Workshop HfM Karlsruhe

Music Information Retrieval

Harmony Analysis

Christof Weiß, Frank Zalkow, Meinard Müller
International Audio Laboratories Erlangen

christof.weiss@audiolabs-erlangen.de
frank.zalkow@audiolabs-erlangen.de
meinard.mueller@audiolabs-erlangen.de
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
Book: Fundamentals of Music Processing

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Music Processing Scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Music Representations</td>
</tr>
<tr>
<td>2</td>
<td>Fourier Analysis of Signals</td>
</tr>
<tr>
<td>3</td>
<td>Music Synchronization</td>
</tr>
<tr>
<td>4</td>
<td>Music Structure Analysis</td>
</tr>
<tr>
<td>5</td>
<td>Chord Recognition</td>
</tr>
<tr>
<td>6</td>
<td>Tempo and Beat Tracking</td>
</tr>
<tr>
<td>7</td>
<td>Content-Based Audio Retrieval</td>
</tr>
<tr>
<td>8</td>
<td>Musically Informed Audio Decomposition</td>
</tr>
</tbody>
</table>

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
In Chapter 5, we consider the problem of analyzing harmonic properties of a piece of music by determining a descriptive progression of chords from a given audio recording. We take this opportunity to first discuss some basic theory of harmony including concepts such as intervals, chords, and scales. Then, motivated by the automated chord recognition scenario, we introduce template-based matching procedures and hidden Markov models—a concept of central importance for the analysis of temporal patterns in time-dependent data streams including speech, gestures, and music.
Dissertation: Tonality-Based Style Analysis

Christof Weiβ
Computational Methods for Tonality-Based Style Analysis of Classical Music Audio Recordings
Dissertation, Ilmenau University of Technology, 2017

Chapter 5: Analysis Methods for Key and Scale Structures
Chapter 6: Design of Tonal Features
Recall: Chroma Features

- Human perception of pitch is periodic
- Two components: tone height (octave) and chroma (pitch class)
Recall: Chroma Features
Recall: Chroma Representations

L. van Beethoven, *Fidelio*, Overture, Slovak Philharmonic
Recall: Chroma Representations

- **Orchestra**
  
  L. van Beethoven, *Fidelio*, Overture, Slovak Philharmonic

- **Piano**
  
  *Fidelio*, Overture, arr. Alexander Zemlinsky
  M. Namekawa, D.R. Davies, piano four hands
Tonal Structures

- Western music (and most other music): Different aspects of harmony
- Referring to different time scales
Tonal Structures

- Western music (and most other music): Different aspects of harmony
- Referring to different time scales
Chord Recognition

Let It Be chords
The Beatles 1970  (Let It Be)

[Intro]
C G Am F C G  
F C Dm C

[Verse 1]
C G Am F  
When I find myself in times of trouble, Mother Mary comes to me  
C G F C Dm C  
Speaking words of wisdom, let it be

C G Am F  
And in my hour of darkness, she is standing right in front of me  
C G F C Dm C  
Speaking words of wisdom, let it be

[Chorus]
C Am G F C  
Let it be, let it be, let it be, let it be  
C G F C Dm C  
Whisper words of wisdom, let it be

Source: www.ultimate-guitar.com
Chord Recognition

C G Am F C G F C

[Music notation and waveform graph with highlighted chords C G Am F C G F C]
Chord Recognition
Chord Recognition

Prefiltering
- Compression
- Overtones
- Smoothing

Chroma representation

Pattern matching
- Major triads
- Minor triads

Postfiltering
- Smoothing
- Transition
- HMM

Recognition result

Audio representation

C G
Chord Recognition: Basics

- Musical chord: Group of three or more notes
- Combination of three or more tones which sound simultaneously
- Types: triads (major, minor, diminished, augmented), seventh chords...
- Here: focus on major and minor triads

C → C Major (C)
Chord Recognition: Basics

- Musical chord: Group of three or more notes
- Combination of three or more tones which sound simultaneously
- Types: triads (major, minor, diminished, augmented), seventh chords...
- Here: focus on major and minor triads

![Musical notation for C Major and C Minor chords]

- Enharmonic equivalence: 12 different root notes possible → 24 chords
Chord Recognition: Basics

Chords appear in different forms:

- Inversions

- Different voicings

- Harmonic figuration: Broken chords (arpeggio)

- Melodic figuration: Different melody note (suspension, passing tone, …)

- Further: Additional notes, incomplete chords
Chord Recognition: Basics

- Templates: Major Triads
Chord Recognition: Basics

- Templates: **Major Triads**
Chord Recognition: Basics

- Templates: **Minor Triads**

![Chord Diagram]
Chord Recognition: Template Matching

Chroma vector for each audio frame

24 chord templates (12 major, 12 minor)

Compute for each frame the similarity of the chroma vector to the 24 templates

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>C#</th>
<th>D</th>
<th>...</th>
<th>Cm</th>
<th>C#m</th>
<th>Dm</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>A#</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>G#</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>G</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>F#</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>F</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>D#</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>C#</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
</tbody>
</table>
Chord Recognition: Template Matching

- Similarity measure: Cosine similarity (inner product of normalized vectors)

Chord template: $t \in \mathbb{R}^{12}$

Chroma vector: $c \in \mathbb{R}^{12}$

Similarity measure: $s(t, c) = \frac{\langle t | c \rangle}{\|t\| \cdot \|c\|}$
Chord Recognition: Template Matching
Chord Recognition: Label Assignment

- Chroma vector for each audio frame
- 24 chord templates (12 major, 12 minor)

Compute for each frame the **similarity** of the chroma vector to the 24 templates

