Workshop HfM Karlsruhe

Music Information Retrieval

Music Synchronization

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

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

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

Accompanying website:
www.music-processing.de
### 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
# 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
Chapter 3: Music Synchronization

3.1 Audio Features
3.2 Dynamic Time Warping
3.3 Applications
3.4 Further Notes

As a first music processing task, we study in Chapter 3 the problem of music synchronization. The objective is to temporally align compatible representations of the same piece of music. Considering this scenario, we explain the need for musically informed audio features. In particular, we introduce the concept of chroma-based music features, which capture properties that are related to harmony and melody. Furthermore, we study an alignment technique known as dynamic time warping (DTW), a concept that is applicable for the analysis of general time series. For its efficient computation, we discuss an algorithm based on dynamic programming—a widely used method for solving a complex problem by breaking it down into a collection of simpler subproblems.
Music Data
Music Data
Music Data
### Music Data

Various interpretations – Beethoven’s Fifth

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bernstein</td>
<td></td>
</tr>
<tr>
<td>Karajan</td>
<td></td>
</tr>
<tr>
<td>Gould (piano)</td>
<td></td>
</tr>
<tr>
<td>MIDI (piano)</td>
<td></td>
</tr>
</tbody>
</table>
Music Synchronization: Audio-Audio

**Given:** Two different audio recordings 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

Beethoven’s Fifth

Karajan

Gould
Music Synchronization: Audio-Audio

Beethoven’s Fifth

Karajan

Gould
Music Synchronization: Audio-Audio

Application: Interpretation Switcher
Music Synchronization: Audio-Audio

Two main steps:

1.) Audio features
   - Robust but discriminative
   - Chroma features
   - Robust to variations in instrumentation, timbre, dynamics
   - Correlate to harmonic progression

2.) Alignment procedure
   - Deals with local and global tempo variations
   - Needs to be efficient
Music Synchronization: Audio-Audio

Beethoven’s Fifth

Karajan

Gould
Music Synchronization: Audio-Audio

Beethoven’s Fifth

Karajan

Gould
Music Synchronization: Audio-Audio

Beethoven’s Fifth

Karajan

Gould
Music Synchronization: Audio-Audio

Beethoven’s Fifth

Karajan

Gould
Music Synchronization: Audio-Audio

Beethoven’s Fifth

Karajan

Gould
Music Synchronization: Audio-Audio

Karajan

Gould
Music Synchronization: Audio-Audio

Cost matrix
Music Synchronization: Audio-Audio

Cost matrix
Music Synchronization: Audio-Audio

Optimal alignment (cost-minimizing warping path)
Music Synchronization: Audio-Audio

Cost matrix
Music Synchronization: Audio-Audio

Optimal alignment (cost-minimizing warping path)
Music Synchronization: Audio-Audio

Optimal alignment (cost-minimizing warping path)
Music Synchronization: Audio-Audio

How to compute the alignment?

⇒ Cost matrices

⇒ Dynamic programming

⇒ Dynamic Time Warping (DTW)
Freude, schoener Götterfunken, 
Tochter aus Elysium, 
Wir betreten feuertrunken, 
Himmlische dein Heiligtum.
Deine Zauber binden wieder, 
Was die Mode streng geteilt; 
Alle Menschen werden Brüder, 
Wo dein sanfter Flügel weilt.

Wem der grosse Wurf gelungen, 
Eines Freundes Freund zu sein, 
Wer ein holdes Weib errungen, 
Mische seine Jubel ein!
Music Synchronization: MIDI-Audio
Music Synchronization: MIDI-Audio

MIDI = meta data

Automated annotation

Audio recording

Sonification of annotations
Music Synchronization: MIDI-Audio

- Automated audio annotation
- Accurate audio access after MIDI-based retrieval
- Automated tracking of MIDI note parameters during audio playback
- Performance analysis
Music Synchronization: MIDI-Audio

MIDI = reference (score)

Tempo information

Audio recording
Performance Analysis: Tempo Curves

Alignment

Local tempo

Reference version

Reference version
Performance Analysis: Tempo Curves

Alignment

Local tempo

1 beat lasting 2 seconds \(\triangleq\) 30 BPM
Performance Analysis: Tempo Curves

