



COMPUTERGESTÜTZTE MUSIKFORSCHUNG 1

Institut für Musikinformatik und Musikwissenschaft
Wintersemester 2025–26



Christophe Weis
christophe.weis@stud.hfm.eu

Woche 08
02.12.2025

Organisation

wöchentlich, Di. 14.30–16.00, K10 Raum 309

Modul **Music Processing**

- **BA MI (HF)/MW (EF), wiss. Schwerpunkt:** Pflicht (4. Semester)
- **BA MI (HF)/MW (EF), künstl. Schwerpunkt:** Wahlpflicht (6. Semester)
- **BA MW (HF)/MI (EF):** Pflicht (4. Semester) – reduzierter Arbeitsaufwand
- **BA MI/MW (KF):** Pflicht (4. Semester)
- **BA:** Wahlfach

Projektarbeit

- eine selbstständige praktische Arbeit aus den Bereichen Musikkodierung, symbolbasierte Musikverarbeitung und –analyse mit Dokumentation (ca. 5000 Zeichen)

Übungen

- Tutorin: Joanna Friedrich-Sroka
- wöchentlich, Di. 11.15–12.45, K10 Raum 309

The background is black with four pink, wavy, brush-stroke-like lines in the corners, forming a larger 'X' shape. The lines have a gradient from light pink to white.

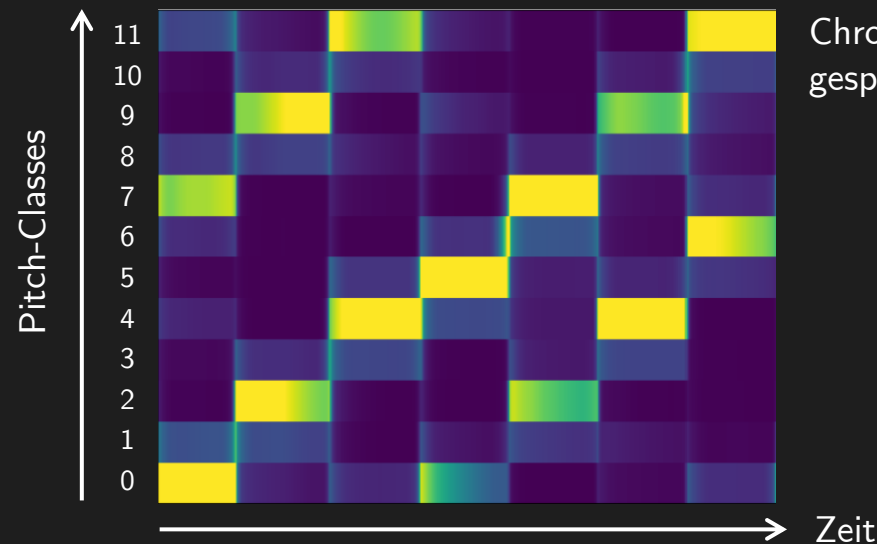
16.

Chromagramme

Chromagramme

- *Idee:*
 - Darstellung, welche *Pitch-Classes* (auch *Chroma* genannt) über die Zeit aktiv sind.
 - Zusammenfassung aller zu einer bestimmten Pitch-Class gehörenden spektralen Informationen in einem einzigen Chroma-Koeffizienten.
- Beispiel für die Erstellung von Chromagrammen von Audiosignalen mithilfe von *Librosa*:

```
my_audio_file = "my_audio_file.wav"
y, sr = librosa.load(my_audio_file)
chroma = librosa.feature.chroma_stft(y=y, sr=sr)
```



Chromagramm einer auf einer Gitarre
gespielten C-Dur-Tonleiter.

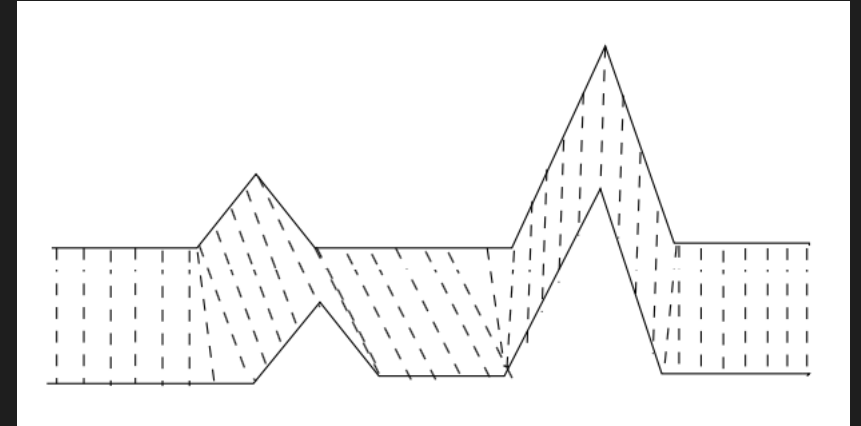
The background of the slide is black, decorated with several thick, wavy, pink lines that resemble stylized, abstract waves or brushstrokes. These lines are positioned in the corners and along the edges, framing the central text.