Assign to each frame the chord label of the template that maximizes the similarity to the chroma vector

|   | C | C# | D | ... | Cm | C#m | Dm | ...
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>A#</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>G#</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>G</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>F#</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>F</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>D#</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>C#</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
</tbody>
</table>
Chord Recognition: Label Assignment

![Chord Recognition Diagram]

- Chord Recognition: Label Assignment
- Time (seconds)
- Chord Chart with time intervals
- Chaord Recognition Diagram with time and chord chart
Chord Recognition: Evaluation
Chord Recognition: Evaluation

- “No-Chord” annotations: not every frame labeled

- Different evaluation measures:
  - Precision: \( P = \frac{\text{#TP}}{\text{#TP} + \text{#FP}} \)
  - Recall: \( R = \frac{\text{#TP}}{\text{#TP} + \text{#FN}} \)
  - F-Measure (balances precision and recall):
    \( F = \frac{2 \cdot P \cdot R}{P + R} \)

- Without “No-Chord” label: \( P = R = F \)
Chord Recognition: Smoothing

- Apply average filter of length $L \in \mathbb{N}$:

![Musical notation with chords C, Dm, G, C]
Chord Recognition: Smoothing

- Apply average filter of length $L \in \mathbb{N}$:
Chord Recognition: Smoothing

- Evaluation on all Beatles songs
Markov Chains

- Probabilistic model for sequential data
- **Markov property**: Next state only depends on current state (no “memory”)
- Consist of:
  - Set of states (hidden)
  - State transition probabilities
  - Initial state probabilities
Markov Chains

Notation:

\[ \alpha_i \text{ for } i \in [1: I] \]

State transition probabilities \( a_{ij} \)

States

\[ \begin{array}{c|ccc}
\hline
& \alpha_1 & \alpha_2 & \alpha_3 \\
\hline
\alpha_1 & a_{11} & a_{12} & a_{13} \\
\alpha_2 & a_{21} & a_{22} & a_{23} \\
\alpha_3 & a_{31} & a_{32} & a_{33} \\
\hline
\end{array} \]

Initial state probabilities \( c_i \)

\[ \begin{array}{c|ccc}
\hline
& \alpha_1 & \alpha_2 & \alpha_3 \\
\hline
c_1 & c_1 & c_2 & c_3 \\
\hline
\end{array} \]

\[ \beta_k \text{ for } k \in [1: K] \]
Markov Chains

- Application examples:
  - Compute probability of a sequence using given a model (evaluation)
  - Compare two sequences using a given model
  - Evaluate a sequence with two different models (classification)
Hidden Markov Models

- States as **hidden** variables

- Consist of:
  - Set of states (hidden)
  - State transition probabilities
  - *Initial state probabilities*
Hidden Markov Models

- States as **hidden** variables

- Consist of:
  - Set of states (hidden)
  - State transition probabilities
  - Initial state probabilities
  - Observations (visible)
Hidden Markov Models

- States as hidden variables

- Consist of:
  - Set of states (hidden)
  - State transition probabilities
  - Initial state probabilities
  - Observations (visible)
  - Emission probabilities
Hidden Markov Models

Notation:

- States: $\alpha_i$ for $i \in [1:I]$
- State transition probabilities: $a_{ij}$
- Initial state probabilities: $c_i$
- Observation symbols: $\beta_k$ for $k \in [1:K]$
- Emission probabilities: $b_{ik}$

Diagram:

- States: A, B, C, F, G
- Transition probabilities:
  - $a_{11}$, $a_{12}$, $a_{13}$
  - $a_{21}$, $a_{22}$, $a_{23}$
  - $a_{31}$, $a_{32}$, $a_{33}$
- Emission probabilities:
  - $b_{11}$, $b_{12}$, $b_{13}$
  - $b_{21}$, $b_{22}$, $b_{23}$
  - $b_{31}$, $b_{32}$, $b_{33}$
- Observation symbols:
  - $\alpha_1$, $\alpha_2$, $\alpha_3$
  - $\beta_1$, $\beta_2$, $\beta_3$
Markov Chains

- Analogon: the student’s life
- Consists of:
  - Set of states (hidden)
  - State transition probabilities
  - Initial state probabilities
Hidden Markov Models

- Analogon: the student's life
- Consists of:
  - **Set of states (hidden)**
  - State transition probabilities
  - *Initial state probabilities*
  - Observations (visible)
  - Emission probabilities
Hidden Markov Models

- Only observation sequence is visible!

Different algorithmic problems:

- **Evaluation problem**
  - Given: observation sequence and model
  - Calculate how well the model matches the sequence

- **Uncovering problem:**
  - Given: observation sequence and model
  - Find: optimal hidden state sequence

- **Estimation problem** ("training" the HMM):
  - Given: observation sequence
  - Find: model parameters
  - Baum-Welch algorithm (Expectation-Maximization)
Uncovering problem

- Given: observation sequence $O = (o_1, ..., o_N)$ of length $N \in \mathbb{N}$ and HMM $\theta$ (model parameters)
- Find: optimal hidden state sequence $S^* = (s_1^*, ..., s_N^*)$
- Corresponds to chord estimation task!

Observation sequence $O = (o_1, o_2, o_3, o_4, o_5, o_6)$
Uncovering problem

- Given: observation sequence \( O = (o_1, ..., o_N) \) of length \( N \in \mathbb{N} \) and HMM \( \Theta \) (model parameters)
- Find: optimal hidden state sequence
- Corresponds to chord estimation task!