Alignment

Reference version

Local tempo

Performed version

Tempo (BPM)

0 120 180 240

0 1 2 3 4 5

Time (beats)

Performed version

0 1 2 3 4 5

Time (beats)

1 beat lasting 1 seconds \( \triangleq \) 60 BPM
Performance Analysis: Tempo Curves

Alignment

Local tempo

1 beat lasting 0.4 seconds ≡ 150 BPM
Performance Analysis: Tempo Curves

Alignment

Tempo curve is obtained by interpolation
Performance Analysis: Tempo Curves

Schumann: Träumerei

Performance:

![Graph showing tempo curves over time](image-url)
Performance Analysis: Tempo Curves

Schumann: Träumerei

Score (reference):

Performance:
Performance Analysis: Tempo Curves

Schumann: Träumerei

Score (reference):

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\end{array}
\]

Performance:

Strategy: Compute score-audio synchronization and derive tempo curve
Performance Analysis: Tempo Curves

Schumann: Träumerei

Score (reference):

Tempo curve:
Performance Analysis: Tempo Curves

Schumann: Träumerei

Score (reference):

Tempo curves:
Performance Analysis: Tempo Curves

Schumann: Träumerei

Score (reference):

Tempo curves:
Performance Analysis: Tempo Curves

Schumann: Träumerei

Score (reference):

Tempo curves:
Performance Analysis: Tempo Curves

Schumann: Träumerei

What can be done if no reference is available?
Performance Analysis: Tempo Curves

Schumann: Träumerei

What can be done if no reference is available?

→ Tempo and Beat Tracking

Tempo curves:
Music Synchronization: Image-Audio
Music Synchronization: Image-Audio
Music Synchronization: **Image-Audio**

Convert data into common mid-level feature representation
Music Synchronization: Image-Audio

Image Processing: Optical Music Recognition

Convert data into common mid-level feature representation
Music Synchronization: Image-Audio

Image Processing: Optical Music Recognition

Convert data into common mid-level feature representation

Audio Processing: Fourier Analyse
Music Synchronization: Image-Audio

Image Processing: Optical Music Recognition

Audio Processing: Fourier Analyse
Music Synchronization: Image-Audio

Application: Score Viewer
Music Synchronization: Lyrics-Audio

Ich träumte von bunten Blumen, so wie sie wohl blühen im Mai
Music Synchronization: Lyrics-Audio

Ich träumte von bunten Blumen, so wie sie wohl blühen im Mai

Extremely difficult!
Music Synchronization: Lyrics-Audio

Lyrics-Audio → Lyrics-MIDI + MIDI-Audio

Ich träumte von bunten Blumen, so wie sie wohl blühen im Mai
Music Synchronization: Lyrics-Audio

Lyrics-Audio → Lyrics-MIDI + MIDI-Audio

Ich träumte von bunten Blumen, so wie sie wohl blühen im Mai
Score-Informed Source Separation
Score-Informed Source Separation
Score-Informed Source Separation
Score-Informed Source Separation

Experimental results for separating left and right hands for piano recordings:

<table>
<thead>
<tr>
<th>Composer</th>
<th>Piece</th>
<th>Database</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bach</td>
<td>BWV 875, Prelude</td>
<td>SMD</td>
<td></td>
</tr>
<tr>
<td>Chopin</td>
<td>Op. 28, No. 15</td>
<td>SMD</td>
<td></td>
</tr>
<tr>
<td>Chopin</td>
<td>Op. 64, No. 1</td>
<td>European Archive</td>
<td></td>
</tr>
</tbody>
</table>
Score-Informed Source Separation

Audio editing
Dynamic Time Warping
Dynamic Time Warping

- Well-known technique to find an optimal alignment between two given (time-dependent) sequences under certain restrictions.

- Intuitively, sequences are warped in a non-linear fashion to match each other.