17.

Dynamic Time Warping (Teil 1)

Dynamic Time Warping (DTW)

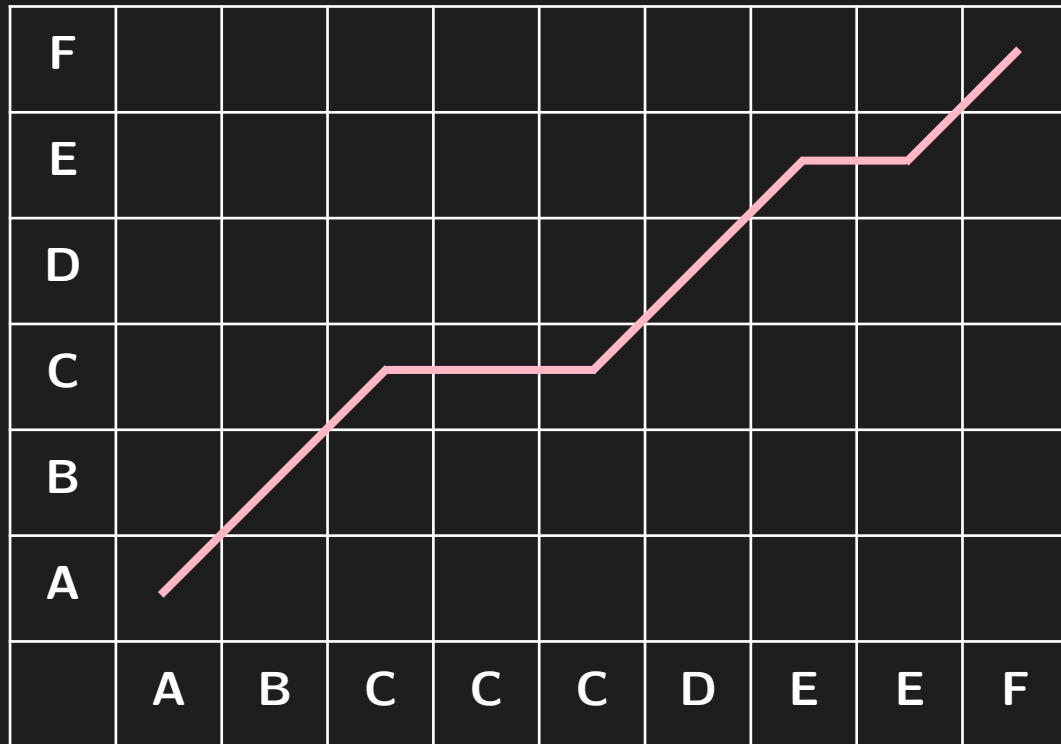
- Alignment-Algorithmus zum Vergleichen von 2 **zeitlichen Sequenzen**
- Zu vergleichende Sequenzen **können asynchron sein**, z. B. unterschiedliche musikalische Tempi besitzen
- *Ziel:*
Jeder zeitliche Index der ersten Sequenz soll mit einem (oder mehreren) entsprechenden Index der zweiten Sequenz synchronisiert werden (und umgekehrt)
- Implementierung mithilfe einer Cost/Score-Matrix (wie andere Alignment-Verfahren)
- Anwendungen: automatische Spracherkennung, Music Information Retrieval, Bioinformatik, ...



https://en.wikipedia.org/wiki/Dynamic_time_warping#/media/File:Dynamic_time_warping.png

Einfaches Beispiel

- *Ausgangslage:* 2 ähnliche Zeichen-Sequenzen, die „unterschiedlich schnell“ verlaufen
 - **ABCDEF**
 - **ABCCCDEEF**
- *Ziel:* Darstellung eines Mappings zwischen den beiden Zeichen-Sequenz mithilfe eines Pfades (eines sogenannten „Warping-Pfades“)



- Ein solches Mapping „**align**“ jedes der Zeichen der 1. Sequenz mit einem oder mehreren Zeichen der 2. Sequenz.
- In der Regel erlaubt man für einen Warping-Pfad nur eine bestimmte Anzahl an **Schritt-Möglichkeiten**, z. B.
 - $(1, 0)$ = „nach rechts“
 - $(0, 1)$ = „nach oben“
 - $(1, 1)$ = „diagonal“

Einfaches Beispiel

- 1. *Schritt*: Erstellung einer **Cost-Matrix (oder Score-Matrix)**
Diese Matrix enthält (in diesem Fall) eine Distanz zwischen allen Zeichen, die in den beiden Sequenzen vorkommen.