Observation sequence \( O = (o_1, o_2, o_3, o_4, o_5, o_6) \)

Hidden state sequence \( S^* = (s_1^*, s_2^*, s_3^*, s_4^*, s_5^*, s_6^*) \)
Uncovering problem

- Given: observation sequence $O = (o_1, ..., o_N)$ of length $N \in \mathbb{N}$ and HMM $\theta$ (model parameters)
- Find: optimal hidden state sequence
- Corresponds to chord estimation task!
Uncovering problem

- **Optimal** hidden state sequence?
  - “Best explains” given observation sequence $O$
  - Maximizes probability $P[O, S | \Theta]$

$$\text{Prob}^* = \max_S P[O, S | \Theta]$$

$$S^* = \arg\max_S P[O, S | \Theta]$$

- Straight-forward computation (naive approach):
  - Compute probability for each possible sequence $S$
  - Number of possible sequences of length $N$ ($I = \text{number of states}$):

$$I \cdot I \cdots \cdot I = I^N$$

$N$ factors

 computationally infeasible!
Viterbi Algorithm

- Based on dynamic programming (similar to DTW)
- Idea: Recursive computation from subproblems
- Use truncated versions of observation sequence

\[ O(1:n) := (o_1, ..., o_n), \text{ length } n \in [1:N] \]

- Define \( D(i, n) \) as the highest probability along a single state sequence \((s_1, ..., s_n)\) that ends in state \( s_n = \alpha_i \)

\[ D(i, n) = \max_{(s_1, ..., s_n)} P[O(1:n), (s_1, ..., s_{n-1}, s_n = \alpha_i) | \Theta] \]

- Then, our solution is the state sequence yielding

\[ \text{Prob}^* = \max_{i \in [1:I]} D(i, N) \]
Viterbi Algorithm

- \( \mathbf{D} \): matrix of size \( I \times N \)
- Recursive computation of \( \mathbf{D}(i, n) \) along the column index \( n \)
- Initialization:
  - \( n = 1 \)
  - Truncated observation sequence: \( O(1) = (o_1) \)
  - Current observation: \( o_1 = \beta_{k_1} \)

\[
\mathbf{D}(i, 1) = c_i \cdot b_{ik_1} \quad \text{for some } i \in [1:I]
\]
Viterbi Algorithm

- **D**: matrix of size $I \times N$
- Recursive computation of $D(i, n)$ along the column index $n$
- **Recursion**:
  - $n \in [2: N]$
  - Truncated observation sequence: $O(1:n) = (o_1, ..., o_n)$
  - Last observation: $o_n = \beta_{k_n}$

\[
D(i, n) = b_{ik_n} \cdot a_{j^*_i} \cdot P[O(1:n-1), (s_1, ..., s_{n-1} = a_{j^*}) | \Theta] \quad \text{for } i \in [1: I]
\]

\[
D(i, n) = b_{ik_n} \cdot a_{j^*_i} \cdot D(j^*, n - 1)
\]

must be maximal!
Viterbi Algorithm

- **D**: matrix of size $I \times N$
- Recursive computation of $D(i, n)$ along the column index $n$
- Recursion:
  - $n \in [2: N]$
  - Truncated observation sequence: $O(1:n) = (o_1, ..., o_n)$
  - Last observation: $o_n = \beta_k n$

\[
D(i, n) = b_{ikn} \cdot a_{j^*i} \cdot P\left[O(1:n - 1), (s_1, ..., s_{n-1} = a_{j^*}) \mid \Theta\right] \quad \text{for } i \in [1: I]
\]

\[
D(i, n) = b_{ikn} \cdot a_{j^*i} \cdot D(j^*, n - 1)
\]

must be maximal (best index $j^*$)

\[
D(i, n) = b_{ikn} \cdot \max_{j \in [1:I]} \left( a_{ji} \cdot D(j, n - 1) \right)
\]
Viterbi Algorithm

- $\mathbf{D}$ given – find optimal state sequence $S^* = (s_1^*, ..., s_N^*) := (\alpha_{i_1}, ..., \alpha_{i_N})$
- Backtracking procedure (reverse order)
- Last element:
  - $n = N$
  - Optimal state: $\alpha_{i_N}$

$$i_N = \arg\max_{j \in [1:N]} \mathbf{D}(j, N)$$
Viterbi Algorithm

- $D$ given – find optimal state sequence $S^* = (s_1^*, ..., s_N^*) := (\alpha_{i_1}, ..., \alpha_{i_N})$
- Backtracking procedure (reverse order)
- Further elements:
  - $n = N - 1, N - 2, ..., 1$
  - Optimal state: $\alpha_{i_n}$

$$i_n = \arg\max_{j \in [1:l]} \left( a_{ji_{n+1}} \cdot D(i, n) \right)$$
Viterbi Algorithm

- $D$ given – find optimal state sequence $S^* = (s_1^*, \ldots, s_N^*) := (\alpha_{i_1}, \ldots, \alpha_{i_N})$

- Backtracking procedure (reverse order)

- Further elements:
  - $n = N - 1, N - 2, \ldots, 1$
  - Optimal state: $\alpha_{i_n}$

$$i_n = \arg\max_{j \in [1:I]} \left( a_{j i_{n+1}} \cdot D(i, n) \right)$$

- Simplification of backtracking: Keep track of maximizing index $j$ in

$$D(i, n) = b_{i k_n} \cdot \max_{j \in [1:I]} \left( a_{j i} \cdot D(j, n - 1) \right)$$

- Define $I \times (N - 1)$ matrix $E$:

$$E(i, n - 1) = \arg\max_{j \in [1:I]} \left( a_{j i} \cdot D(j, n - 1) \right)$$
Viterbi Algorithm

Summary

Initialization

States $i \in [1: I]$ 

Sequence index $n \in [1: N]$
Viterbi Algorithm

Summary

Initialization

\[ D(i, 1) = c_i b_{i_k^1} \]

