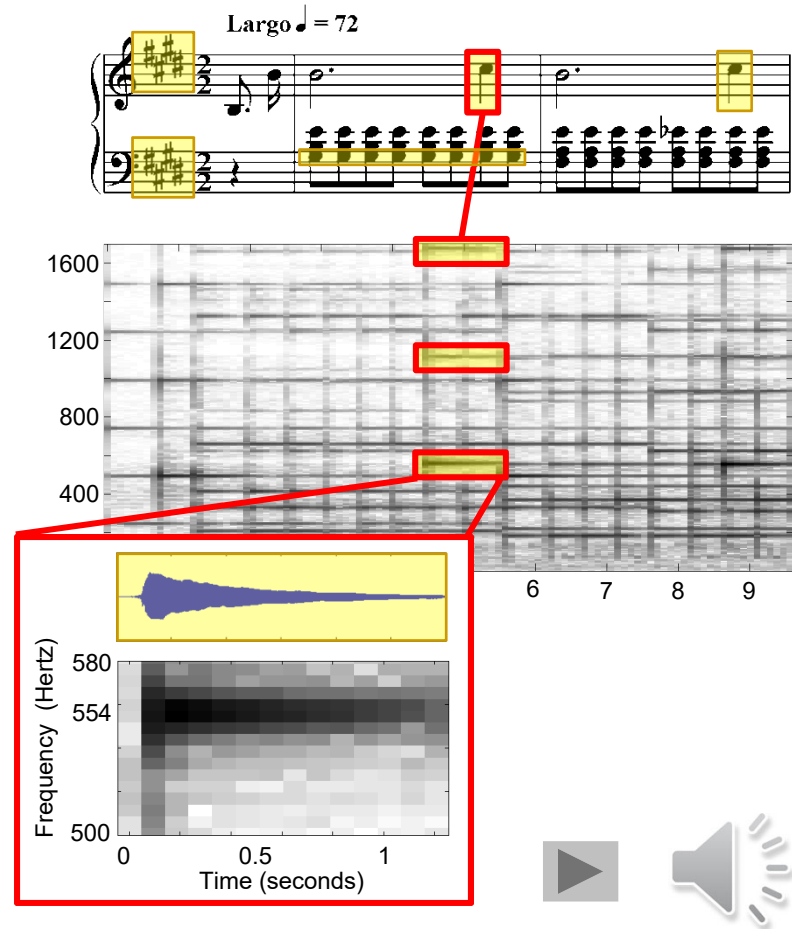
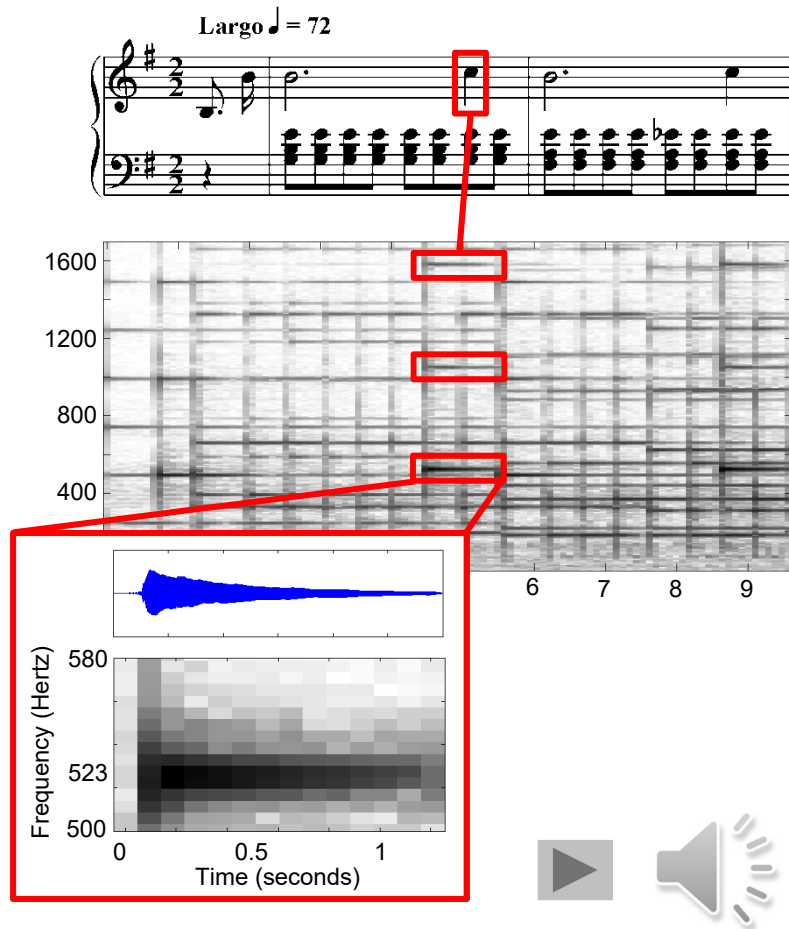
NMF as Autoencoder

Smaragdis, Venkataramani: A Neural Network Alternative to Non-Negative Audio Models. ICASSP, 2017.