Cost-Matrix

F	1	1	1	1	1	1	1	1	0
E	1	1	1	1	1	1	0	0	1
D	1	1	1	1	1	0	1	1	1
C	1	1	0	0	0	1	1	1	1
B	1	0	1	1	1	1	1	1	1
A	0	1	1	1	1	1	1	1	1
	A	B	C	C	C	D	E	E	F

In diesem einfachen Beispiel definieren wir diese Distanz ganz trivial wie folgt:

$$d(a, b) = 0, \text{ falls } a = b$$
$$d(a, b) = 1, \text{ falls } a \neq b$$

Einfaches Beispiel

- 2. Schritt: Wie gut ist ein Warping-Path?
 - Berechnung der „**Accumulated Costs**“ = die Summe aller vom Warping-Path „aufgesammelten“ Kosten
 - Abspeichern der Accumulated Costs in einer **Accumulated-Cost-Matrix**
 - Berechnung dieser Accumulated-Cost-Matrix ausgehend von der Cost-Matrix

Cost-Matrix

F	1	1	1	1	1	1	1	1	0
E	1	1	1	1	1	1	0	0	1
D	1	1	1	1	1	0	1	1	1
C	1	1	0	0	0	1	1	1	1
B	1	0	1	1	1	1	1	1	1
A	0	1	1	1	1	1	1	1	1
	A	B	C	C	C	D	E	E	F

Accumulated-Cost-Matrix

F	?	?	?	?	?	?	?	?	?
E	?	?	?	?	?	?	?	?	?
D	?	?	?	?	?	?	?	?	?
C	?	?	?	?	?	?	?	?	?
B	?	?	?	?	?	?	?	?	?
A	?	?	?	?	?	?	?	?	?
	A	B	C	C	C	D	E	E	F

Einfaches Beispiel

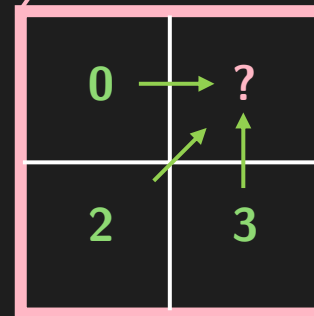
Cost-Matrix

F	1	1	1	1	1	1	1	1	0
E	1	1	1	1	1	1	0	0	1
D	1	1	1	1	1	0	1	1	1
C	1	1	0	0	0	1	1	1	1
B	1	0	1	1	1	1	1	1	1
A	0	1	1	1	1	1	1	1	1
	A	B	C	C	C	D	E	E	F

Accumulated-Cost-Matrix

F	5	4	3						
E	4	3	2						
D	3	2	1						
C	2	1	0	0					
B	1	0	1	2	3				
A	0	1	2	3	4	5	6	7	8
	A	B	C	C	C	D	E	E	F

$$? = 0 + \min \begin{cases} 0 \rightarrow \\ 2 \nearrow \\ 3 \uparrow \end{cases}$$



Einfaches Beispiel

Cost-Matrix

F	1	1	1	1	1	1	1	1	0
E	1	1	1	1	1	1	0	0	1
D	1	1	1	1	1	0	1	1	1
C	1	1	0	0	0	1	1	1	1
B	1	0	1	1	1	1	1	1	1
A	0	1	1	1	1	1	1	1	1
	A	B	C	C	C	D	E	E	F

Accumulated-Cost-Matrix

F	5	4	3						
E	4	3	2						
D	3	2	1						
C	2	1	0	0	0				
B	1	0	1	2	3				
A	0	1	2	3	4	5	6	7	8
	A	B	C	C	C	D	E	E	F

0	0
2	3

Einfaches Beispiel

Cost-Matrix

F	1	1	1	1	1	1	1	1	0
E	1	1	1	1	1	1	0	0	1
D	1	1	1	1	1	0	1	1	1
C	1	1	0	0	0	1	1	1	1
B	1	0	1	1	1	1	1	1	1
A	0	1	1	1	1	1	1	1	1
	A	B	C	C	C	D	E	E	F

Accumulated-Cost-Matrix

F	5	4	3	3	3	2	1	1	0
E	4	3	2	2	2	1	0	0	1
D	3	2	1	1	1	0	1	2	3
C	2	1	0	0	0	1	2	3	4
B	1	0	1	2	3	4	5	6	7
A	0	1	2	3	4	5	6	7	8
	A	B	C	C	C	D	E	E	F

Einfaches Beispiel

- 3. Schritt: **Optimaler Warping-Pfad** durch **Backtracking** in der **Accumulated-Cost-Matrix**:
 - Start „oben rechts“, Ziel „unten links“
 - Zu jedem Zeitpunkt:
 - Einhalten der „Schritt-Möglichkeiten“: $(1, 0)$, $(0, 1)$ und $(1, 1)$
 - Auswahl des **Eintrags mit den niedrigsten Accumulated-Costs**

Accumulated-Cost-Matrix

F	5	4	3	3	3	2	1	1	0
E	4	3	2	2	2	1	0	0	1
D	3	2	1	1	1	0	1	2	3
C	2	1	0	0	0	1	2	3	4
B	1	0	1	2	3	4	5	6	7
A	0	1	2	3	4	5	6	7	8
	A	B	C	C	C	D	E	E	F

