



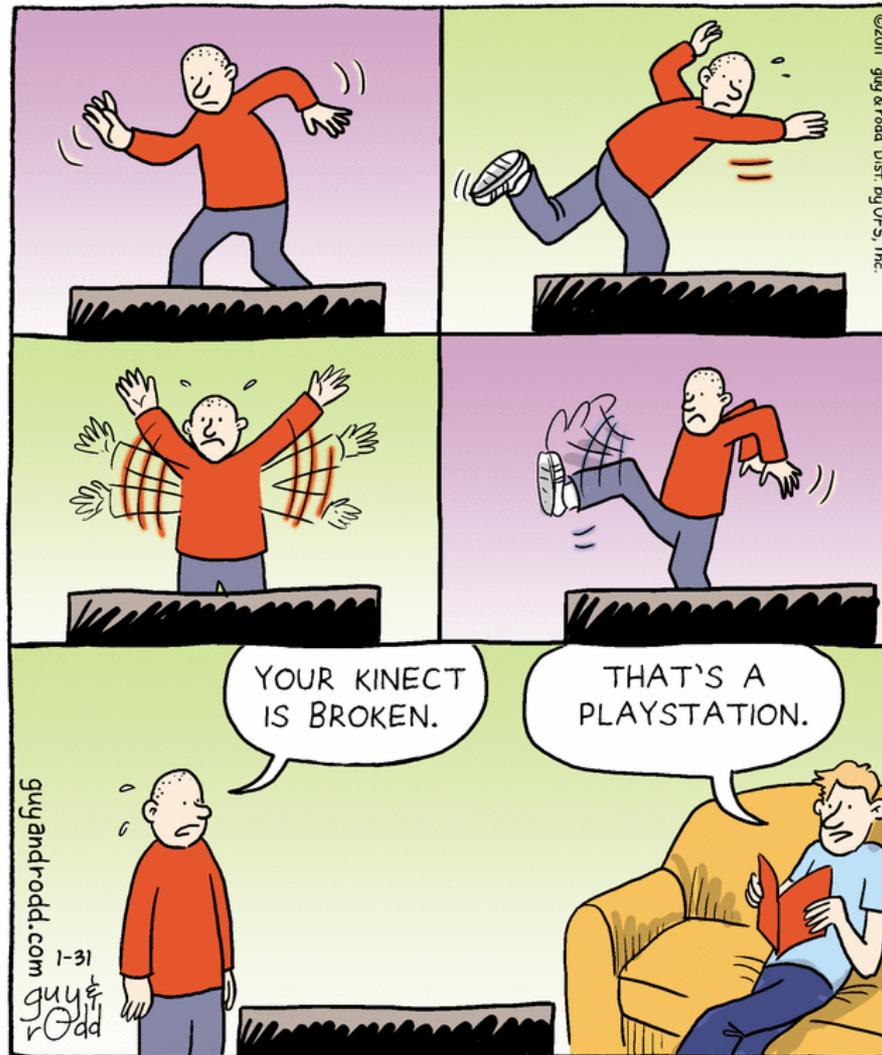
**TECHNISCHE
UNIVERSITÄT
DRESDEN**

Fakultät Informatik Institut für Software- und Multimediatechnik, Professur für Computergraphik und Visualisierung

Gestenbasierte Steuerung von interaktiven Umgebungen mithilfe der Microsoft Kinect

Diplomverteidigung
Ludwig Schmutzler

Dresden, 31.01.2013



Quelle: <http://www.gocomics.com/brevity/2011/01/31>

- Verfolgen und Interpretieren von Nutzerbewegungen
→ Steuerung virtueller Objekte / Auslösen von Aktionen
- Microsoft Kinect ermöglicht das Nachverfolgen in 3D



Bild-Quelle: <http://www.xbox.com/de-DE/Kinect/GetStarted>

- Beschränkung auf *Hände* und *Arme*, Ausnutzung d. Potenzials der Kinect

Anforderungen:

- Robustheit gegenüber

Lichtverhältnissen

Kleidung

Körperproportionen

Kamera-/Nutzerstandort

- Beschränkung auf *Hände* und *Arme*, Ausnutzung d. Potenzials der Kinect

Anforderungen:

- Robustheit gegenüber

Lichtverhältnissen

Kleidung

Körperproportionen

Kamera-/Nutzerstandort



Kinect

- Beschränkung auf *Hände* und *Arme*, Ausnutzung d. Potenzials der Kinect

Anforderungen:

- Robustheit gegenüber



- Beschränkung auf *Hände* und *Arme*, Ausnutzung d. Potenzials der Kinect

Anforderungen:

- Robustheit gegenüber



- schnelle Berechnungen, Echtzeitfähigkeit
- Erkennung von arbiträren Gesten aus dem Bewegungsfluss heraus

besondere Herausforderungen:

- Entwickeln eines Gestenkonzeptes
- komplexes Zusammenspiel der einzelnen System-Komponenten realisieren
- Ungenauigkeiten des Trackings durch robustes Filtern ausgleichen

1. Grundlagen

Modellkomponenten, Gestenkonzept

2. Gestenerkennungssystem

Sensordaten, Posenmodell, Erkennungsverfahren

3. Diskussion

Ergebnisse, Bewertung, Ausblick

4. Demonstration

1. Grundlagen

Ein Blick in die reale Welt:

Gesten sind Kommunikationsmittel, wichtig für den Menschen

→ Beispiel

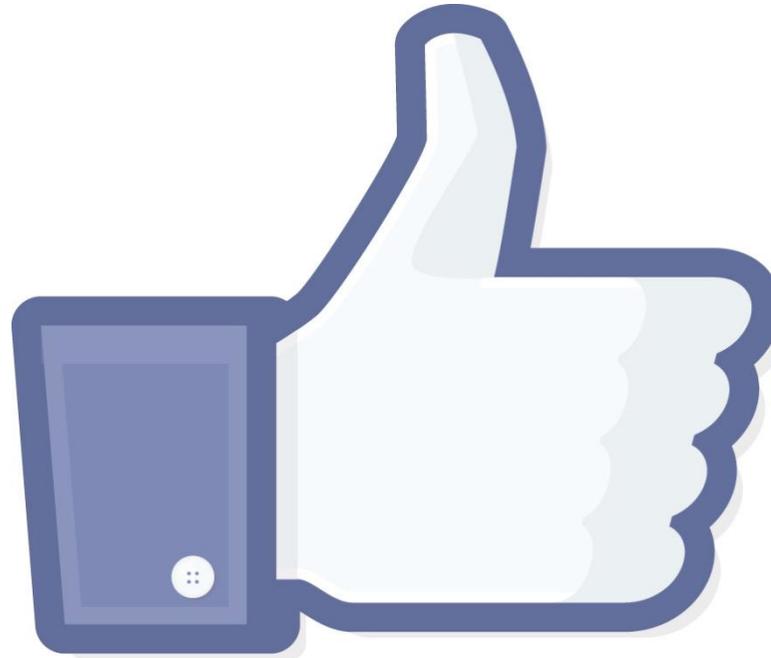


Bild-Quelle: <http://www.facebook.com/brandpermissions/logos.php>





Bedeutungsmodellierung ist
Aufgabe der Anwendung

→ Demonstrationsanwendung
im Rahmen der Arbeit

Geste



(Wieder-)Erkennen einer Struktur über ein Modell

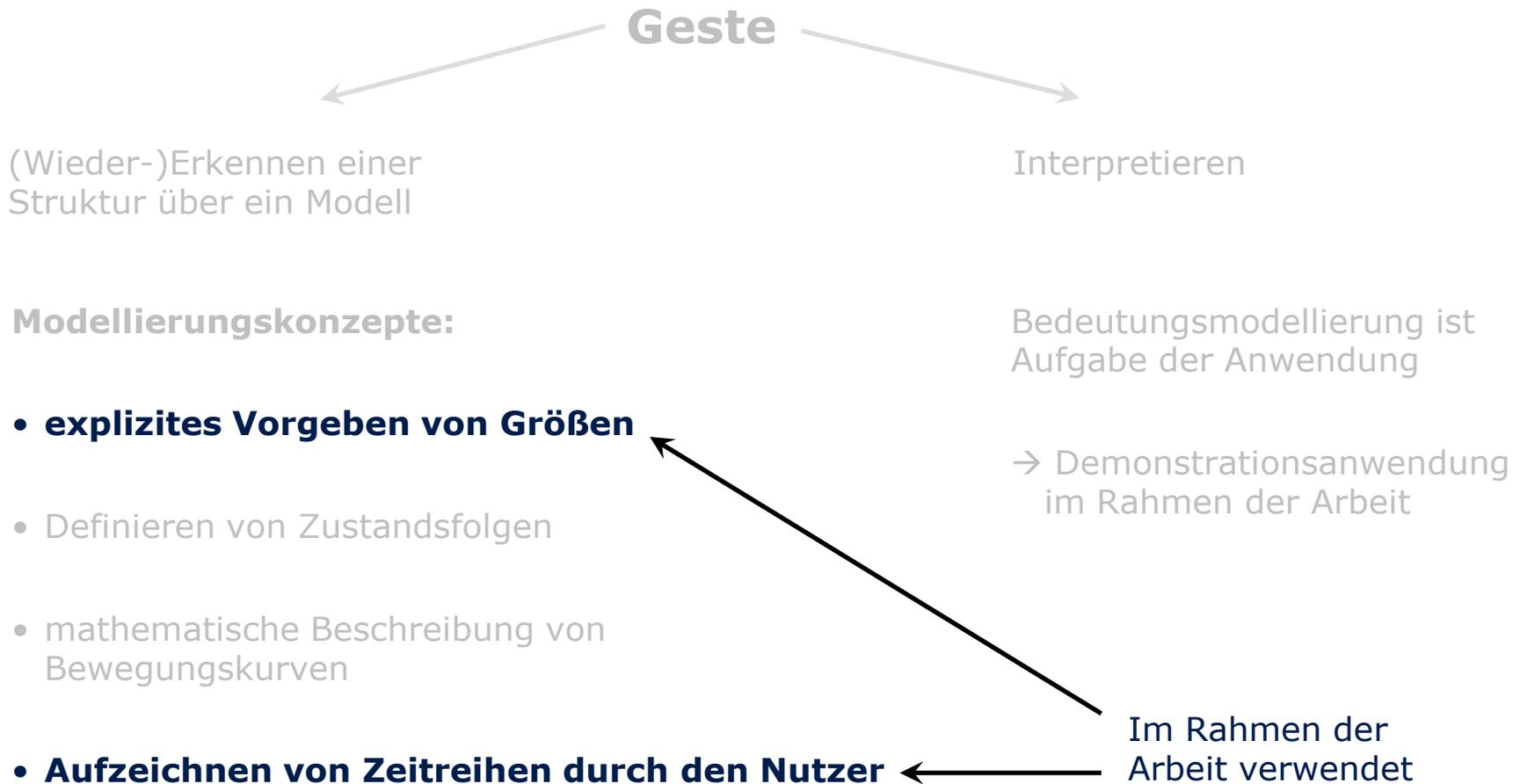
Interpretieren

Modellierungskonzepte:

- explizites Vorgeben von Größen
- Definieren von Zustandsfolgen
- mathematische Beschreibung von Bewegungskurven
- Aufzeichnen von Zeitreihen durch den Nutzer

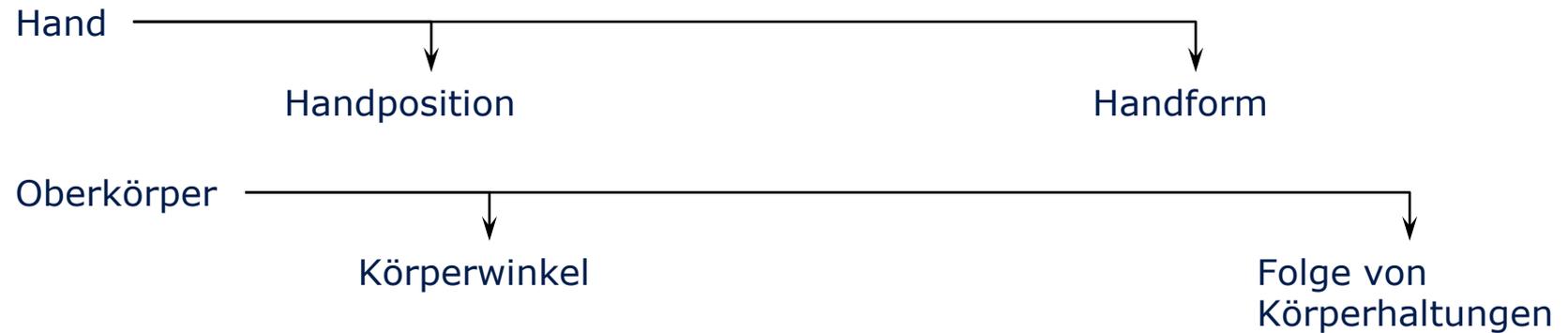
Bedeutungsmodellierung ist Aufgabe der Anwendung

→ Demonstrationsanwendung im Rahmen der Arbeit



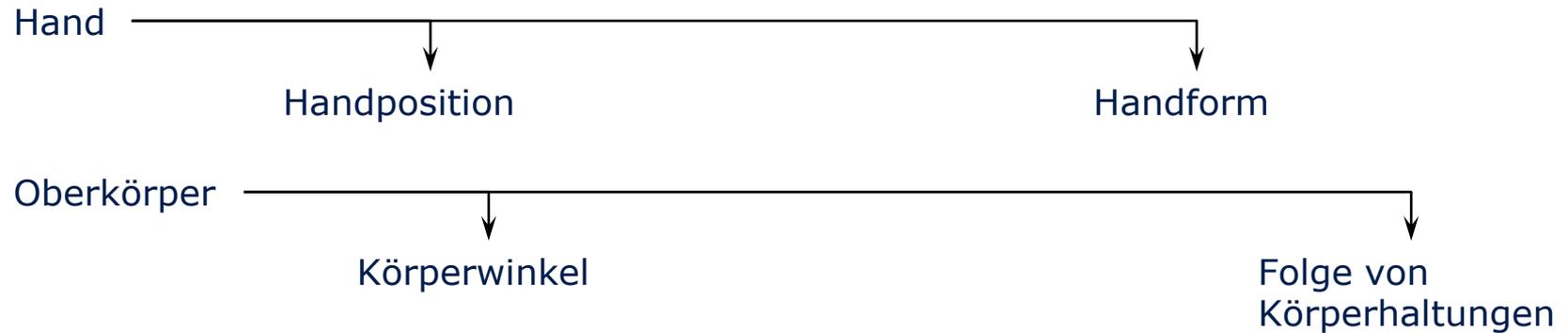
Noch etwas mehr Modellierung...