- Originally used to compare different speech patterns in automatic speech recognition
Dynamic Time Warping

Sequence $X$  
\[x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7 \quad x_8 \quad x_9\]

Sequence $Y$  
\[y_1 \quad y_2 \quad y_3 \quad y_4 \quad y_5 \quad y_6 \quad y_7\]
Dynamic Time Warping

Time alignment of two time-dependent sequences, where the aligned points are indicated by the arrows.
Dynamic Time Warping

Time alignment of two time-dependent sequences, where the aligned points are indicated by the arrows.
Dynamic Time Warping

The objective of DTW is to compare two (time-dependent) sequences

\[ X := (x_1, x_2, \ldots, x_N) \]

of length \( N \in \mathbb{N} \) and

\[ Y := (y_1, y_2, \ldots, y_M) \]

of length \( M \in \mathbb{N} \). Here,

\[ x_n, y_m \in \mathcal{F}, \; n \in [1 : N], \; m \in [1 : M], \]

are suitable features that are elements from a given feature space denoted by \( \mathcal{F} \).
Dynamic Time Warping

To compare two different features \( x, y \in \mathcal{F} \) one needs a local cost measure which is defined to be a function

\[
c : \mathcal{F} \times \mathcal{F} \rightarrow \mathbb{R}_{\geq 0}
\]

Typically, \( c(x, y) \) is small (low cost) if \( x \) and \( y \) are similar to each other, and otherwise \( c(x, y) \) is large (high cost).
Dynamic Time Warping

Evaluating the local cost measure for each pair of elements of the sequences \( \mathbf{X} \) and \( \mathbf{Y} \) one obtains the cost matrix

\[
C \in \mathbb{R}^{N \times M}
\]

defined by

\[
C(n, m) := c(x_n, y_m).
\]

Then the goal is to find an alignment between \( \mathbf{X} \) and \( \mathbf{Y} \) having minimal overall cost. Intuitively, such an optimal alignment runs along a “valley” of low cost within the cost matrix \( C \).
Dynamic Time Warping

Cost matrix $C$
Dynamic Time Warping

Cost matrix $C$

$C(5, 6)$
Dynamic Time Warping

Cost matrix $C$
Dynamic Time Warping

Cost matrix $C$
Dynamic Time Warping

The next definition formalizes the notion of an alignment.

A **warping path** is a sequence $p = (p_1, \ldots, p_L)$ with

$$p_\ell = (n_\ell, m_\ell) \in [1 : N] \times [1 : M]$$

for $\ell \in [1 : L]$ satisfying the following three conditions:

- **Boundary condition:** $p_1 = (1, 1)$ and $p_L = (N, M)$
- **Monotonicity condition:** $n_1 \leq n_2 \leq \ldots \leq n_L$ and $m_1 \leq m_2 \leq \ldots \leq m_L$
- **Step size condition:** $p_{\ell+1} - p_\ell \in \{(1, 0), (0, 1), (1, 1)\}$ for $\ell \in [1 : L - 1]$
Dynamic Time Warping

Warping path

Each matrix entry (cell) corresponds to a pair of indices.

Cell = (6, 3)

Boundary cells:
\[p_1 = (1, 1)\]
\[p_L = (N, M) = (9, 7)\]
Dynamic Time Warping

Warping path

Correct warping path

Sequence $X$: $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9$

Sequence $Y$: $y_1, y_2, y_3, y_4, y_5, y_6, y_7$
Dynamic Time Warping

Warping path

Violation of boundary condition

Sequence X

\[
\begin{align*}
x_1 & \quad x_2 & \quad x_3 & \quad x_4 & \quad x_5 & \quad x_6 & \quad x_7 & \quad x_8 & \quad x_9 \\
\end{align*}
\]

Sequence Y

\[
\begin{align*}
y_1 & \quad y_2 & \quad y_3 & \quad y_4 & \quad y_5 & \quad y_6 & \quad y_7 \\
\end{align*}
\]
Dynamic Time Warping

Warping path

Violation of monotonicity condition
Dynamic Time Warping

Warping path

Violation of step size condition

Sequence $X$ and $Y$ with a violated step size condition.
Dynamic Time Warping

The total cost \( c_p(X, Y) \) of a warping path \( p \) between \( X \) and \( Y \) with respect to the local cost measure \( c \) is defined as

\[
c_p(X, Y) := \sum_{\ell=1}^{L} c(x_{n_{\ell}}, y_{m_{\ell}})
\]

Furthermore, an optimal warping path between \( X \) and \( Y \) is a warping path \( p^* \) having minimal total cost among all possible warping paths. The DTW distance \( \text{DTW}(X, Y) \) between \( X \) and \( Y \) is then defined as the total cost of \( p^* \)

\[
\text{DTW}(X, Y) := c_{p^*}(X, Y) = \min\{c_p(X, Y) \mid p \text{ is a warping path}\}
\]
Dynamic Time Warping

- The warping path $p^*$ is not unique (in general).