Recursion

States
\[ i \in [1: I] \]

Sequence index
\[ n \in [1: N] \]
Viterbi Algorithm

Summary

Initialization

Recursion

Sequence index $n \in [1:N]$
**Viterbi Algorithm**

**Summary**

- **Initialization**
  - States $i \in [1: I]$  
  - $D(i, 1) = c_i b_{ik_1}$

- **Recursion**
  - $D(j, n-1)$  
  - $a_{ji}$  
  - $b_{ik_n}$  
  - $D(i, n)$

Sequence index $n \in [1: N]$
Viterbi Algorithm

Summary

Initialization

Recursion

Termination

Sequence index $n \in [1:N]$
Viterbi Algorithm

Summary

States $i \in [1: I]$

Sequence index $n \in [1: N]$

Initialization

Recursion

Termination

Backtracking matrix $E$

$D(i, 1) = c_i b_{i k_1}$

$D(j, n - 1)$

$D(i, n)$

$D(i, N)$

$E$
Viterbi Algorithm

Computational Complexity

Per recursion step: \( I \cdot I \)

Total recursion: \( I^2 \cdot N \)
Viterbi Algorithm

Summary

**Algorithm:** VITERBI

**Input:** HMM specified by $\Theta = (A,A,C,B,B)$

Observation sequence $O = (o_1 = \beta_{k_1}, o_2 = \beta_{k_2}, \ldots, o_N = \beta_{k_N})$

**Output:** Optimal state sequence $S^* = (s_1^*, s_2^*, \ldots, s_N^*)$

**Procedure:** Initialize the $(I \times N)$ matrix $D$ by $D(i,1) = c_i b_{ik_1}$ for $i \in [1:I]$. Then compute in a nested loop for $n = 2, \ldots, N$ and $i = 1, \ldots, I$:

\[
D(i,n) = \max_{j \in [1:I]} \left( a_{ji} \cdot D(j, n-1) \right) \cdot b_{ik_n}
\]

\[
E(i,n-1) = \arg\max_{j \in [1:I]} \left( a_{ji} \cdot D(j, n-1) \right)
\]

Set $i_N = \arg\max_{j \in [1:I]} D(j, N)$ and compute for decreasing $n = N-1, \ldots, 1$ the maximizing indices

\[
i_n = \arg\max_{j \in [1:I]} \left( a_{ji_{n+1}} \cdot D(j, n) \right) = E(i_{n+1}, n).
\]

The optimal state sequence $S^* = (s_1^*, \ldots, s_N^*)$ is defined by $s_n^* = \alpha_{i_n}$ for $n \in [1:N]$. 
Viterbi Algorithm: Example

HMM:

States
\( \alpha_i \) for \( i \in [1:I] \)

State transition probabilities
\( a_{ij} \)

Observation symbols
\( \beta_k \) for \( k \in [1:K] \)

Emission probabilities
\( b_{ik} \)

Initial state probabilities
\( c_i \)

<table>
<thead>
<tr>
<th>A</th>
<th>( \alpha_1 )</th>
<th>( \alpha_2 )</th>
<th>( \alpha_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha_1 )</td>
<td>( a_{11} )</td>
<td>( a_{12} )</td>
<td>( a_{13} )</td>
</tr>
<tr>
<td>( \alpha_2 )</td>
<td>( a_{21} )</td>
<td>( a_{22} )</td>
<td>( a_{23} )</td>
</tr>
<tr>
<td>( \alpha_3 )</td>
<td>( a_{31} )</td>
<td>( a_{32} )</td>
<td>( a_{33} )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B</th>
<th>( \beta_1 )</th>
<th>( \beta_2 )</th>
<th>( \beta_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha_1 )</td>
<td>( b_{11} )</td>
<td>( b_{12} )</td>
<td>( b_{13} )</td>
</tr>
<tr>
<td>( \alpha_2 )</td>
<td>( b_{21} )</td>
<td>( b_{22} )</td>
<td>( b_{23} )</td>
</tr>
<tr>
<td>( \alpha_3 )</td>
<td>( b_{31} )</td>
<td>( b_{32} )</td>
<td>( b_{33} )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C</th>
<th>( \alpha_1 )</th>
<th>( \alpha_2 )</th>
<th>( \alpha_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c_1 )</td>
<td>( c_2 )</td>
<td>( c_3 )</td>
<td></td>
</tr>
</tbody>
</table>
## Viterbi Algorithm: Example

**HMM:**

<table>
<thead>
<tr>
<th>States</th>
<th>Observation symbols</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_i$ for $i \in [1:I]$</td>
<td>$\beta_k$ for $k \in [1:K]$</td>
</tr>
</tbody>
</table>

**State transition probabilities**

$\alpha_{ij}$

<table>
<thead>
<tr>
<th>A</th>
<th>$\alpha_1$</th>
<th>$\alpha_2$</th>
<th>$\alpha_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_1$</td>
<td>0.8</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td>0.2</td>
<td>0.7</td>
<td>0.1</td>
</tr>
<tr>
<td>$\alpha_3$</td>
<td>0.1</td>
<td>0.3</td>
<td>0.6</td>
</tr>
</tbody>
</table>

**Emission probabilities**

$b_{ik}$

<table>
<thead>
<tr>
<th>B</th>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
<th>$\beta_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_1$</td>
<td>0.7</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td>0.1</td>
<td>0.9</td>
<td>0</td>
</tr>
<tr>
<td>$\alpha_3$</td>
<td>0</td>
<td>0.2</td>
<td>0.8</td>
</tr>
</tbody>
</table>