- modularer Mensch, funktionale Trennung von Hand und restl. Oberkörper



Noch etwas mehr Modellierung...

- modularer Mensch, funktionale Trennung von Hand und restl. Oberkörper



Gestenarten [KS05]:

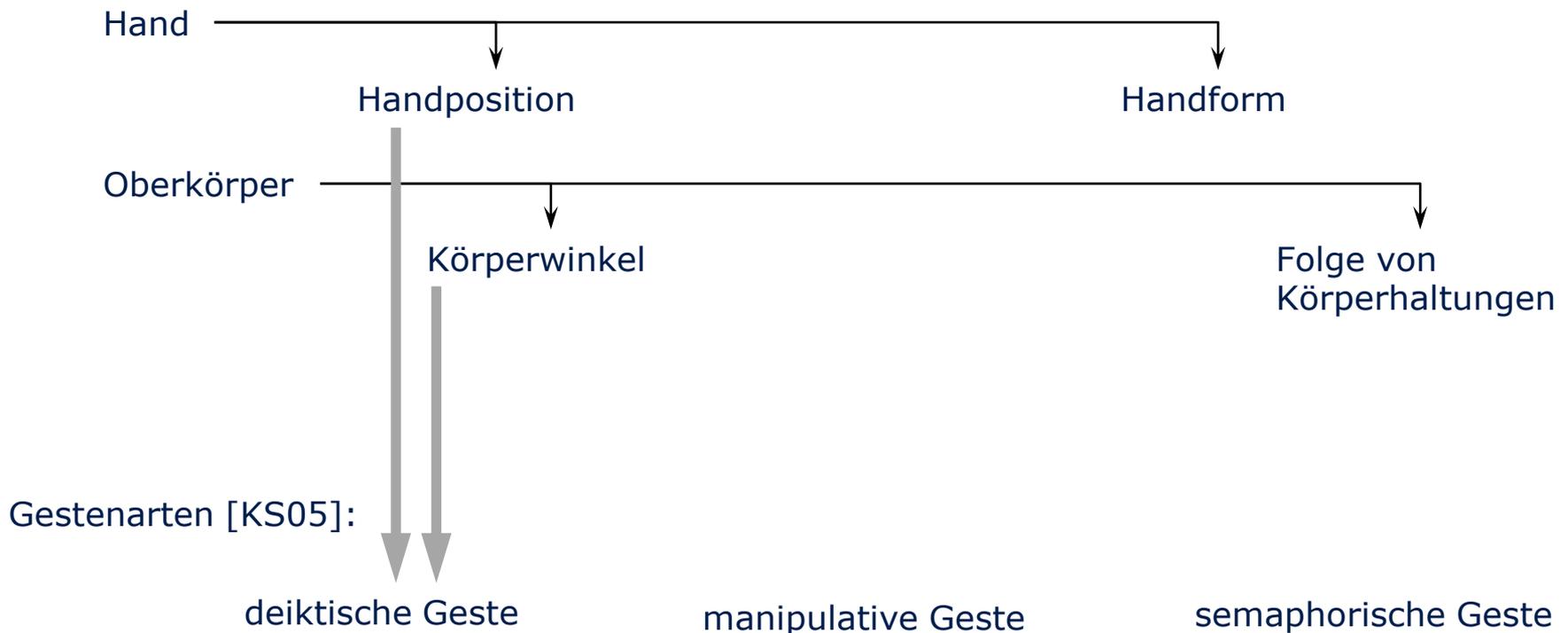
deiktische Geste

manipulative Geste

semaphorische Geste

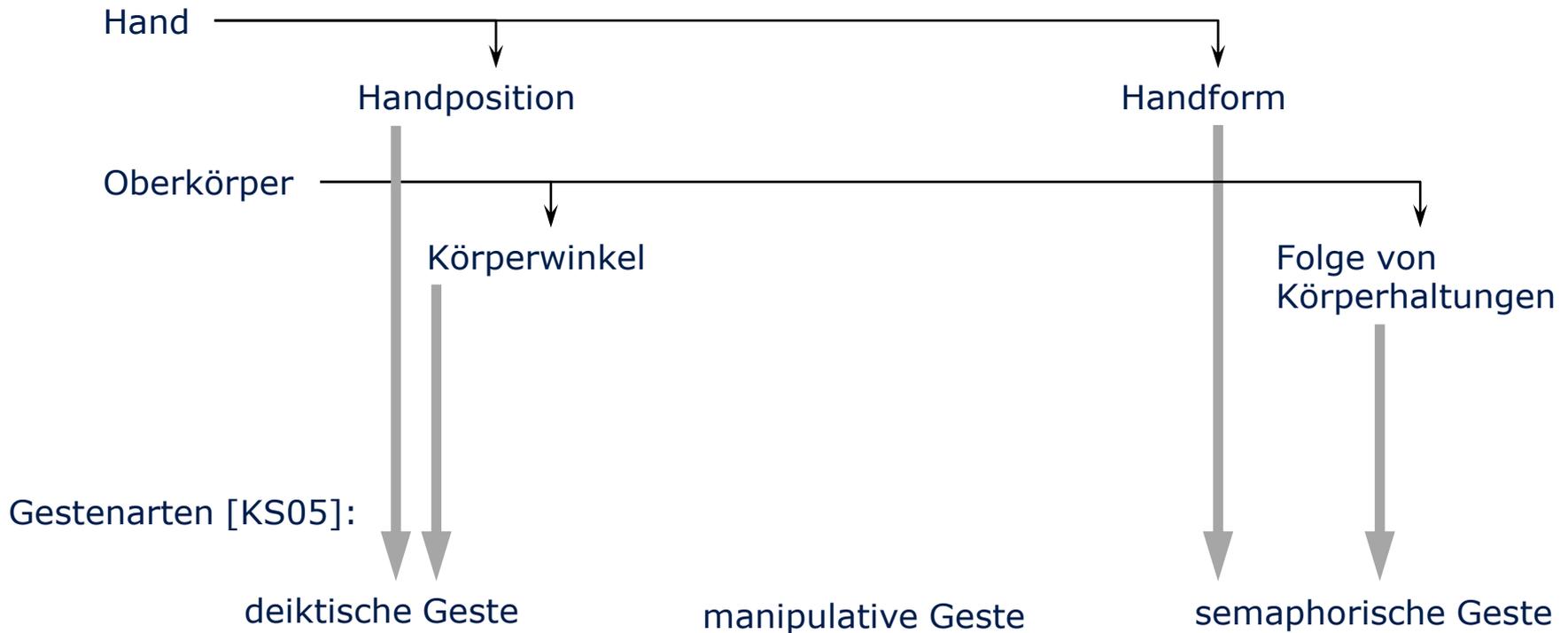
Noch etwas mehr Modellierung...

- modularer Mensch, funktionale Trennung von Hand und restl. Oberkörper



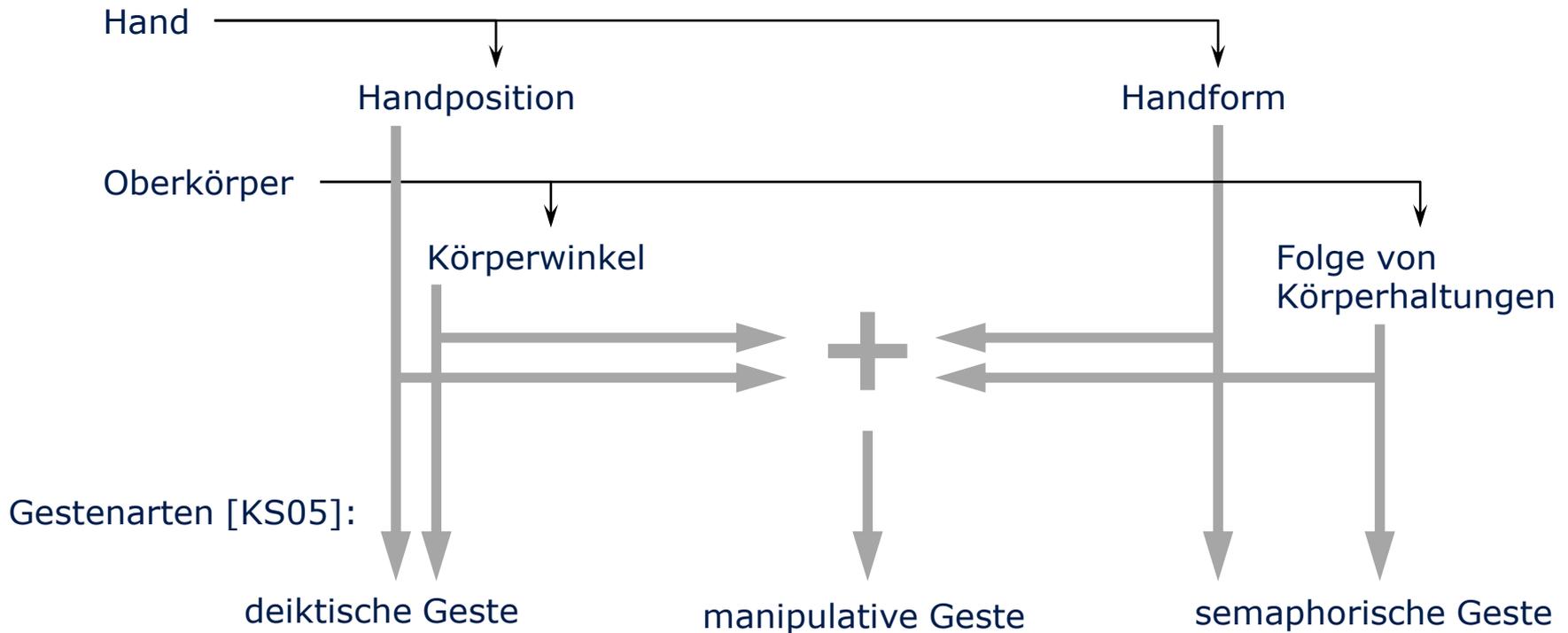
Noch etwas mehr Modellierung...