- DTW does (in general) not define a metric since it may not satisfy the triangle inequality.

- There exist exponentially many warping paths.

- How can $p^*$ be computed efficiently?
Dynamic Time Warping

**Notation:**

- $X(1:n) := (x_1, \ldots, x_n), \ 1 \leq n \leq N$
- $Y(1:m) := (y_1, \ldots, y_m), \ 1 \leq m \leq M$
- $D(n,m) := \text{DTW}(X(1:n), Y(1:m))$

The matrix $D$ is called the **accumulated cost matrix**.

The entry $D(n,m)$ specifies the cost of an optimal warping path that aligns $X(1:n)$ with $Y(1:m)$. 
Dynamic Time Warping

Lemma:

\((i)\) \quad D(N, M) = DTW(X, Y)

\((ii)\) \quad D(1, 1) = C(1, 1)

\((iii)\) \quad D(n, 1) = \sum_{k=1}^{n} C(k, 1)

\quad D(1, m) = \sum_{k=1}^{m} C(1, k)

\((iv)\) \quad D(n, m) = \min \left( \begin{array}{c}
D(n-1, m-1) \\
D(n-1, m) \\
D(n, m-1)
\end{array} \right) + C(n, m)

\text{for } n > 1, m > 1

Proof: \((i) \rightarrow (iii)\) are clear by definition
Dynamic Time Warping

Proof of (iv): Induction via \( n, m \):

Let \( n > 1, m > 1 \) and \( q = (q_1, \ldots, p_{L-1}, p_L) \) be an optimal warping path for \( X(1 : n) \) and \( Y(1 : m) \). Then \( q_L = (n, m) \) (boundary condition).

Let \( q_{L-1} = (a, b) \). The step size condition implies

\[
(a, b) \in \{(n - 1, m - 1), (n - 1, m), (n, m - 1)\}
\]

The warping path \( (q_1, \ldots, q_{L-1}) \) must be optimal for \( X(1 : a), Y(1 : b) \). Thus,

\[
D(n, m) = c_{(q_1, \ldots, q_{L-1})}(X(1 : a), Y(1 : b)) + C(n, m)
\]
Dynamic Time Warping

Accumulated cost matrix

Given the two feature sequences \( X \) and \( Y \), the matrix \( D \) is computed recursively.

- Initialize \( D \) using (ii) and (iii) of the lemma.
- Compute \( D(n, m) \) for \( n > 1, m > 1 \) using (iv).
- \( \text{DTW}(X, Y) = D(N, M) \) using (i).

Note:

- Complexity \( O(NM) \).
- Dynamic programming: “overlapping-subproblem property”
Dynamic Time Warping

Optimal warping path

Given to the algorithm is the accumulated cost matrix $D$. The optimal path $p^* = (p_1, \ldots, p_L)$ is computed in reverse order of the indices starting with $p_L = (N, M)$. Suppose $p_\ell = (n, m)$ has been computed. In case $(n, m) = (1, 1)$, one must have $\ell = 1$ and we are done. Otherwise,

$$p_{\ell-1} := \begin{cases} 
(1, m - 1), & \text{if } n = 1 \\
(n - 1, 1), & \text{if } m = 1 \\
\text{argmin}\{D(n - 1, m - 1),} & \text{otherwise,}
D(n - 1, m), D(n, m - 1)\},
\end{cases}$$

where we take the lexicographically smallest pair in case “argmin” is not unique.
Dynamic Time Warping