**Initial state probabilities**

$c_i$

<table>
<thead>
<tr>
<th>C</th>
<th>$\alpha_1$</th>
<th>$\alpha_2$</th>
<th>$\alpha_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.6</td>
<td>0.2</td>
<td>0.2</td>
</tr>
</tbody>
</table>
Viterbi Algorithm: Example

HMM:

States
\( \alpha_i \) for \( i \in [1:I] \)

State transition probabilities
\( a_{ij} \)

<table>
<thead>
<tr>
<th></th>
<th>( \alpha_1 )</th>
<th>( \alpha_2 )</th>
<th>( \alpha_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.8</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>B</td>
<td>0.2</td>
<td>0.7</td>
<td>0.1</td>
</tr>
<tr>
<td>C</td>
<td>0.1</td>
<td>0.3</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Observation symbols
\( \beta_k \) for \( k \in [1:K] \)

Emission probabilities
\( b_{ik} \)

<table>
<thead>
<tr>
<th></th>
<th>( \beta_1 )</th>
<th>( \beta_2 )</th>
<th>( \beta_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.7</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>B</td>
<td>0.1</td>
<td>0.9</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0.2</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Initial state probabilities
\( c_i \)

<table>
<thead>
<tr>
<th></th>
<th>( \alpha_1 )</th>
<th>( \alpha_2 )</th>
<th>( \alpha_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>0.6</td>
<td>0.2</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Observation sequence
\( O = (o_1, o_2, o_3, o_4, o_5, o_6) \)
Viterbi Algorithm: Example

HMM:

States
\( \alpha_i \) for \( i \in [1:I] \)

State transition probabilities
\( a_{ij} \)

Observation symbols
\( \beta_k \) for \( k \in [1:K] \)

Emission probabilities
\( b_{ik} \)

Initial state probabilities
\( c_i \)

Input

Observation sequence
\[ O = (o_1, o_2, o_3, o_4, o_5, o_6) \]

Viterbi algorithm
Viterbi Algorithm: Example

HMM:

Input:
- o₁, o₂, o₃, o₄, o₅, o₆
- β₁, β₂, β₃

Viterbi algorithm

Initialization:
\[ D(i, 1) = c_i \cdot b_{ik_1} \]
Viterbi Algorithm: Example

HMM:

States
\( \alpha_i \) for \( i \in [1: I] \)

State transition probabilities \( a_{ij} \)

<table>
<thead>
<tr>
<th></th>
<th>( \alpha_1 )</th>
<th>( \alpha_2 )</th>
<th>( \alpha_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha_1 )</td>
<td>0.8</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>( \alpha_2 )</td>
<td>0.2</td>
<td>0.7</td>
<td>0.1</td>
</tr>
<tr>
<td>( \alpha_3 )</td>
<td>0.1</td>
<td>0.3</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Observation symbols \( \beta_k \) for \( k \in [1: K] \)

Emission probabilities \( b_{ik} \)

<table>
<thead>
<tr>
<th></th>
<th>( \beta_1 )</th>
<th>( \beta_2 )</th>
<th>( \beta_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha_1 )</td>
<td>0.7</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>( \alpha_2 )</td>
<td>0.1</td>
<td>0.9</td>
<td>0</td>
</tr>
<tr>
<td>( \alpha_3 )</td>
<td>0</td>
<td>0.2</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Initial state probabilities \( c_i \)

<table>
<thead>
<tr>
<th></th>
<th>( \alpha_1 )</th>
<th>( \alpha_2 )</th>
<th>( \alpha_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha_1 )</td>
<td>0.6</td>
<td>0.2</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Input

\( o_1, o_2, o_3, o_4, o_5, o_6 \)
\( \beta_1, \beta_3, \beta_1, \beta_3, \beta_3, \beta_2 \)

Viterbi algorithm

Initialization

\( D(i, 1) = c_i \cdot b_{ik_1} \)

Recursion

\( D(i, n) = b_{ik_n} \cdot \max_{j \in [1:I]} \left( a_{ji} \cdot D(j, n - 1) \right) \)

\( E(i, n - 1) = \arg\max_{j \in [1:I]} \left( a_{ji} \cdot D(j, n - 1) \right) \)
Viterbi Algorithm: Example

HMM:

States
\[ \alpha_i \text{ for } i \in [1:I] \]

State transition probabilities \( a_{ij} \)

Observation symbols
\[ \beta_k \text{ for } k \in [1:K] \]

Emission probabilities \( b_{ik} \)

Initial state probabilities \( c_i \)

Input
\[ o_1, o_2, o_3, o_4, o_5, o_6, \beta_1, \beta_3, \beta_1, \beta_3, \beta_3, \beta_2 \]

Viterbi algorithm

\[ D(i, 1) = c_i \cdot b_{i1} \]

Recursion
\[ D(i, n) = b_{ik_n} \cdot \max_{j \in [1:I]} \left( a_{ji} \cdot D(j, n - 1) \right) \]

\[ E(i, n - 1) = \arg\max_{j \in [1:I]} \left( a_{ji} \cdot D(j, n - 1) \right) \]
Viterbi Algorithm: Example

HMM:

States
\( \alpha_i \) for \( i \in [1:I] \)

State transition probabilities
\( a_{ij} \)

Observation symbols
\( \beta_k \) for \( k \in [1:K] \)

Emission probabilities
\( b_{ik} \)

Initial state probabilities
\( c_i \)

Input
\( o_1, o_2, o_3, o_4, o_5, o_6 \)
\( \beta_1, \beta_3, \beta_1, \beta_3, \beta_3, \beta_2 \)