- modularer Mensch, funktionale Trennung von Hand und restl. Oberkörper



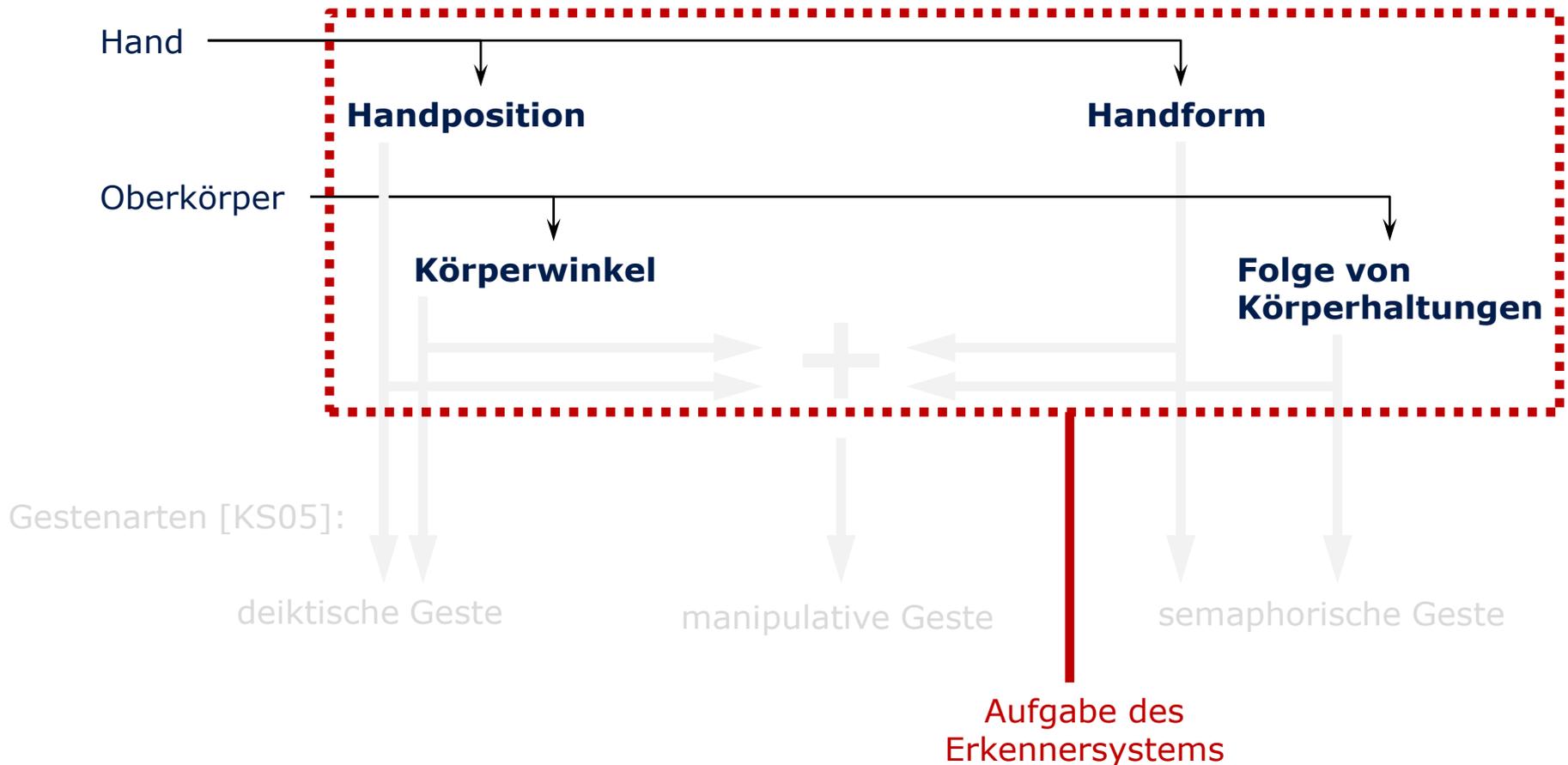
Noch etwas mehr Modellierung...

- modularer Mensch, funktionale Trennung von Hand und restl. Oberkörper



Noch etwas mehr Modellierung...

- modularer Mensch, funktionale Trennung von Hand und restl. Oberkörper



2. Gestenerkennungssystem

Gesamtsystem



Gesamtsystem



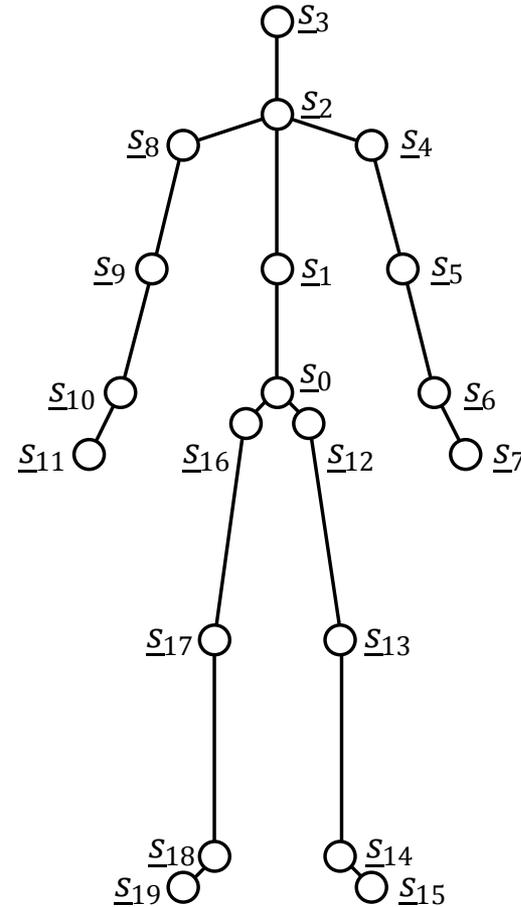
Kinect Sensor liefert mit 30 fps [Mic12]:

- RGB-Video (bis 1280x960)
- Tiefendaten (bis 640x480), Objekt - Abstand in Millimetern (13-bit)

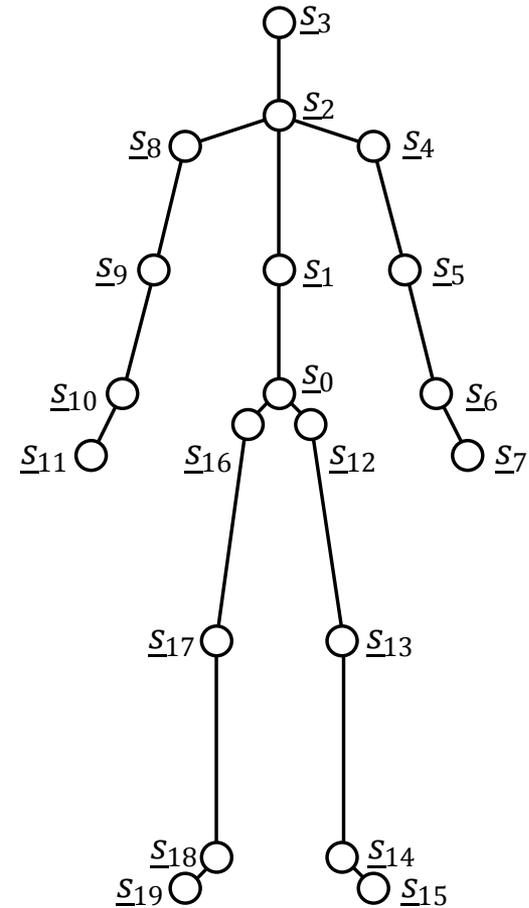
Kinect Sensor liefert mit 30 fps [Mic12]:

- RGB-Video (bis 1280x960)
- Tiefendaten (bis 640x480), Objekt - Abstand in Millimetern (13-bit)
- Skelettdaten:

20 markante Körperpunkte im sog. Skeleton Space, werden automatisch nachverfolgt

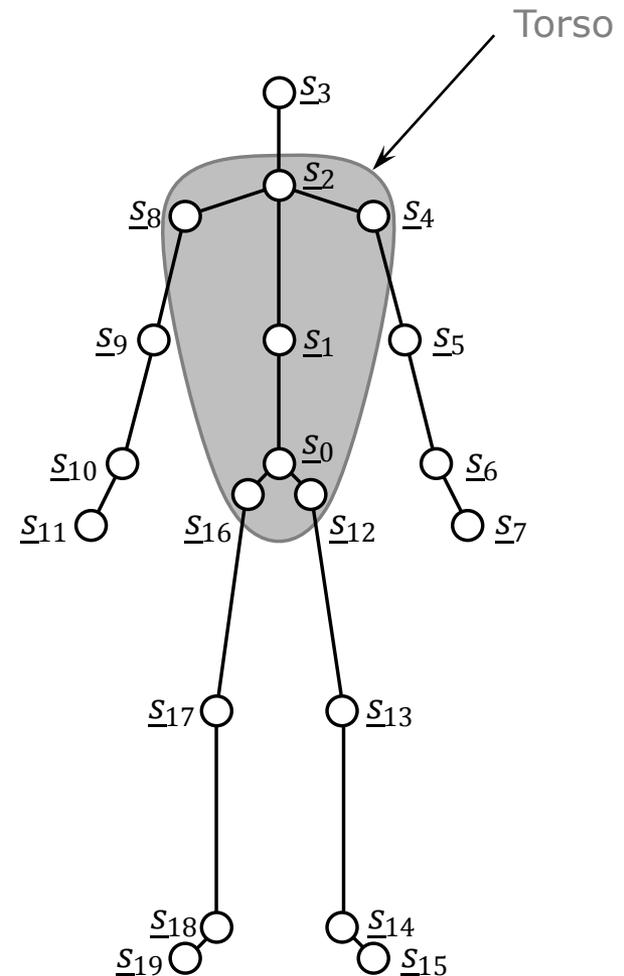


- interne Glättungsfunktionen für Skelettdaten
- dennoch: Trackingprobleme bei Überdeckung von Körperteilen

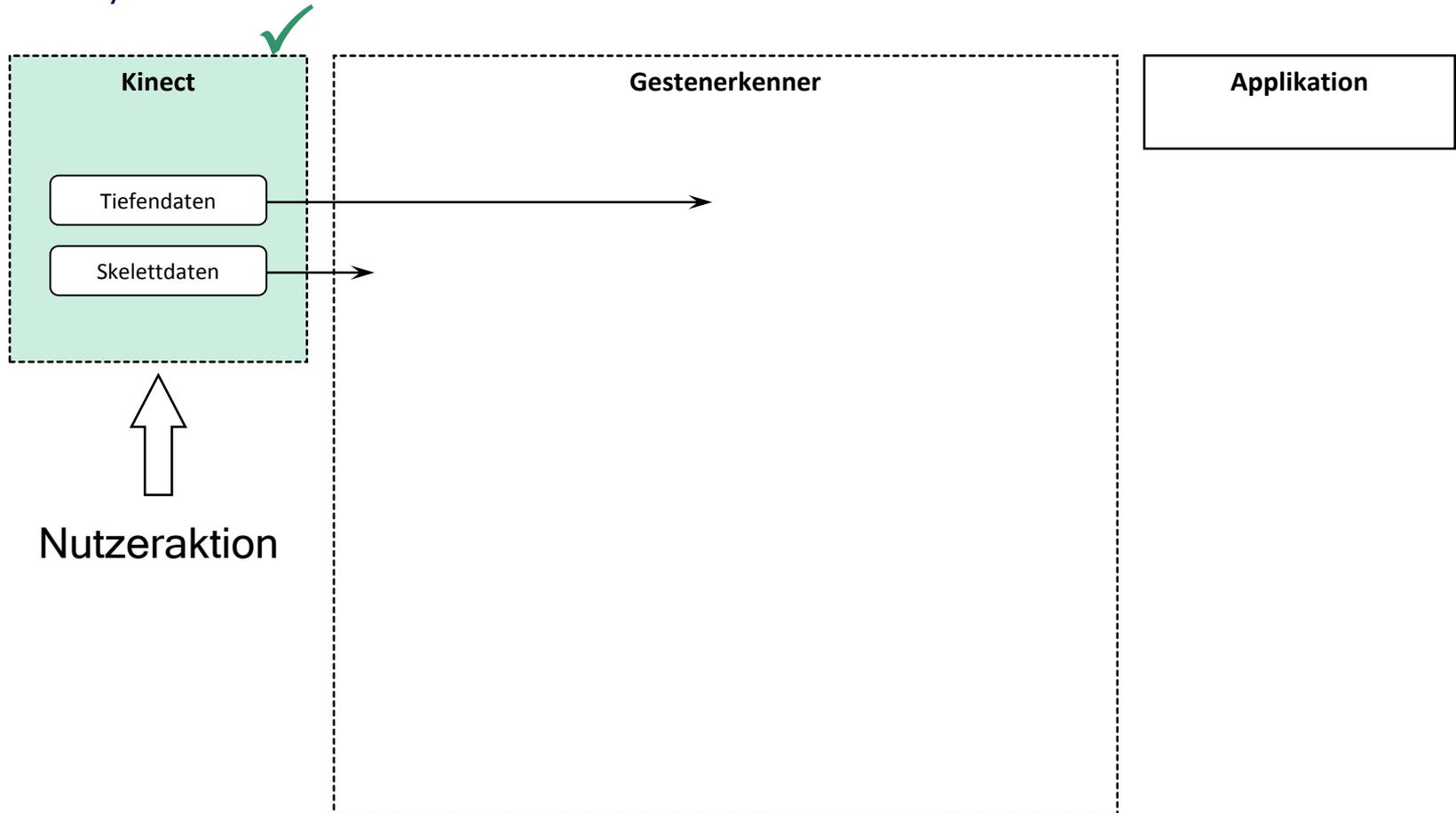


- interne Glättungsfunktionen für Skelettdaten
- dennoch: Trackingprobleme bei Überdeckung von Körperteilen