Summary

\[ D(n,m) = \text{DTW}(X,Y) \]

\[ D(n,1) \]

\[ D(n-1,m-1) \]

\[ D(n-1,m) \]

\[ D(1,m) \]

\[ D(n,m-1) \]

\[ m = 10 \]

\[ n = 6 \]
Dynamic Time Warping

Summary

**Algorithm: DTW**

**Input:** Cost matrix $C$ of size $N \times M$

**Output:** Accumulated cost matrix $D$

  - Optimal warping path $P^*$

**Procedure:** Initialize $(N \times M)$ matrix $D$ by

$$D(n, 1) = \sum_{k=1}^{n} C(k, 1) \text{ for } n \in [1 : N]$$

$$D(1, m) = \sum_{k=1}^{m} C(1, k) \text{ for } m \in [1 : M].$$

Then compute in a nested loop for $n = 2, \ldots, N$ and $m = 2, \ldots, M$:

$$D(n, m) = C(n, m) + \min \{ D(n-1, m-1), D(n-1, m), D(n, m-1) \}.$$ 

Set $\ell = 1$ and $q_\ell = (N, M)$. Then repeat the following steps until $q_\ell = (1, 1)$:

1. Increase $\ell$ by one and let $(n, m) = q_{\ell-1}$.
2. If $n = 1$, then $q_\ell = (1, m-1)$,
   - else if $m = 1$, then $q_\ell = (n-1, m)$,
   - else $q_\ell = \text{argmin} \{ D(n-1, m-1), D(n-1, m), D(n, m-1) \}$.
   (If "argmin" is not unique, take lexicographically smallest cell.)

Set $L = \ell$ and return $P^* = (q_L, q_{L-1}, \ldots, q_1)$ as well as $D$. 
Dynamic Time Warping

Example

\[ X = (1, 3, 3, 8, 1) \]
\[ Y = (2, 0, 0, 8, 7, 2) \]
\[ c(x, y) = |x - y|, \ x, y \in \mathbb{R} \]

\[
\begin{array}{cccccc}
1 & 1 & 1 & 7 & 6 & 1 \\
6 & 8 & 8 & 0 & 1 & 6 \\
1 & 3 & 3 & 5 & 4 & 1 \\
1 & 3 & 3 & 5 & 4 & 1 \\
1 & 1 & 1 & 7 & 6 & 1 \\
2 & 0 & 0 & 8 & 7 & 2 \\
\end{array}
\quad
\begin{array}{ccccccc}
10 & 10 & 11 & 14 & 13 & 9 \\
9 & 11 & 13 & 7 & 8 & 14 \\
3 & 5 & 7 & 10 & 12 & 13 \\
2 & 4 & 5 & 8 & 12 & 13 \\
1 & 2 & 3 & 10 & 16 & 17 \\
2 & 0 & 0 & 8 & 7 & 2 \\
\end{array}
\]

Alignment

Optimal warping path: \( P^* = ((1, 1), (2, 2), (3, 3), (4, 4), (4, 5), (5, 6)) \)
Dynamic Time Warping

Step size conditions

\[ \Sigma = \{(1,0), (0,1), (1,1)\} \]
Dynamic Time Warping

Step size conditions

\[ \Sigma = \{(2,1), (1,2), (1,1)\} \]
Dynamic Time Warping

Step size conditions
Dynamic Time Warping

- Computation via dynamic programming
- Memory requirements and running time: $O(NM)$
- Problem: Infeasible for large $N$ and $M$
- Example: Feature resolution 10 Hz, pieces 15 min
  $$\Rightarrow N, M \sim 10,000$$
  $$\Rightarrow N \cdot M \sim 100,000,000$$
Dynamic Time Warping

Global constraints

Sakoe-Chiba band

Itakura parallelogram
Dynamic Time Warping

Global constraints

Sakoe-Chiba band

Itakura parallelogram

Problem: Optimal warping path not in constraint region
Dynamic Time Warping

Multiscale approach

Compute optimal warping path on coarse level
Dynamic Time Warping

Multiscale approach

Project on fine level
Dynamic Time Warping

Multiscale approach

Specify constraint region
Dynamic Time Warping

Multiscale approach

Compute constrained optimal warping path
Dynamic Time Warping

Multiscale approach

- Suitable features?
- Suitable resolution levels?
- Size of constraint regions?

Good trade-off between efficiency and robustness?

Suitable parameters depend very much on application!