Viterbi algorithm

\[
<table>
<thead>
<tr>
<th>D</th>
<th>o_1 = \beta_1</th>
<th>o_2 = \beta_3</th>
<th>o_3 = \beta_1</th>
<th>o_4 = \beta_3</th>
<th>o_5 = \beta_3</th>
<th>o_6 = \beta_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>\alpha_1</td>
<td>0.4200</td>
<td>0.1008</td>
<td>0.0564</td>
<td>0.0135</td>
<td>0.0033</td>
<td>0</td>
</tr>
<tr>
<td>\alpha_2</td>
<td>0.0200</td>
<td>0</td>
<td>0.0010</td>
<td>0</td>
<td>0</td>
<td>0.0006</td>
</tr>
<tr>
<td>\alpha_3</td>
<td>0</td>
<td>0.0336</td>
<td>0</td>
<td>0.0045</td>
<td>0.0022</td>
<td>0.0003</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>E</th>
<th>o_1 = \beta_1</th>
<th>o_2 = \beta_3</th>
<th>o_3 = \beta_1</th>
<th>o_4 = \beta_3</th>
<th>o_5 = \beta_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>\alpha_1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>\alpha_2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>\alpha_3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Backtracking

\[
i_N = \arg\max_{j \in [1:I]} D(i, n)
\]

\[
i_n = E(i_{n+1}, n)
\]
Viterbi Algorithm: Example

HMM:

States
\( \alpha_i \) for \( i \in [1:I] \)

State transition probabilities
\( a_{ij} \)

Observation symbols
\( \beta_k \) for \( k \in [1:K] \)

Emission probabilities
\( b_{ik} \)

Initial state probabilities
\( c_i \)

Input

<table>
<thead>
<tr>
<th>o_1</th>
<th>o_2</th>
<th>o_3</th>
<th>o_4</th>
<th>o_5</th>
<th>o_6</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta_1 )</td>
<td>( \beta_3 )</td>
<td>( \beta_1 )</td>
<td>( \beta_3 )</td>
<td>( \beta_3 )</td>
<td>( \beta_2 )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>States</th>
<th>( \alpha_1 )</th>
<th>( \alpha_2 )</th>
<th>( \alpha_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha_1 )</td>
<td>0.8</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>( \alpha_2 )</td>
<td>0.2</td>
<td>0.7</td>
<td>0.1</td>
</tr>
<tr>
<td>( \alpha_3 )</td>
<td>0.1</td>
<td>0.3</td>
<td>0.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Observation symbols</th>
<th>( \beta_1 )</th>
<th>( \beta_2 )</th>
<th>( \beta_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha_1 )</td>
<td>0.7</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>( \alpha_2 )</td>
<td>0.1</td>
<td>0.9</td>
<td>0</td>
</tr>
<tr>
<td>( \alpha_3 )</td>
<td>0</td>
<td>0.2</td>
<td>0.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Initial state probabilities</th>
<th>( c_1 )</th>
<th>( c_2 )</th>
<th>( c_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c_1 )</td>
<td>0.6</td>
<td>0.2</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Viterbi algorithm

\[ i_N = \arg\max_{j \in [1:I]} D(i, n) \]

Backtracking

\[ i_n = E(i_{n+1}, n) \]

\( i_6 = 2 \)
Viterbi Algorithm: Example

**HMM:**

States
\[ \alpha_i \text{ for } i \in [1:I] \]

State transition probabilities
\[ a_{ij} \]

<table>
<thead>
<tr>
<th>States</th>
<th>( \alpha_1 )</th>
<th>( \alpha_2 )</th>
<th>( \alpha_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha_1 )</td>
<td>0.8</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>( \alpha_2 )</td>
<td>0.2</td>
<td>0.7</td>
<td>0.1</td>
</tr>
<tr>
<td>( \alpha_3 )</td>
<td>0.1</td>
<td>0.3</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Emission probabilities
\[ b_{ik} \]

<table>
<thead>
<tr>
<th>Observation symbols</th>
<th>( \beta_1 )</th>
<th>( \beta_2 )</th>
<th>( \beta_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha_1 )</td>
<td>0.7</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>( \alpha_2 )</td>
<td>0.1</td>
<td>0.9</td>
<td>0</td>
</tr>
<tr>
<td>( \alpha_3 )</td>
<td>0</td>
<td>0.2</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Initial state probabilities
\[ c_i \]

<table>
<thead>
<tr>
<th>States</th>
<th>( \alpha_1 )</th>
<th>( \alpha_2 )</th>
<th>( \alpha_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha_1 )</td>
<td>0.6</td>
<td>0.2</td>
<td>0.2</td>
</tr>
</tbody>
</table>

**Input**

Observation sequence
\[ O = (o_1, o_2, o_3, o_4, o_5, o_6) \]

**Viterbi algorithm**

<table>
<thead>
<tr>
<th>States</th>
<th>( \alpha_1 )</th>
<th>( \alpha_2 )</th>
<th>( \alpha_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha_1 )</td>
<td>0.4200</td>
<td>0.1008</td>
<td>0.0564</td>
</tr>
<tr>
<td>( \alpha_2 )</td>
<td>0.0200</td>
<td>0</td>
<td>0.0010</td>
</tr>
<tr>
<td>( \alpha_3 )</td>
<td>0</td>
<td>0.0336</td>
<td>0.0045</td>
</tr>
</tbody>
</table>

**Output**

Optimal state sequence
\[ S^* = (\alpha_1, \alpha_1, \alpha_1, \alpha_3, \alpha_3, \alpha_2) \]
HMM: Application to Chord Recognition

- Effect of HMM-based chord estimation and smoothing:

(a) Template Matching (frame-wise)
(b) HMM
HMM: Application to Chord Recognition

- Parameters: Transition probabilities
- Estimated from data
HMM: Application to Chord Recognition