Constraint Autoencoders

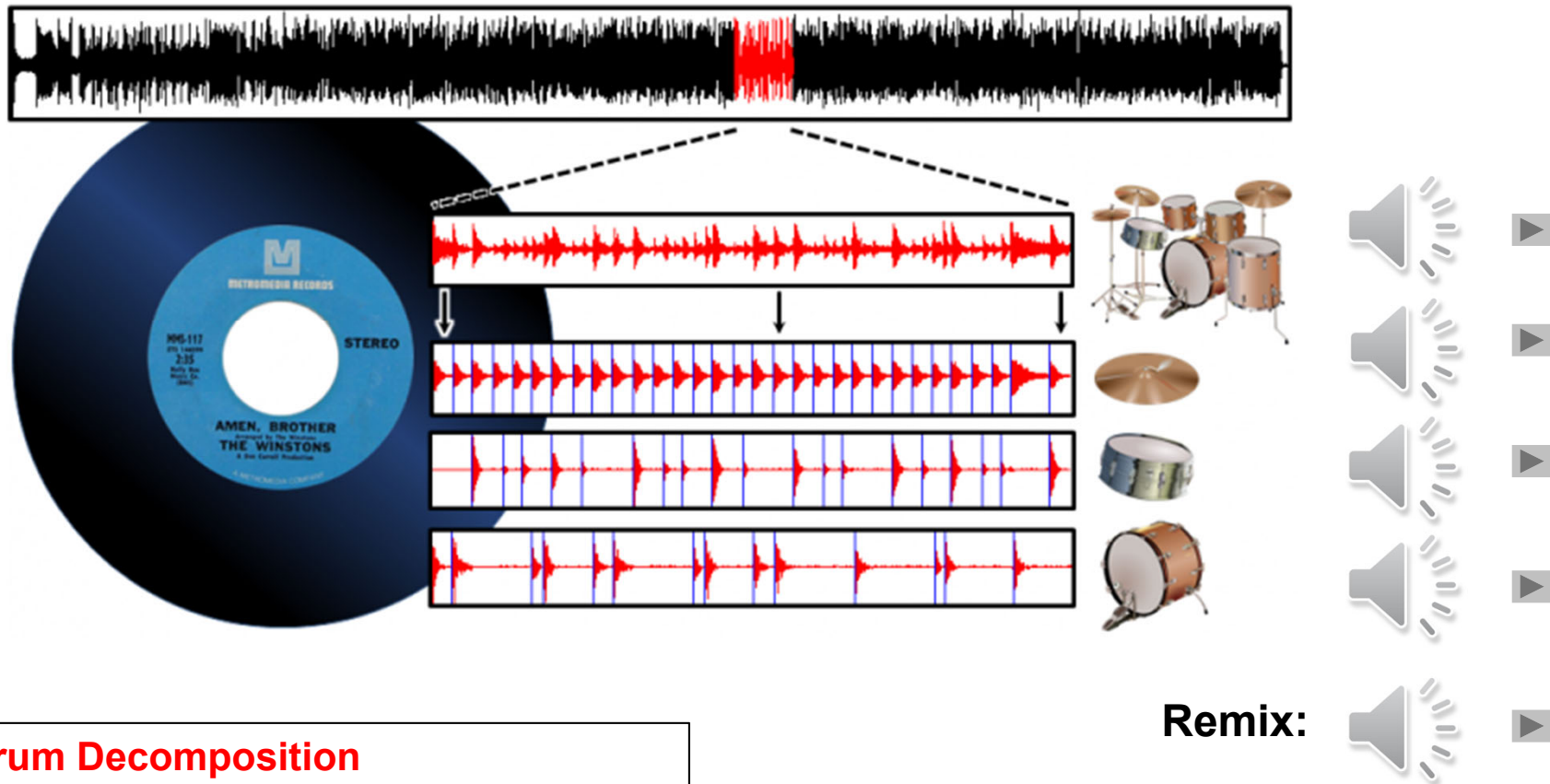
Ewert, Sandler: Structured dropout for weak label and multi-instance learning and its application to score-informed source separation. ICASSP, 2017

Score-Informed Audio Decomposition



Score-Informed Audio Decomposition

Informed Drum-Sound Decomposition



Drum Decomposition

Dittmar, Müller: Reverse Engineering the Amen Break – Score-Informed Separation and Restoration Applied to Drum Recordings. IEEE/ACM TASLP 24(9), 2016.