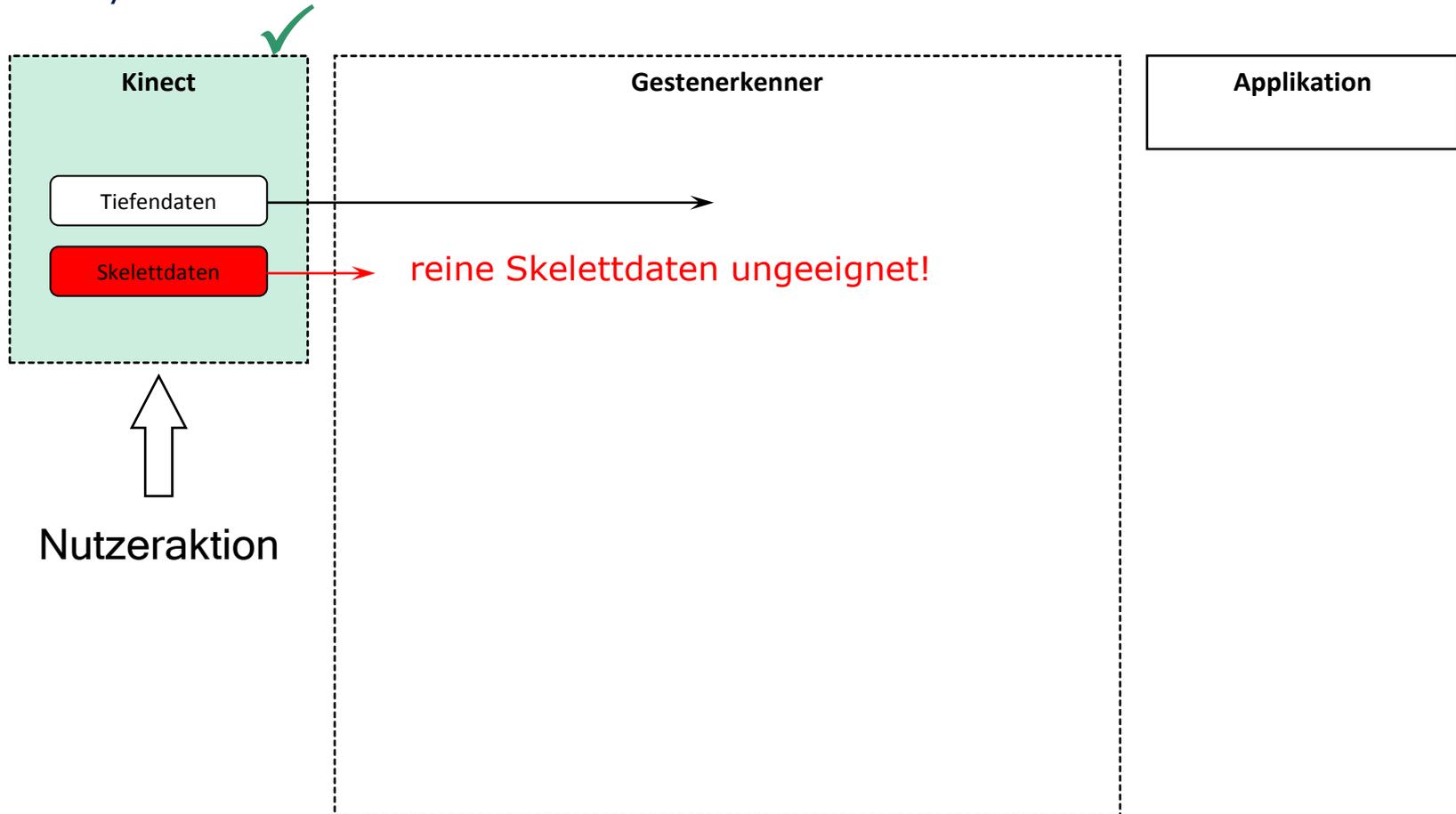
→ Steigerung der Robustheit durch Neuberechnung der Torsopunkte



Gesamtsystem



Gesamtsystem



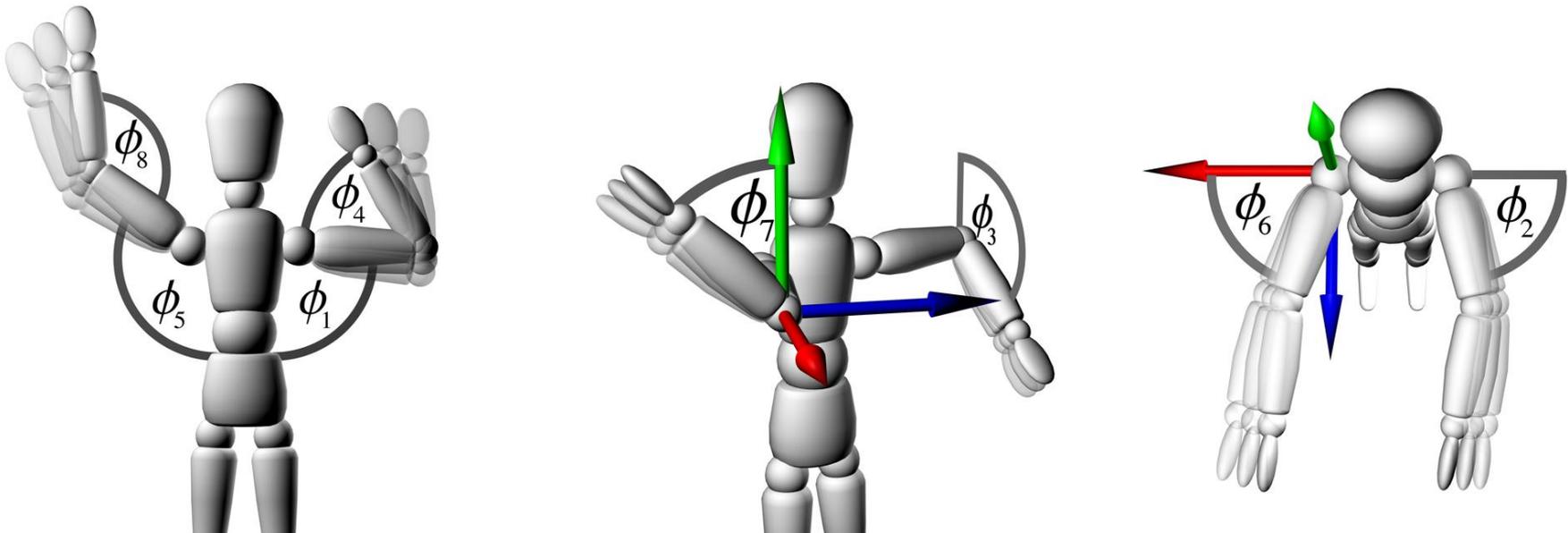
gesucht ist Beschreibungsform invariant gegenüber:

- Lage und Orientierung der Kinect Kamera
- Körpermaßen der Nutzer

gesucht ist Beschreibungsform invariant gegenüber:

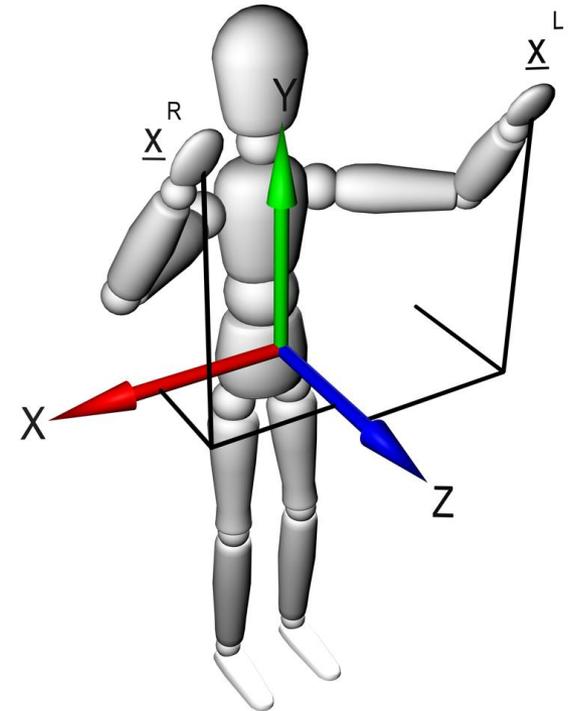
- Lage und Orientierung der Kinect Kamera
- Körpermaßen der Nutzer

→ Winkelrepräsentationen sind geeignet und werden häufig verwendet
(vgl. [YPL07, RKH11, PL06, BP+04, OFH08])



zusätzlich:

- Handpositionen in lokalem Koordinatensystem
- werden robust aus Hauptkomponenten-Analyse der Torsopunkte ermittelt (vgl. [RKH11])



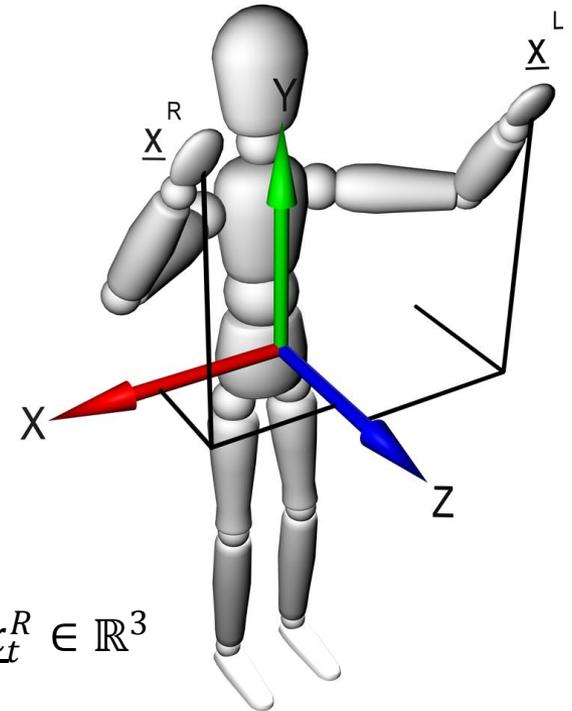
zusätzlich:

- Handpositionen in lokalem Koordinatensystem
- werden robust aus Hauptkomponenten-Analyse der Torsopunkte ermittelt (vgl. [RKH11])

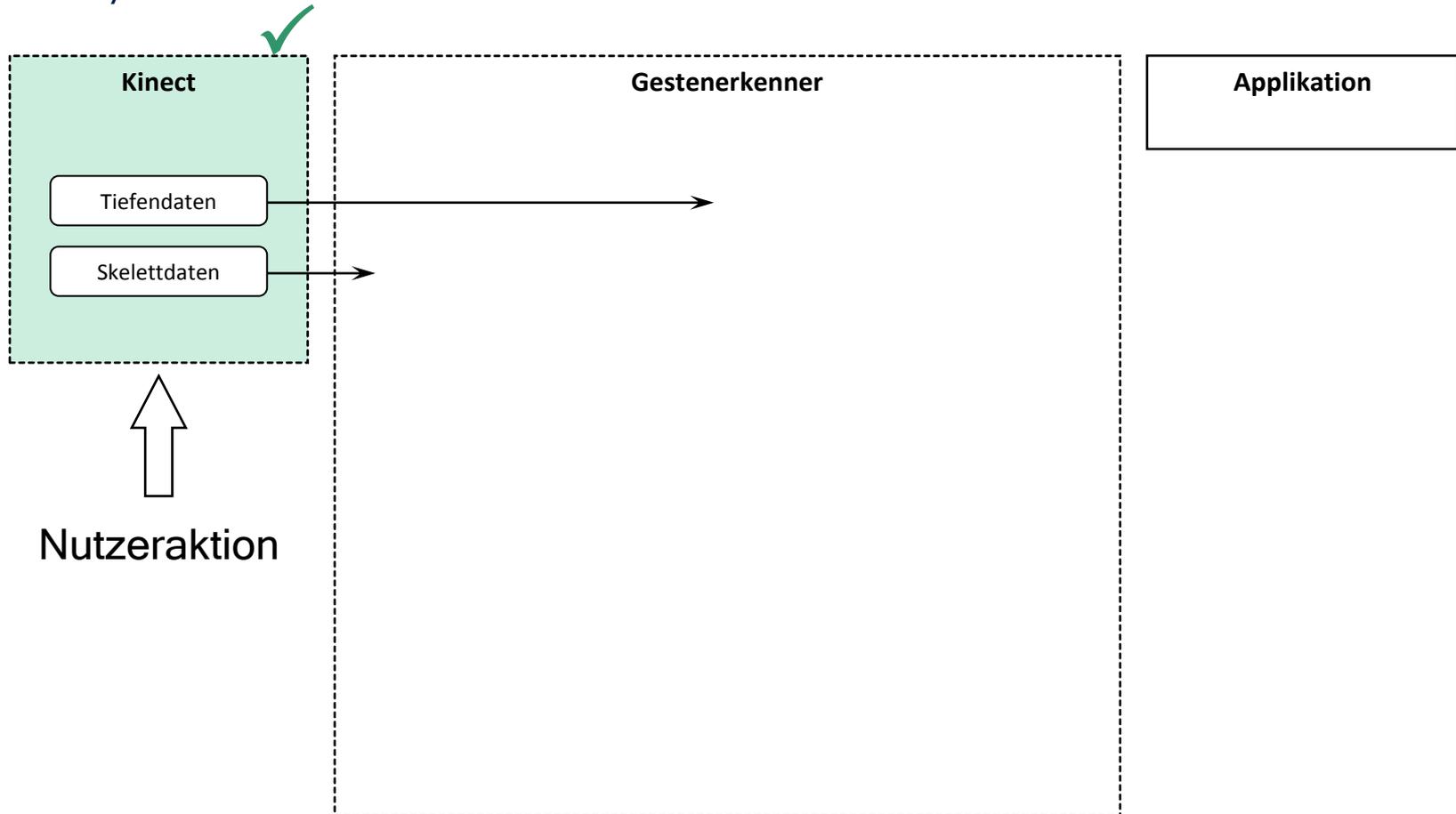
- Pose zum Zeitpunkt t

$$P_t = \{\phi_{1,t}, \phi_{2,t}, \dots, \phi_{8,t}, \underline{x}_t^L, \underline{x}_t^R\} \quad \phi_{i,t} \in \mathbb{R}, i \in [1,8], \underline{x}_t^L, \underline{x}_t^R \in \mathbb{R}^3$$