- Parameters: Transition probabilities
- Estimated from data
HMM: Application to Chord Recognition

- Parameters: *Transition probabilities*
- Transposition-invariant
HMM: Application to Chord Recognition

- Parameters: Transition probabilities
- Uniform transition matrix (only smoothing)
HMM: Application to Chord Recognition

- Evaluation on all Beatles songs
Chord Recognition: Further Challenges

- Chord ambiguities
- Acoustic ambiguities (overtones)
  - Use advanced templates (model overtones, learned templates)
  - Enhanced chroma (logarithmic compression, overtone reduction)
- Tuning inconsistency
Chord Recognition: Public System

- Chord recognition
  - Typically: Feature extraction, pattern matching, filtering (HMM)
  - "Out-of-the-box" solutions (Sonic Visualizer, Chordino plugin)

Sonic Visualizer, Chordino Vamp Plugin
(Queen Mary University of London)
Tonal Structures

- **Global key detection**
- **Local key detection**
- **Chord recognition**
- **Music transcription**

**Movement level**
- Global key: C major

**Segment level**
- C major
- Local key: G major
- C major

**Chord level**
- CM, GM\(^7\), Am
- Chords

**Note level**
- Melody
- Middle voices
- Bass line
Tonal Structures

Global key detection

Local key detection

Chord recognition

Music transcription

Movement level
Segment level
Chord level
Note level

Global key | C major
Local key | G major | C major
CM | GM7 | Am | Chords
Melody | Middle voices | Bass line
Local Key Detection

- Key as an important musical concept ("Symphony in C major")
- Modulations → Local approach
- Key relations: Circle of fifth (keys)
Local Key Detection

- Key as an important musical concept ("Symphony in C major")
- Modulations → Local approach
- Diatonic Scales
  - Simplification of keys
  - Perfect-fifth relation

Circle of fifths (pitches) →

2b diatonic  di 1# diatonic
Local Key Detection

- Example: J.S. Bach, Choral "Durch Dein Gefängnis" *(Johannespassion)*
- **Score** – Piano reduction
Local Key Detection

- Example: J.S. Bach, Choral "Durch Dein Gefängnis" (*Johannespassion*)
- **Audio** – Waveform (Scholars Baroque Ensemble, Naxos 1994)
Tonal Structures: Local Diatonic Scales

- Example: J.S. Bach, Choral "Durch Dein Gefängnis" (*Johannespassion*)
- **Audio** – Spectrogram (Scholars Baroque Ensemble, Naxos 1994)
Local Key Detection: Chroma Features

- Example: J.S. Bach, Choral "Durch Dein Gefängnis" (*Johannespassion*)
- **Audio** – Chroma features (Scholars Baroque Ensemble, Naxos 1994)
Local Key Detection: Chroma Smoothing

- Summarize pitch classes over a certain time
  - Chroma smoothing
  - Parameters: blocksize $b$ and hopsize $h$
Local Key Detection: Chroma Smoothing

- Choral (Bach)
Local Key Detection: Chroma Smoothing

- Choral (Bach) — smoothed with $b = 42$ seconds and $h = 15$ seconds
Local Key Detection: Diatonic Scales

- Choral (Bach) — Re-ordering to **perfect fifth** series
Local Key Detection: Diatonic Scales

- Choral (Bach) — Re-ordering to **perfect fifth** series
Local Key Detection: Diatonic Scales

- Choral (Bach) — Diatonic Scale Estimation (7 fifths)
Local Key Detection: Diatonic Scales

- Choral (Bach) — Diatonic Scale Estimation (7 fifths)
Local Key Detection: Diatonic Scales

- Choral (Bach) — Diatonic Scale Estimation: Multiply chroma values*
Local Key Detection: Diatonic Scales

- Choral (Bach) — Diatonic Scale Estimation: Multiply chroma values
Local Key Detection: Diatonic Scales

- Choral (Bach) — Diatonic Scale Estimation
Local Key Detection: Diatonic Scales

- Choral (Bach) — Diatonic Scale Estimation

4 #
(E major)
Local Key Detection: Diatonic Scales

- Choral (Bach) — Diatonic Scale Estimation: Shift to **global key**
Local Key Detection: Diatonic Scales

- Choral (Bach) — $0 \triangleq 4\#$

Weiss / Habryka, *Chroma-Based Scale Matching for Audio Tonality Analysis*, CIM 2014
Local Key Detection: Examples

- L. v. Beethoven – Sonata No. 10 op. 14 Nr. 2, 1. Allegro — 0 ≤ 1
  (Barenboim, EMI 1998)
Local Key Detection: Examples

- R. Wagner, *Die Meistersinger von Nürnberg*, Vorspiel — 0 ≅ 0
  (Polish National Radio Symphony Orchestra, J. Wildner, Naxos 1993)
DFG-funded Project: Computational Analysis of Harmonic Structures

- With Prof. Rainer Kleinertz, Musicology, Uni Saarland

- Richard Wagner, *Der Ring des Nibelungen*
  - Four operas, up to 15 hours of music
  - How is harmony organized at the large scale?
  - Analyses by A. Lorenz 1924
  - Hypothesis of „Poetico-musical periods“
Cross-Version Analysis