Score-Informed Audio Decomposition

Major challenge: Reconstructed sound events often have artifacts

Approaches:

- Resynthesize certain sound components
- Differentiable Digital Signal Processing (DDSP) combines classical DSP and deep learning
- Generative adversarial networks may help to reduce the artifacts

DDSP

Engel et al.: DDSP:
Differentiable Digital Signal
Processing. ICLR, 2020.

Score-Informed Audio Decomposition

Audio mosaicing (style transfer)

Target signal: Beatles–Let it be



Source signal: Bees

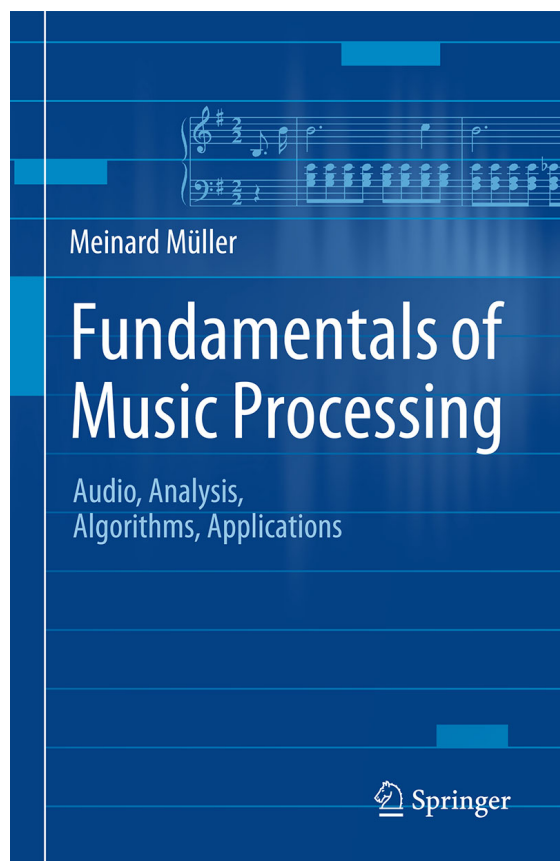


Mosaic signal: **Let it Bee**

Audio Mosaicing

Driedger, Prätzlich, Müller: Let It Bee – Towards NMF-Inspired Audio Mosaicing. ISMIR, 2015.

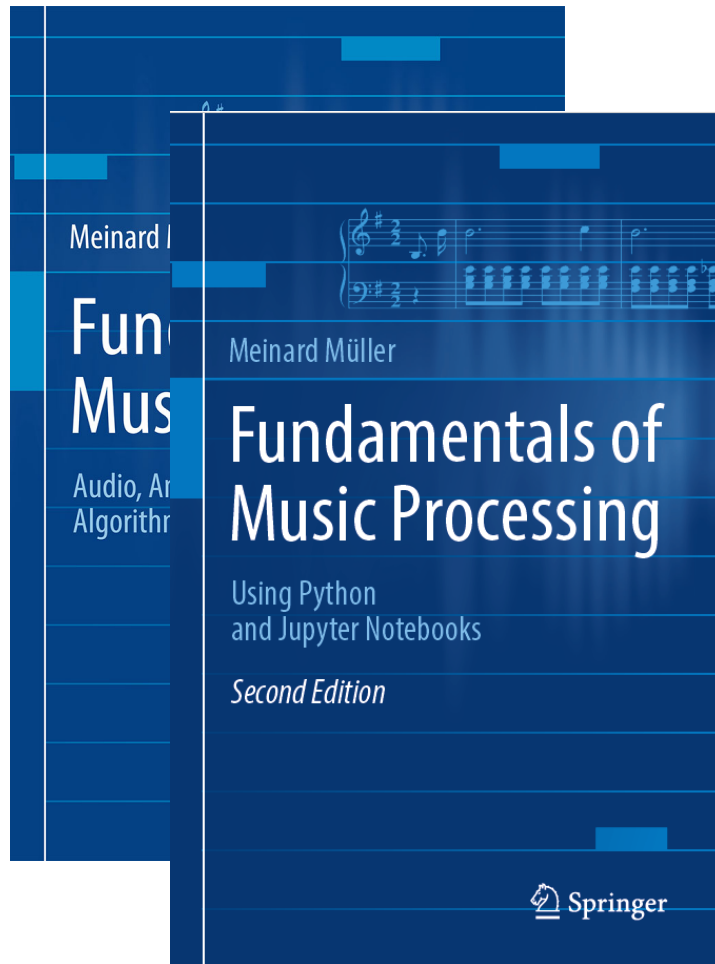
Fundamentals of Music Processing (FMP)