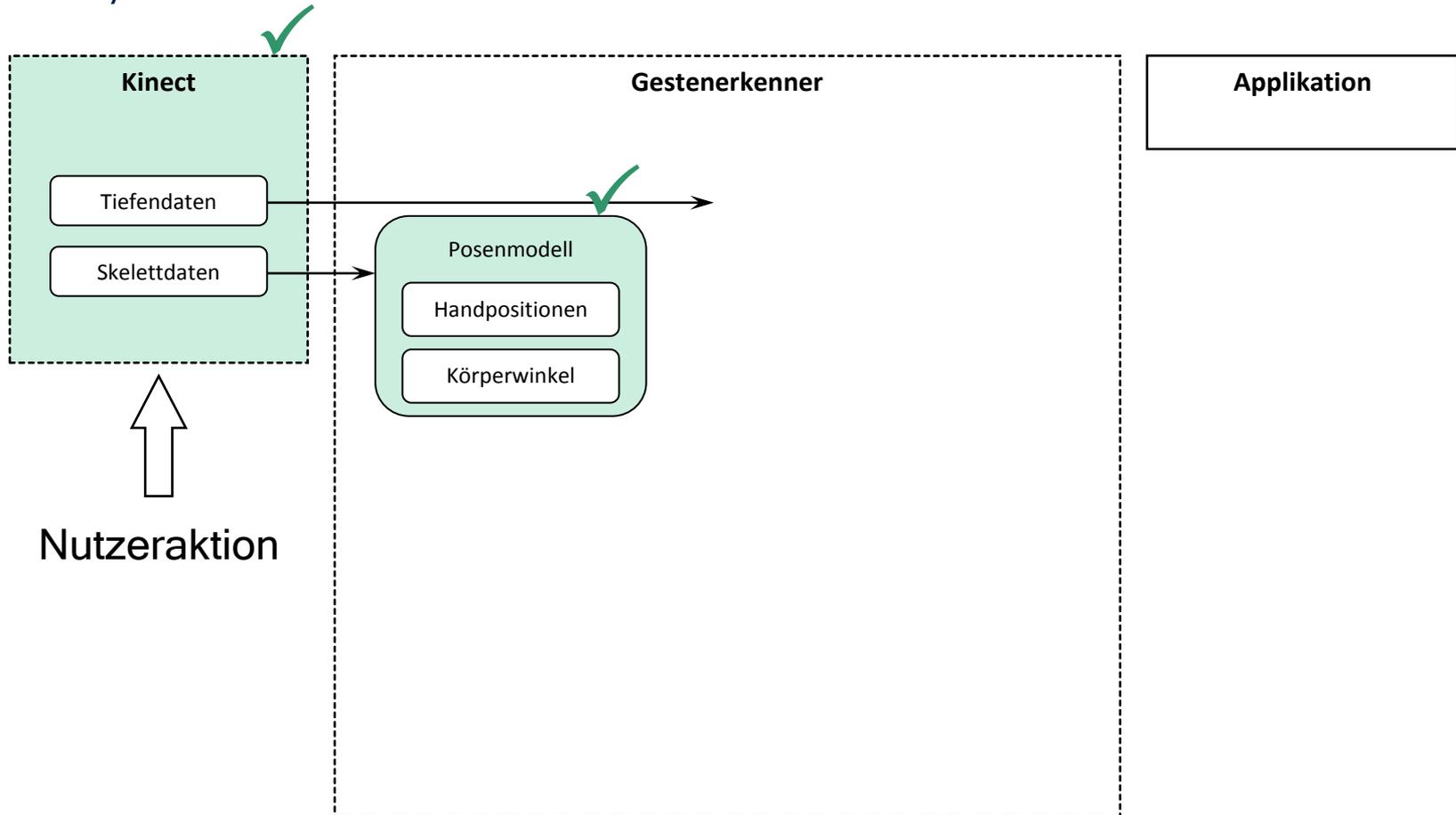
- Redundanzreduktion: $60 \rightarrow 14$