- Up to 18 versions
- 3 versions manually annotated

<table>
<thead>
<tr>
<th>No.</th>
<th>Conductor</th>
<th>Recording</th>
<th>hh:mm:ss</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Barenboim</td>
<td>1991–92</td>
<td>14:54:55</td>
</tr>
<tr>
<td>4</td>
<td>Furtwängler</td>
<td>1953</td>
<td>15:04:22</td>
</tr>
<tr>
<td>5</td>
<td>Haitink</td>
<td>1988–91</td>
<td>14:27:10</td>
</tr>
<tr>
<td>6</td>
<td>Janowski</td>
<td>1980–83</td>
<td>14:08:34</td>
</tr>
<tr>
<td>7</td>
<td>Karajan</td>
<td>1967–70</td>
<td>14:58:08</td>
</tr>
<tr>
<td>8</td>
<td>Keilberth/Furtwängler</td>
<td>1952–54</td>
<td>14:19:56</td>
</tr>
<tr>
<td>9</td>
<td>Krauss</td>
<td>1953</td>
<td>14:12:27</td>
</tr>
<tr>
<td>10</td>
<td>Levine</td>
<td>1987–89</td>
<td>15:21:52</td>
</tr>
<tr>
<td>11</td>
<td>Neuhold</td>
<td>1993–95</td>
<td>14:04:35</td>
</tr>
<tr>
<td>12</td>
<td>Sawallisch</td>
<td>1989</td>
<td>14:06:50</td>
</tr>
<tr>
<td>13</td>
<td>Solti</td>
<td>1958–65</td>
<td>14:36:58</td>
</tr>
<tr>
<td>14</td>
<td>Swarowsky</td>
<td>1968</td>
<td>14:56:34</td>
</tr>
<tr>
<td>15</td>
<td>Thielemann</td>
<td>2011</td>
<td>14:31:13</td>
</tr>
<tr>
<td>16</td>
<td>Weigle</td>
<td>2010–12</td>
<td>14:48:46</td>
</tr>
</tbody>
</table>
Cross-Version Analysis

- Idea: Use analysis results based on different interpretations (versions)
- Tonal characteristics should not depend on interpretation
  → Test reliability of the method

- Visualize consistency with gray scheme
Die Walküre WWV 86 B

Act 1

Act 2

Act 3
Die Walküre WWV 86 B

Act 1
Die Walküre WWV 86 B

- Act 1, measures 955–1012
- Sieglinde’s narration
Die Walküre WWV 86 B

- Act 1, measures 955–1012
- Sieglinde’s narration
Die Walküre WWV 86 B

Act 1

Act 2

Act 3
Die Walküre WWV 86 B

- Act 3, measures 724–789
- Wotan’s punishment
Die Walküre WWV 86 B

- Act 3, measures 724–789
- Wotan’s punishment
Die Walküre WWV 86 B
Exploring Tonal-Dramatic Relationships

- Histograms of Analysis over time

<table>
<thead>
<tr>
<th>Opera</th>
<th>WWV 86</th>
<th>Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Das Rheingold</td>
<td>A</td>
<td>3897</td>
</tr>
<tr>
<td>Die Walküre</td>
<td>B</td>
<td>5322</td>
</tr>
<tr>
<td>Siegfried</td>
<td>C</td>
<td>6682</td>
</tr>
<tr>
<td>Götterdämmerung</td>
<td>D</td>
<td>6040</td>
</tr>
</tbody>
</table>
Exploring Tonal-Dramatic Relationships

Sword motif – *Die Walküre*

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Diatonic Scales

-5 -4 -3 -2 -1 0 +1 +2 +3 +4 +5 +6
Exploring Tonal-Dramatic Relationships

**Sword motif – Siegfried**

![Sword motif diagram](image)

- **A**
- **B**
- **C**
- **D**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Diatonic Scales**

![Diatonic Scales graph](image)
Exploring Tonal-Dramatic Relationships

Valhalla motif – *Das Rheingold*

![Chord Chart]

- [Image of Chord Chart]

Chords:
- B♭m
- Db
- Fm
- A♭
- G♭m
- B♭
- Dm
- F
- Am
- Cm
- E♭
- Bm
- F♯m
- C♯m
- G♯m
- D♯m

[Sound Icons]
Exploring Tonal-Dramatic Relationships

**Valhalla motif – Die Walküre**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Tonal Structures: Complexity

- Global chroma statistics (audio)
- 1567 – G. da Palestrina, Missa de Beata Virgine, Credo

![Circle of fifths diagram](image)
Tonal Structures: Complexity

- Global chroma statistics (audio)
- 1725 – J. S. Bach, Orchestral Suite No. 4 BWV 1069, 1. Ouverture (D major)
Tonal Structures: Complexity

- Global chroma statistics (audio)
- 1783 – W. A. Mozart, „Linz“ symphony KV 425, 1. Adagio / Allegro (C major)

![Salience diagram showing pitch class distribution](chart.png)
Tonal Structures: Complexity

- Global chroma statistics (audio)
- 1883 – J. Brahms, Symphony No. 3, 1. Allegro con brio (F major)

![Circle of fifths diagram]

*Circle of fifths →*
Tonal Structures: Complexity

- Global chroma statistics (audio)
- 1940 – A. Webern, Variations for Orchestra op. 30
Tonal Structures: Complexity

- Realization of complexity measure $\Gamma$
  - Entropy / Flatness measures
  - Distribution over *Circle of Fifths*

$\Gamma = 0$

$\Gamma = 1$

$0 < \Gamma < 1$

- Relating to different time scales!

$\Gamma = \sqrt{1 - r}$
Tonal Structures: Complexity

Weiss / Müller, *Quantifying and Visualizing Tonal Complexity*, CIM 2014
Tonal Structures: Complexity

L. van Beethoven
Sonata Op. 2, No. 3
1st movement
Tonal Structures: Complexity

Beethoven Sonatas, 1st movements

- Op. 2, No. 3
- Op. 57, No. 1 ("Appassionata")
- Op. 106, No. 1 ("Hammerklavier")