Meinard Müller
Fundamentals of Music Processing
Audio, Analysis, Algorithms, Applications
Springer, 2015

Accompanying website:
www.music-processing.de

Fundamentals of Music Processing (FMP)

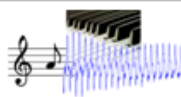

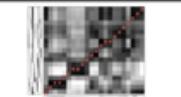
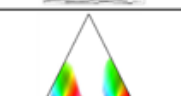

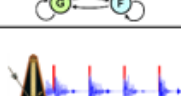




Meinard Müller
Fundamentals of Music Processing
Audio, Analysis, Algorithms, Applications
Springer, 2015

Accompanying website:
www.music-processing.de

2nd edition
Meinard Müller
Fundamentals of Music Processing
Using Python and Jupyter Notebooks
Springer, 2021

Fundamentals of Music Processing (FMP)

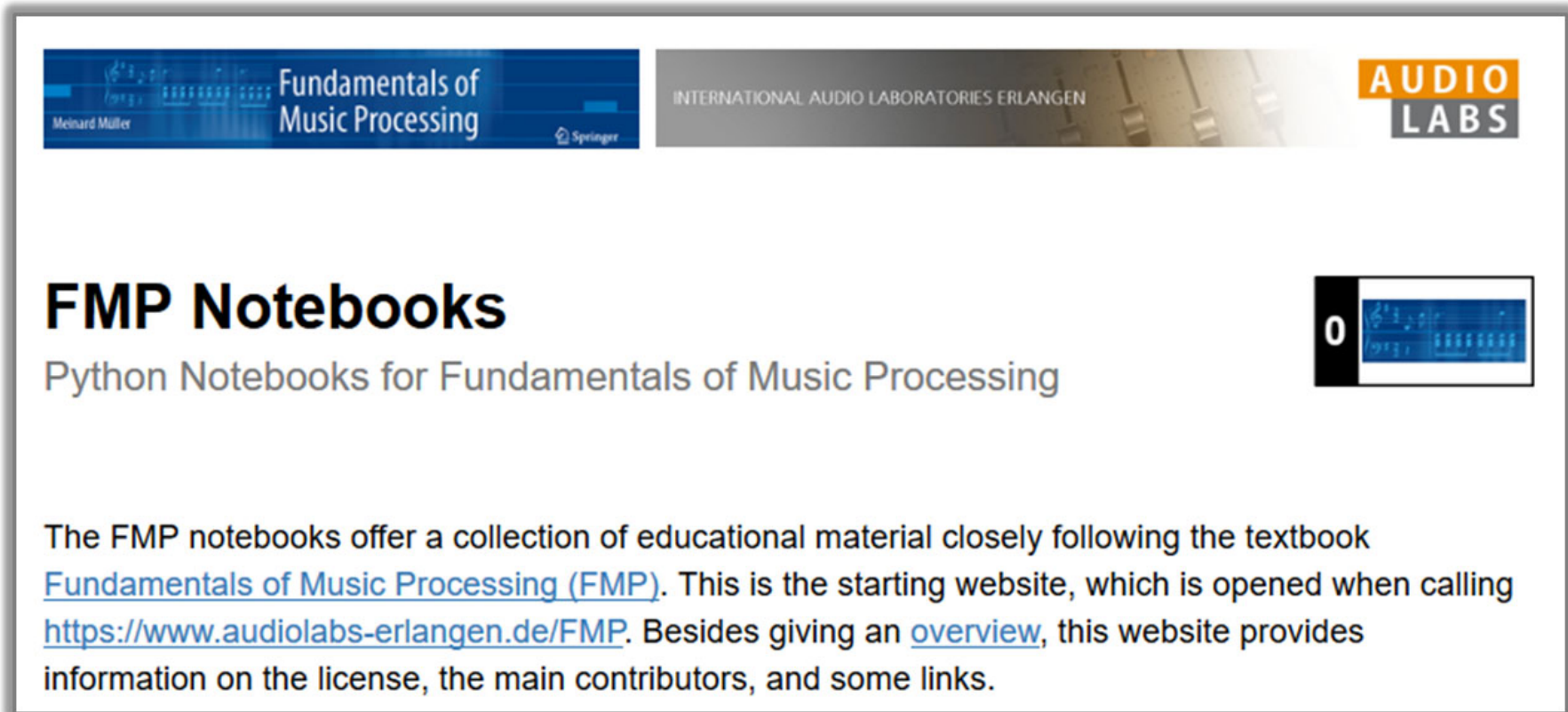
Chapter		Music Processing Scenario
1		Music Representations
2		Fourier Analysis of Signals
3		Music Synchronization
4		Music Structure Analysis
5		Chord Recognition
6		Tempo and Beat Tracking
7		Content-Based Audio Retrieval
8		Musically Informed Audio Decomposition