Gesamtsystem



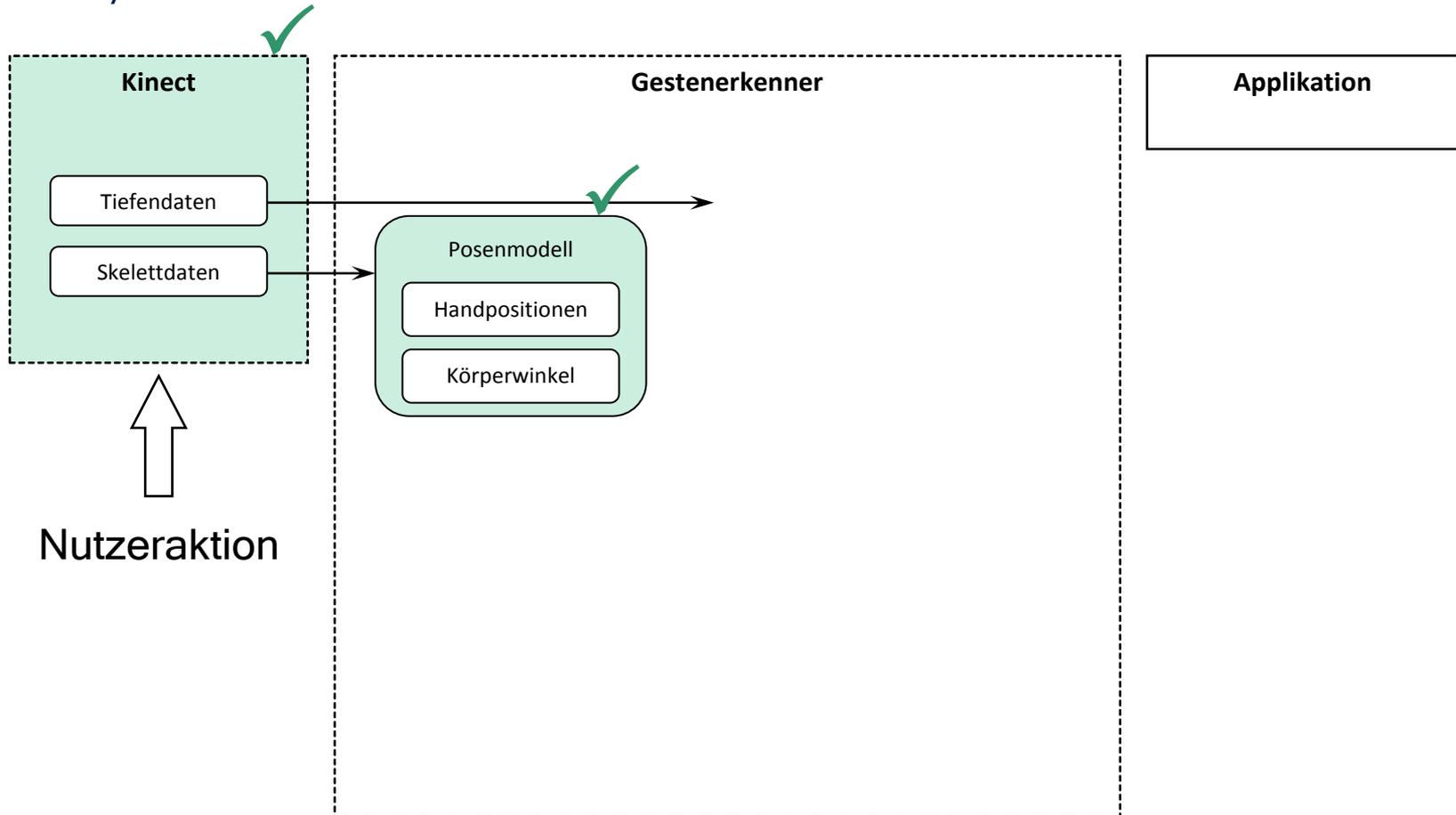
Gesamtsystem



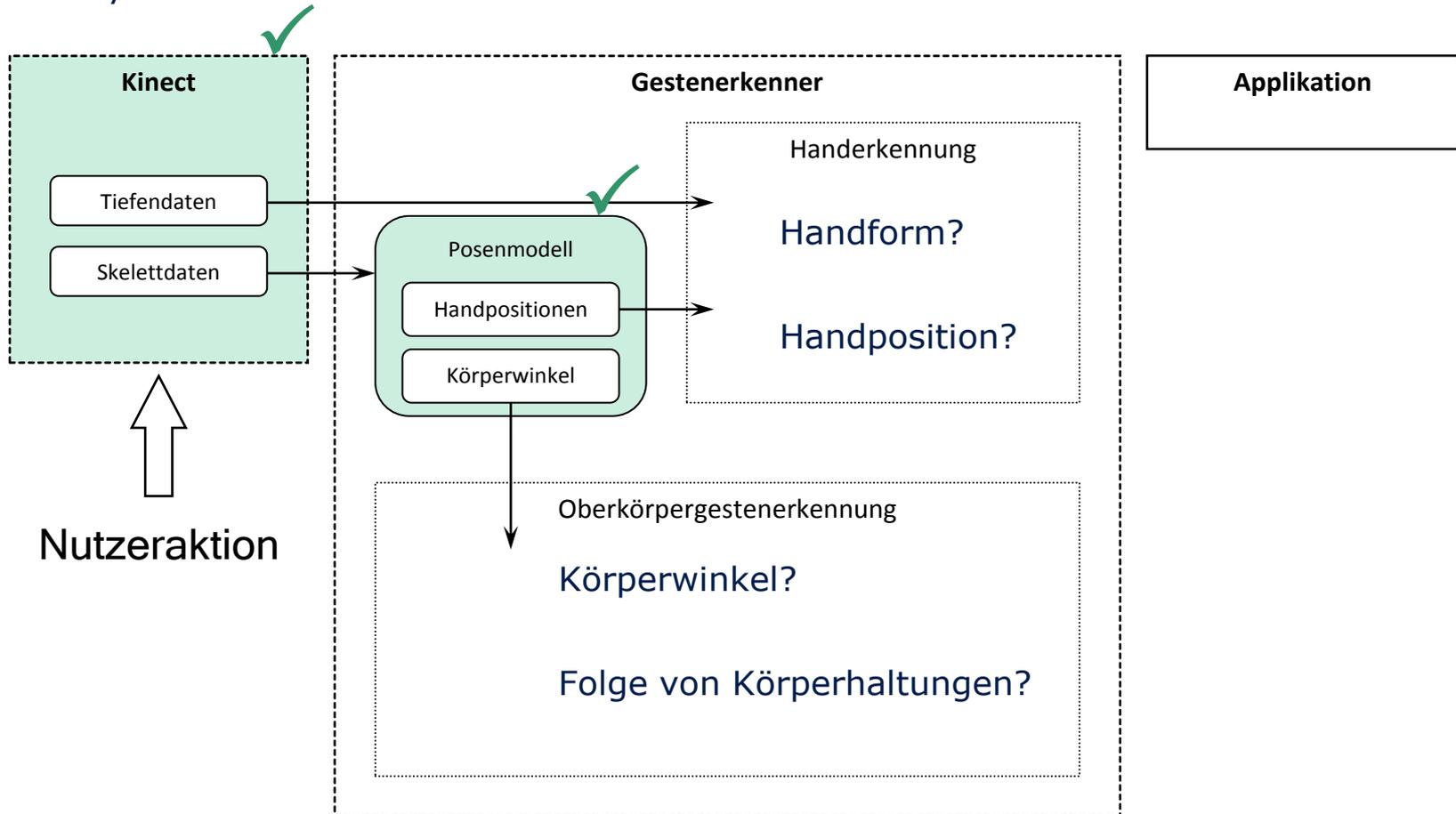
Gesamtsystem



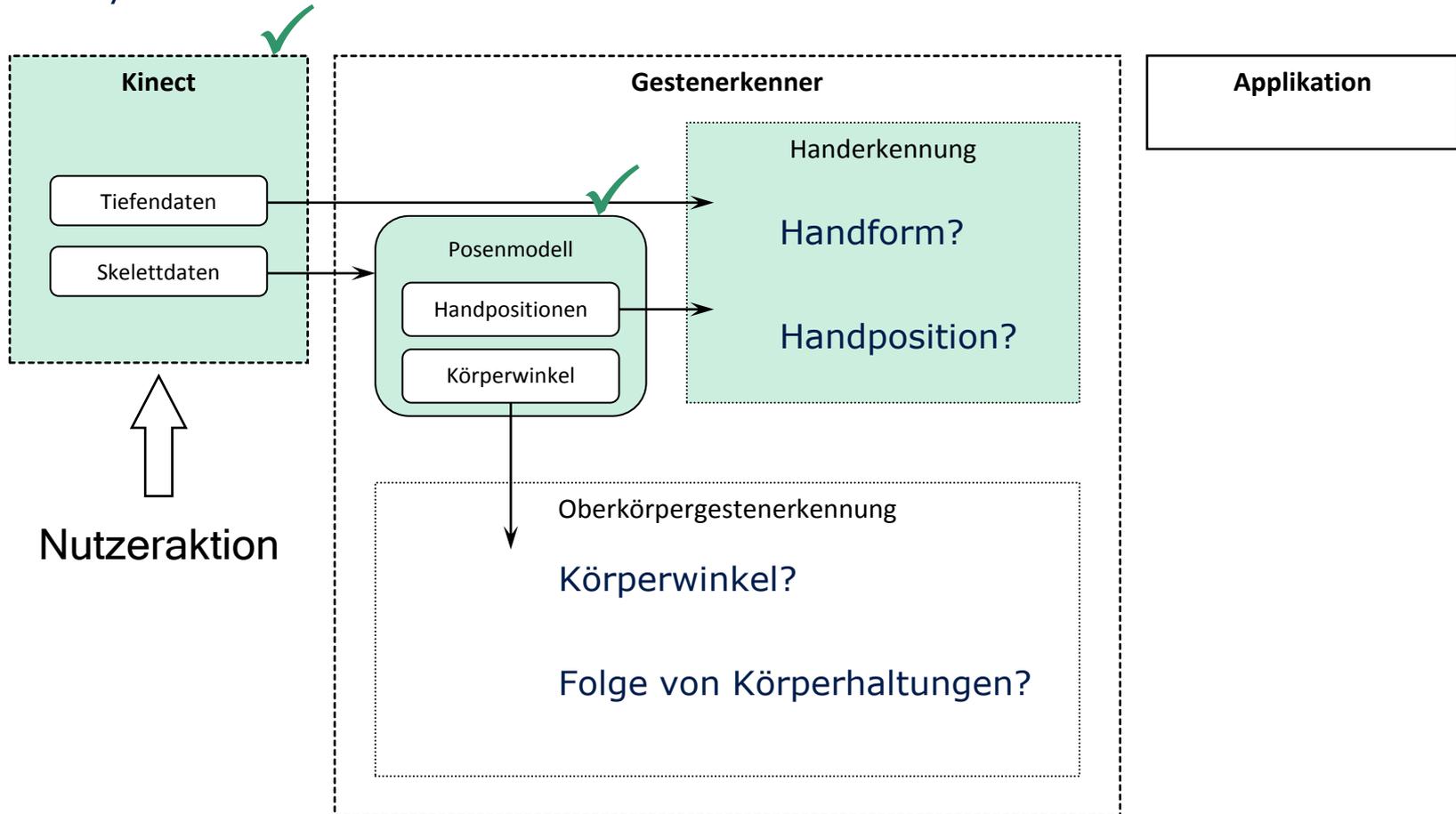
Gesamtsystem



Gesamtsystem



Gesamtsystem



deiktische Handgeste: Handposition → virtuelle Zeigerposition

1. Aufgabe:

- ruhige Zeigerführung → temporale Glättung der Handposition

2. Aufgabe:

- Genaues Positionieren und große Bildschirmbereiche abdecken?

→ nicht-lineare Übersetzung von Hand- in Zeigerposition

semaphorische Handgesten:

- Klassifizierung der Handform:
 - Handfläche als Punktmengen [TLB12]
 - Fitten von Spline-Kurven [RVCB03]
 - Nachbar-Pixelsuche [Li12]
 - Radiale Histogramme [Tan11,RYZ11]

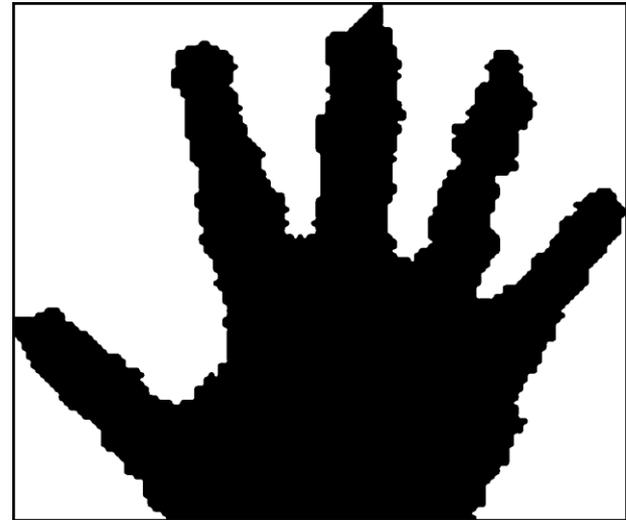
semaphorische Handgesten:

- Klassifizierung der Handform:
 - Handfläche als Punktmengen [TLB12]
 - Fitten von Spline-Kurven [RVCB03]
 - Nachbar-Pixelsuche [Li12]
 - Radiale Histogramme [Tan11,RYZ11]
- hier: Auswertung der Konturlinie in *Polardarstellung* ,
zwei einfache Handstellungen: "offen" u. "geschlossen"

Kinect liefert lediglich 2 Skelettpunkte für Hand
→ bildbasiertes Verfahren über Tiefendaten

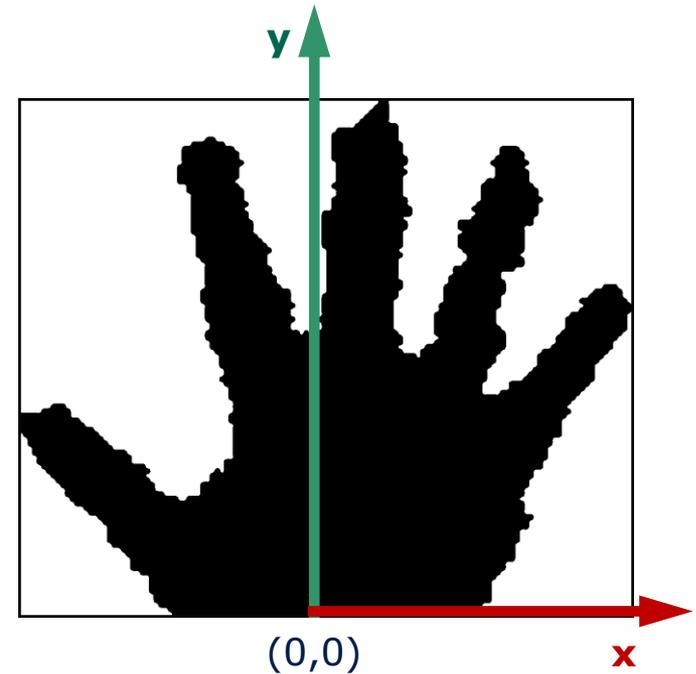
Vorgehen (I)

- Finden der Handwurzel im Tiefenbild
(über Kinect API)



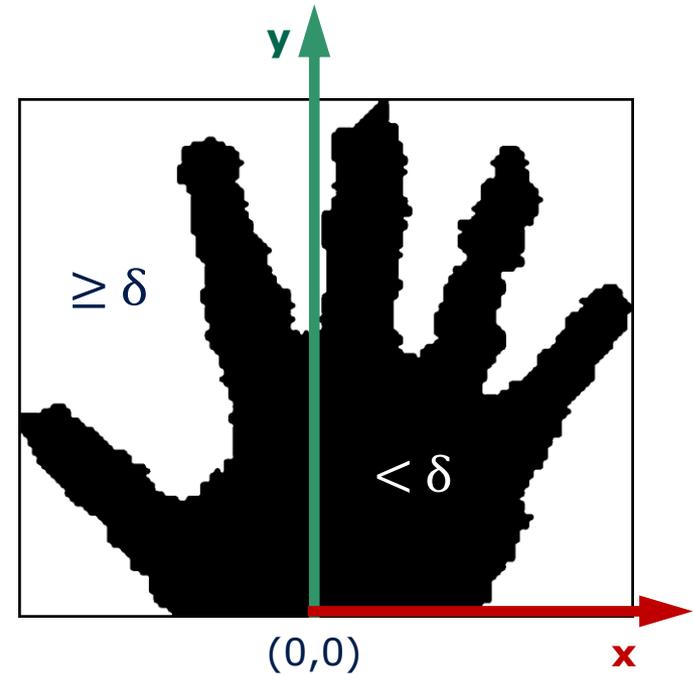
Vorgehen (I)

- Finden der Handwurzel im Tiefenbild (über Kinect API)
- Definition eines lokalen Koordinatensystems



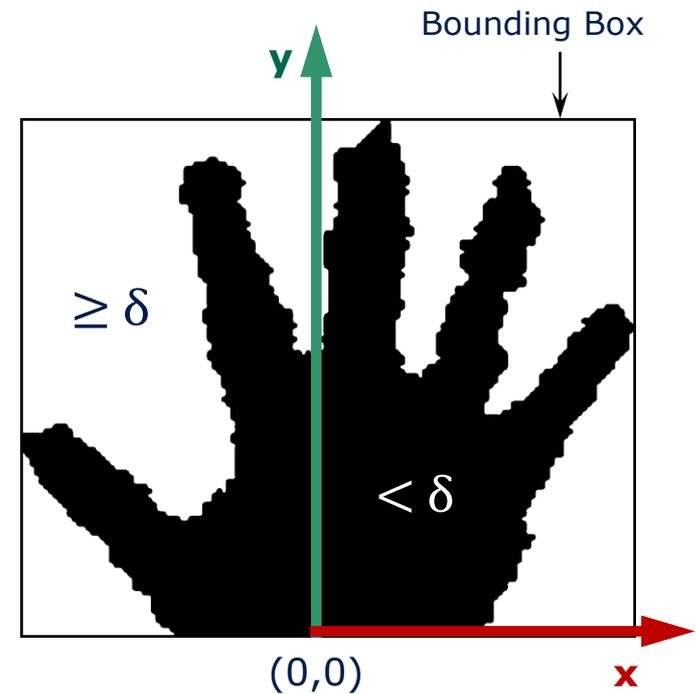
Vorgehen (I)

- Finden der Handwurzel im Tiefenbild (über Kinect API)
- Definition eines lokalen Koordinatensystems
- „Handpixel“ definiert über Schwellwert



Vorgehen (I)

- Finden der Handwurzel im Tiefenbild (über Kinect API)
- Definition eines lokalen Koordinatensystems
- „Handpixel“ definiert über Schwellwert



- Anforderungen: Lokalität und Kontinuität

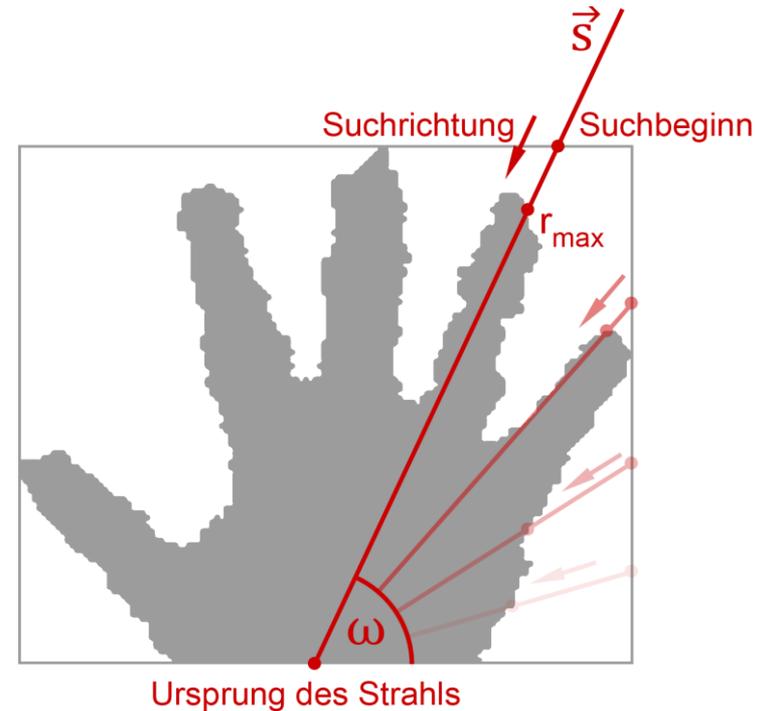
→ Bounding-Box-Verfahren (achsenparallele Suchstrahlen, beginnend im Ursprung)

Vorgehen (II)

- Handkontur über radiale Suchstrahlen

$$\vec{s}_i = r \cdot \begin{pmatrix} \cos \omega_i \\ \sin \omega_i \end{pmatrix}$$