Meinard Müller
Fundamentals of Music Processing
Audio, Analysis, Algorithms, Applications
Springer, 2015

Accompanying website:
www.music-processing.de

2nd edition
Meinard Müller
Fundamentals of Music Processing
Using Python and Jupyter Notebooks
Springer, 2021

FMP Notebooks: Education & Research



The screenshot shows the header of the FMP Notebooks website. On the left, there is a blue banner for the book "Fundamentals of Music Processing" by Meinard Müller, published by Springer. To the right of this banner is the text "INTERNATIONAL AUDIO LABORATORIES ERLANGEN" and the "AUDIO LABS" logo. Below the banner, the main heading "FMP Notebooks" is displayed in a large, bold, black font. Underneath it, the subtitle "Python Notebooks for Fundamentals of Music Processing" is shown in a smaller, grey font. To the right of the subtitle is a small icon of a notebook with a white cover and a blue spine, featuring a white circle with the number "0" on the cover. Below the subtitle, a paragraph of text describes the notebooks: "The FMP notebooks offer a collection of educational material closely following the textbook [Fundamentals of Music Processing \(FMP\)](#). This is the starting website, which is opened when calling <https://www.audiolabs-erlangen.de/FMP>. Besides giving an [overview](#), this website provides information on the license, the main contributors, and some links."

<https://www.audiolabs-erlangen.de/FMP>

References (FMP Textbook & Notebooks)

- Meinard Müller: Fundamentals of Music Processing – Using Python and Jupyter Notebooks. 2nd Edition, Springer, 2021.
<https://www.springer.com/gp/book/9783030698072>
- Meinard Müller and Frank Zalkow: libfmp: A Python Package for Fundamentals of Music Processing. Journal of Open Source Software (JOSS), 6(63): 1–5, 2021.
<https://joss.theoj.org/papers/10.21105/joss.03326>
- Meinard Müller: An Educational Guide Through the FMP Notebooks for Teaching and Learning Fundamentals of Music Processing. Signals, 2(2): 245–285, 2021.
<https://www.mdpi.com/2624-6120/2/2/18>
- Meinard Müller and Frank Zalkow: FMP Notebooks: Educational Material for Teaching and Learning Fundamentals of Music Processing. Proc. International Society for Music Information Retrieval Conference (ISMIR): 573–580, 2019.
<https://zenodo.org/record/3527872#.YOhEQOgzaUk>
- Meinard Müller, Brian McFee, and Katherine Kinnaird: Interactive Learning of Signal Processing Through Music: Making Fourier Analysis Concrete for Students. IEEE Signal Processing Magazine, 38(3): 73–84, 2021.
<https://ieeexplore.ieee.org/document/9418542>

Resources (Group Meinard Müller)

- FMP Notebooks:

<https://www.audiolabs-erlangen.de/FMP>

- libfmp:

<https://github.com/meinardmueller/libfmp>

- synctoolbox:

<https://github.com/meinardmueller/synctoolbox>

- libtsm:

<https://github.com/meinardmueller/libtsm>

- Preparation Course Python (PCP) Notebooks:

<https://www.audiolabs-erlangen.de/resources/MIR/PCP/PCP.html>

<https://github.com/meinardmueller/PCP>

Resources

- librosa:
<https://librosa.org/>
- madmom:
<https://github.com/CPJKU/madmom>
- Essentia Python tutorial:
https://essentia.upf.edu/essentia_python_tutorial.html
- mirdata:
<https://github.com/mir-dataset-loaders/mirdata>
- open-unmix:
<https://github.com/sigsep/open-unmix-pytorch>
- Open Source Tools & Data for Music Source Separation:
<https://source-separation.github.io/tutorial/landing.html>