- Strahlen werden effizient von außen nach innen verfolgt (Kontinuität der Handfläche)



Vorgehen (II)

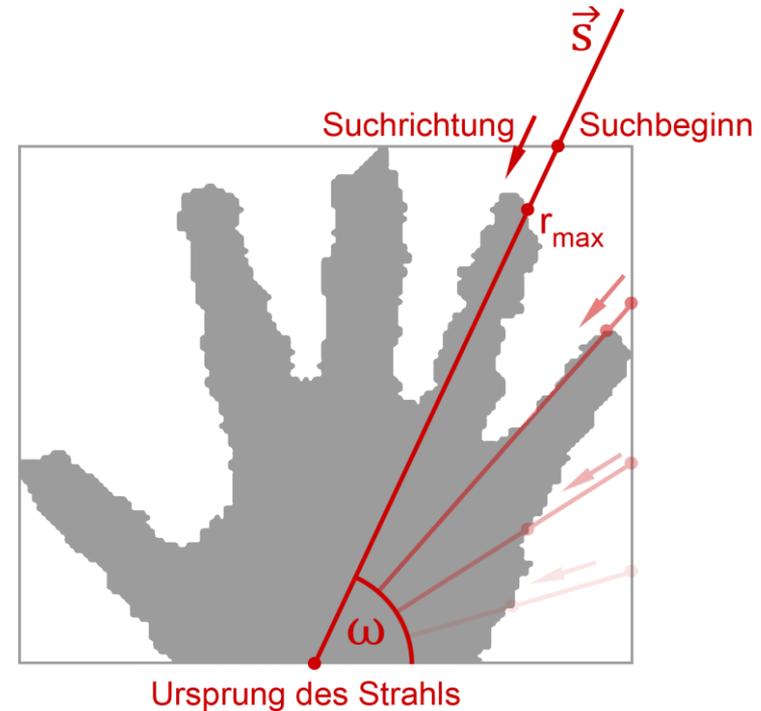
- Handkontur über radiale Suchstrahlen

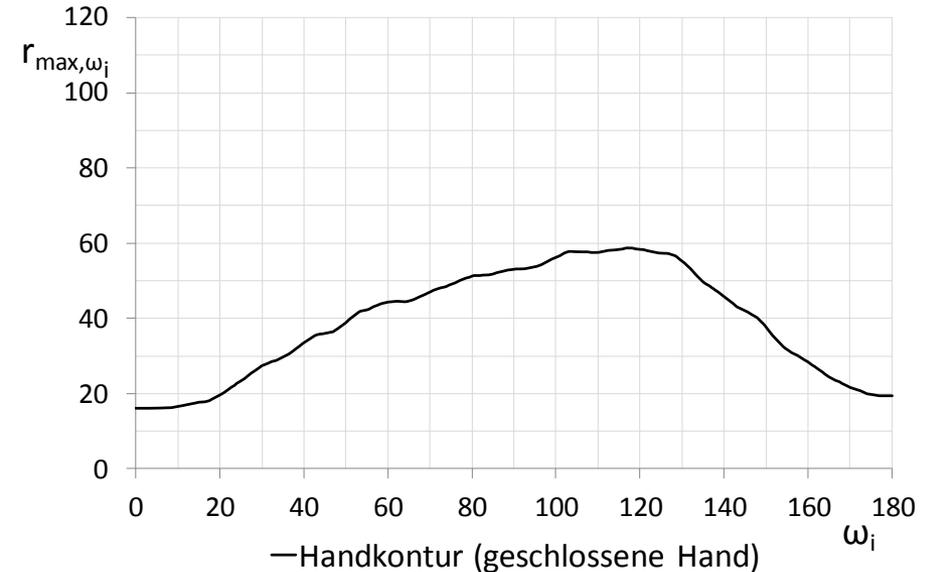
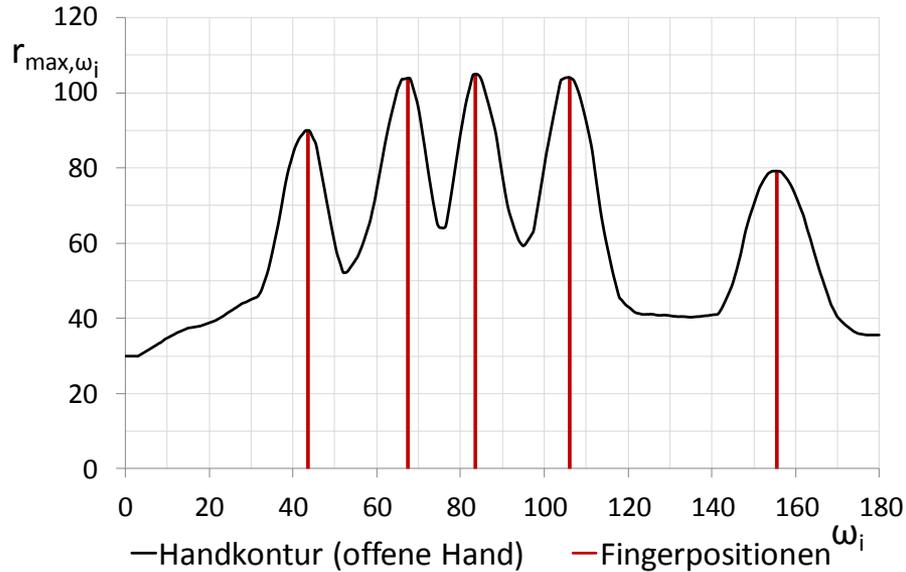
$$\vec{s}_i = r \cdot \begin{pmatrix} \cos \omega_i \\ \sin \omega_i \end{pmatrix}$$

- Strahlen werden effizient von außen nach innen verfolgt (Kontinuität der Handfläche)

- Konturlinie $\vec{k} := (r_{\max,i})$

→ Glätten und Auszählen der lokalen Maxima (=Fingeranzahl)

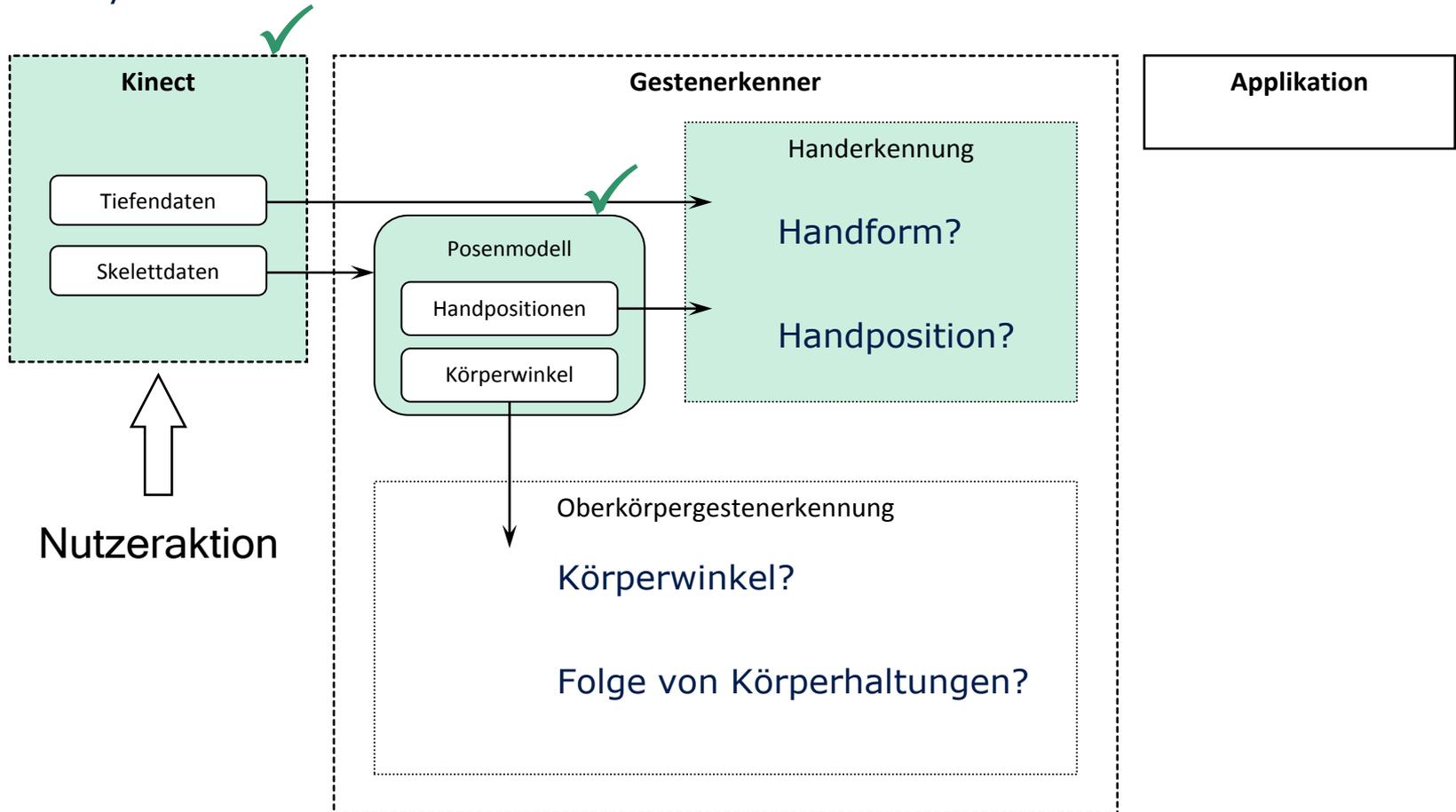




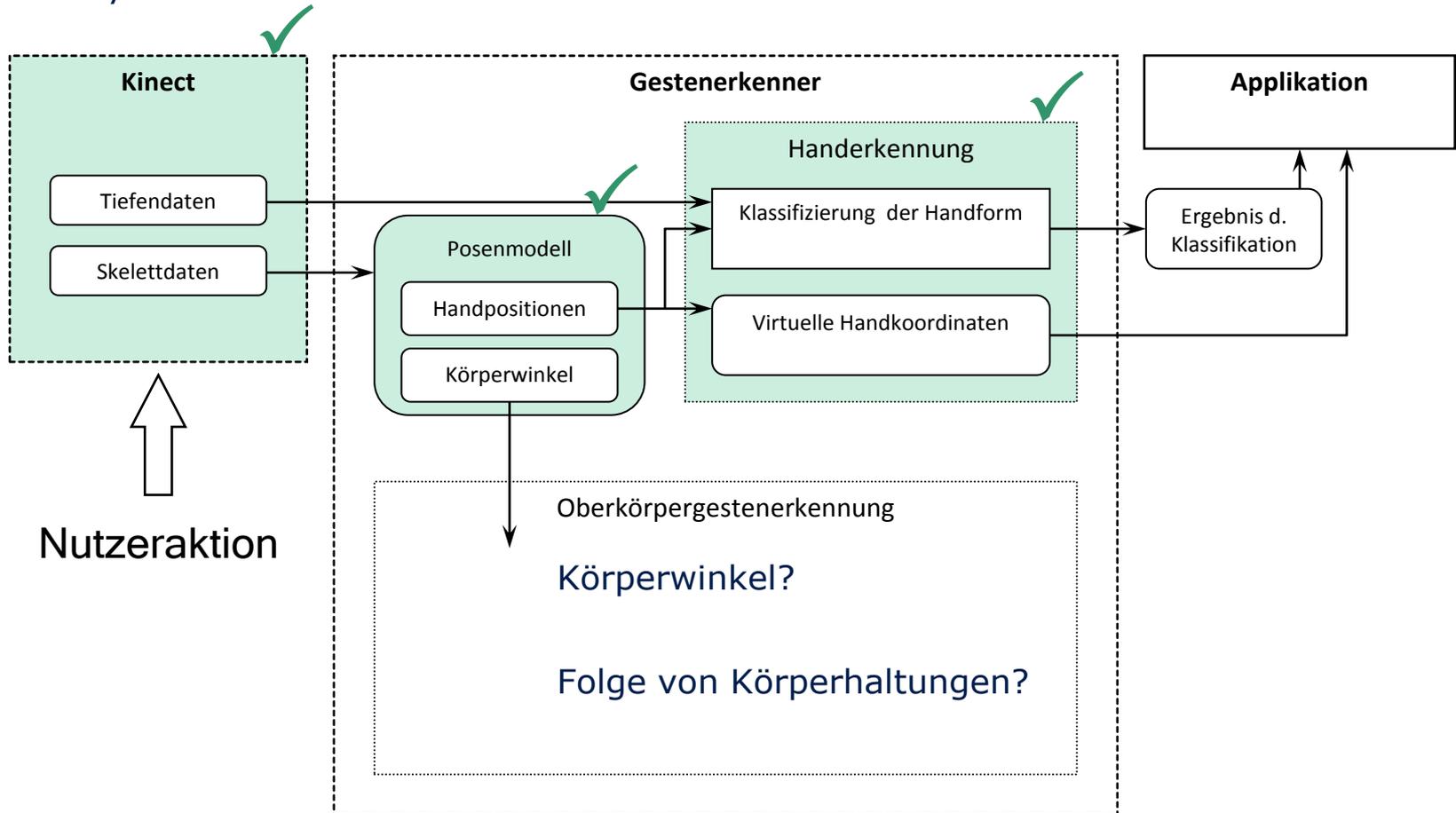
- praxisgerechte Klassifikation:

- Anzahl der lokalen Maxima ≤ 1 → „geschlossene Hand“
- Anzahl der lokalen Maxima > 1 → „offene Hand“

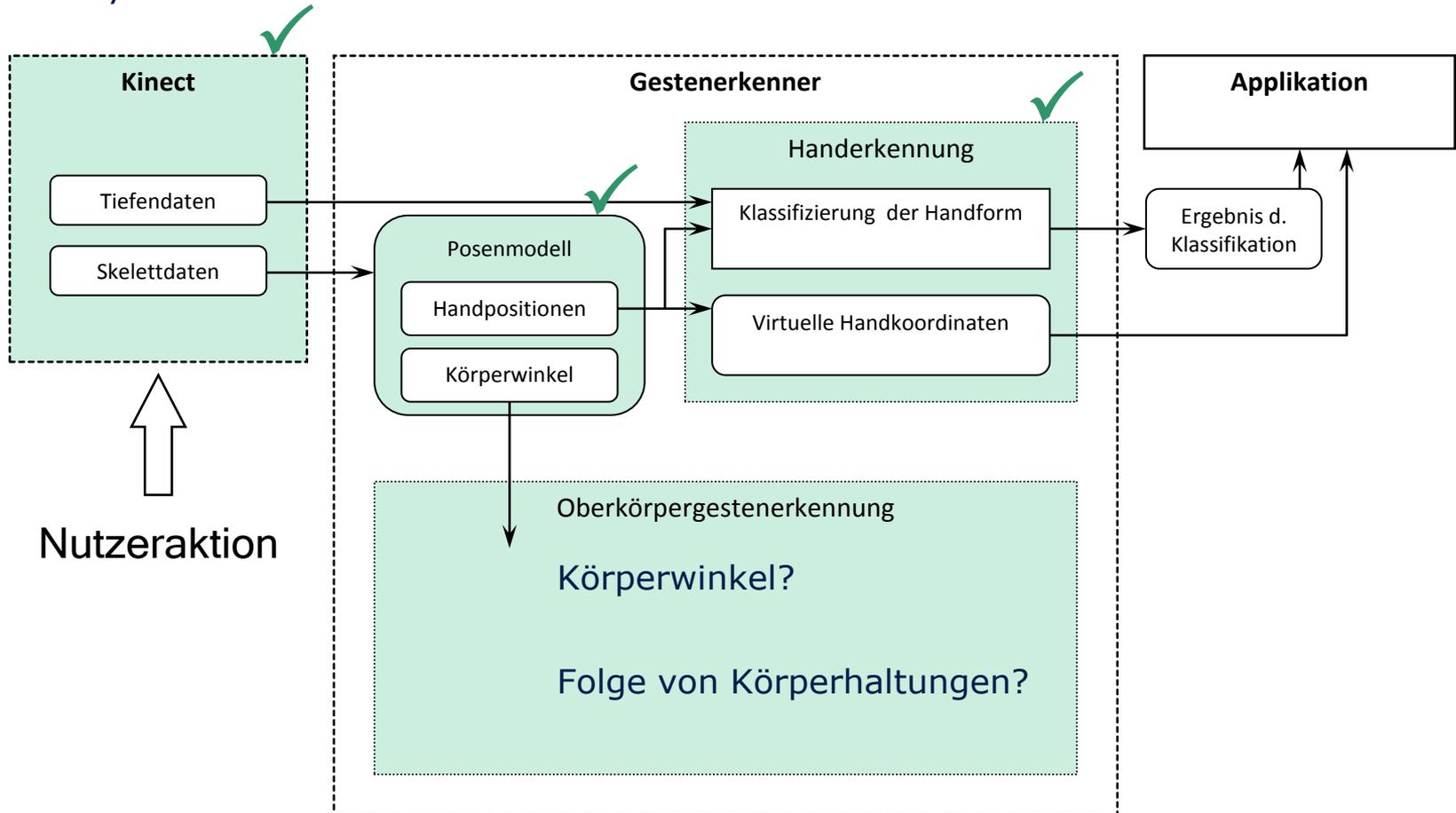
Gesamtsystem



Gesamtsystem



Gesamtsystem



Vorverarbeitung:

- Oberkörpergesten (OKG) werden nur über Winkel modelliert, Extraktion aus der Pose

$$\vec{\phi}_t = \varphi(P_t) = (\phi_{1,t}, \phi_{2,t}, \dots, \phi_{8,t})$$

→ Merkmalsvektor (MV), wird temporal geglättet

deiktische Oberkörpergeste:

Weiterleiten des aktuellen MV an die Applikation

semaphorische OKG: Folge von Körperhaltungen

zeitreihenbasierter Modellansatz:

- Geste = Folge von Merkmalsvektoren → Merkmalsvektorsequenz (MVS)

$$\Phi^{(i,j)} := (\vec{\phi}_i, \vec{\phi}_{i+1}, \dots, \vec{\phi}_{j-1}, \vec{\phi}_j)$$

semaphorische OKG: Folge von Körperhaltungen

zeitreihenbasierter Modellansatz:

- Geste = Folge von Merkmalsvektoren → Merkmalsvektorsequenz (MVS)

$$\Phi^{(i,j)} := (\vec{\phi}_i, \vec{\phi}_{i+1}, \dots, \vec{\phi}_{j-1}, \vec{\phi}_j)$$

2 Kern-Aufgaben des Erkennens:

- Auffinden
- Klassifizieren

Auffinden potenzieller OKG:

- automatisches Segmentieren der Nutzeraktionen aus dem Interaktionsfluss heraus

Warum?

→ Nutzer soll nicht manuell anzeigen

Auffinden potenzieller OKG:

- automatisches Segmentieren der Nutzeraktionen aus dem Interaktionsfluss heraus

Warum?

→ Nutzer soll nicht manuell anzeigen

- Absuchen aller möglichen vergangenen MVS, zu rechenintensiv

→ Markieren von Zeitpunkten potenziellen Gesten-Beginns/-Endes, dann Absuchen

Auffinden potenzieller OKG:

- über Gradienten und deren Änderung (vgl. [KLJ04]) → hier: Winkelgeschwindigkeiten
- Beobachtung:
 - Ruhe → niedriger Geschwindigkeitsbetrag
 - Bewegung → hoher Geschwindigkeitsbetrag

Auffinden potenzieller OKG:

- über Gradienten und deren Änderung (vgl. [KLJ04]) → hier: Winkelgeschwindigkeiten
- Beobachtung:
 - Ruhe → niedriger Geschwindigkeitsbetrag
 - Bewegung → hoher Geschwindigkeitsbetrag

Vorgehen (I)

- Markieren bei Übergang von einer in die andere Phase
- Spezialfälle: Verweilen in einer Haltung oder andauerndes Weiterbewegen
→ Warten bis bestimmte Zahl von Frames abgelaufen, dann markieren

Auffinden potenzieller OKG:

Vorgehen (II)

- Suchfenster **W** konstanter Größe über die markierten Zeitindizes
- Bilden von temporären MVS:

$$\{\Phi_{\text{temp}}^{(i,j)} \mid i, j \in W, i < j\}$$

Auffinden potenzieller OKG:

Vorgehen (II)

- Suchfenster **W** konstanter Größe über die markierten Zeitindizes
- Bilden von temporären MVS:

$$\{\Phi_{\text{temp}}^{(i,j)} \mid i, j \in W, i < j\}$$

- alle temporären MVS klassifizieren
- Weiterrücken von **W** je nach Klassifikationsergebnis

Auffinden potenzieller OKG:

Vorgehen (II)

- Suchfenster **W** konstanter Größe über die markierten Zeitindizes
- Bilden von temporären MVS:

$$\{\Phi_{\text{temp}}^{(i,j)} \mid i, j \in W, i < j\}$$

- alle temporären MVS **klassifizieren** 
- Weiterrücken von **W** je nach Klassifikationsergebnis

Klassifikation von MVS

Grundlage der Erkennung?

→ Trainingsdaten! Aufzeichnen von MVS

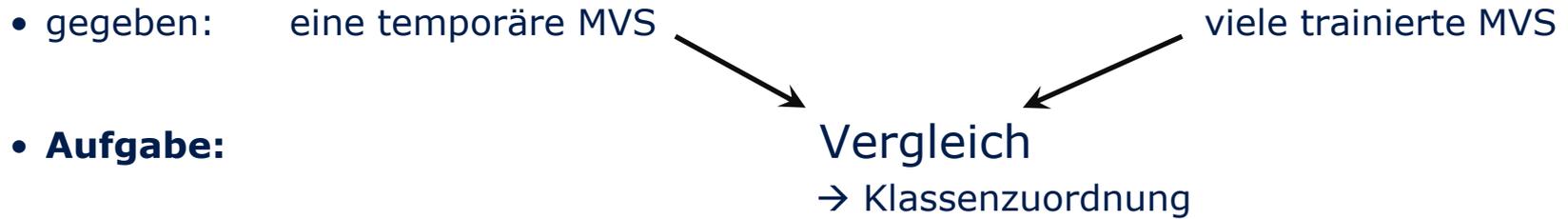
- Bilden von Klassen, d.h. mehrere MVS-Bsp. der gleichen Bewegung werden zusammengefasst

Klassifikation von MVS

- gegeben: eine temporäre MVS

viele trainierte MVS

Klassifikation von MVS



Klassifikation von MVS

verwendeter Algorithmus: k-Nächste-Nachbarn-Verfahren

- hinreichend schnell, einfach zu implementieren

Klassifikation von MVS

verwendeter Algorithmus: k-Nächste-Nachbarn-Verfahren

- hinreichend schnell, einfach zu implementieren

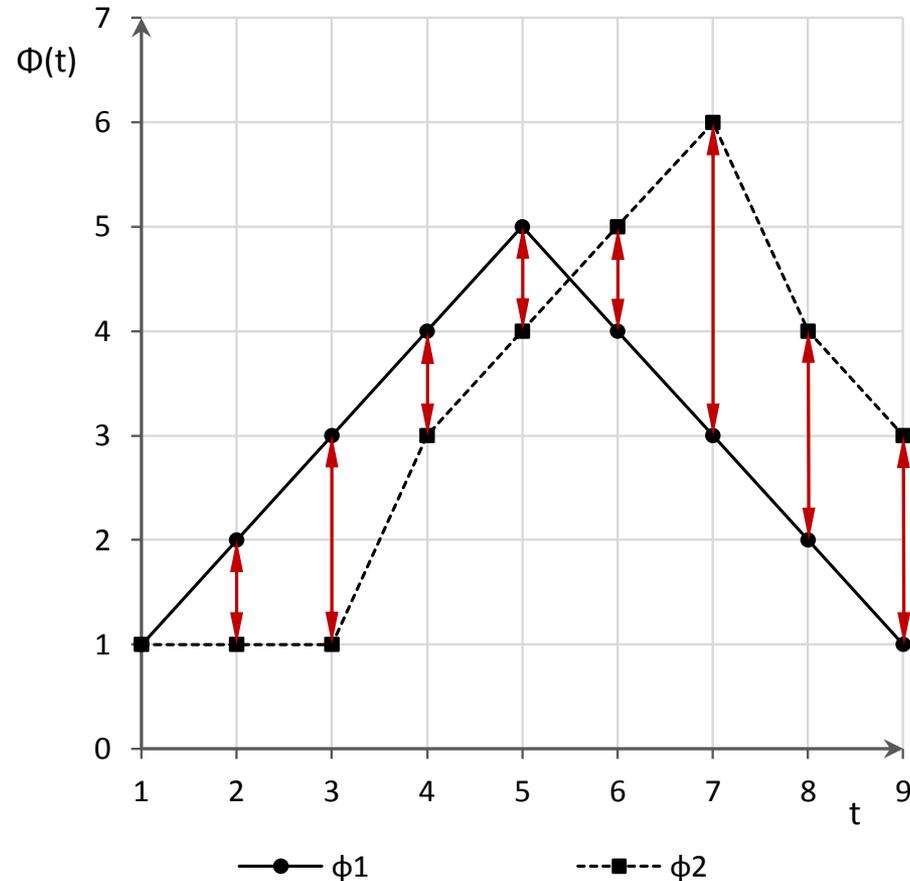
- untersuchte Distanzmaße:
 - Euklidischer Abstand
 - modifizierter euklidischer Abstand (dazu Komplexitätsmaß nach [BWK11])
 - Pearson-Distanz
 - FM-Distanz [OFH08]
 - Dynamic Time Warping (DTW)

Klassifikation von MVS

Schwächen der „klassischen“ Maße:

- MVS müssen gleiche Länge haben (lösbar über Resampling)
- Zuordnung der MV-Korrespondenzen ist festgelegt

Bsp. 1D – Zeitreihen



Klassifikation von MVS

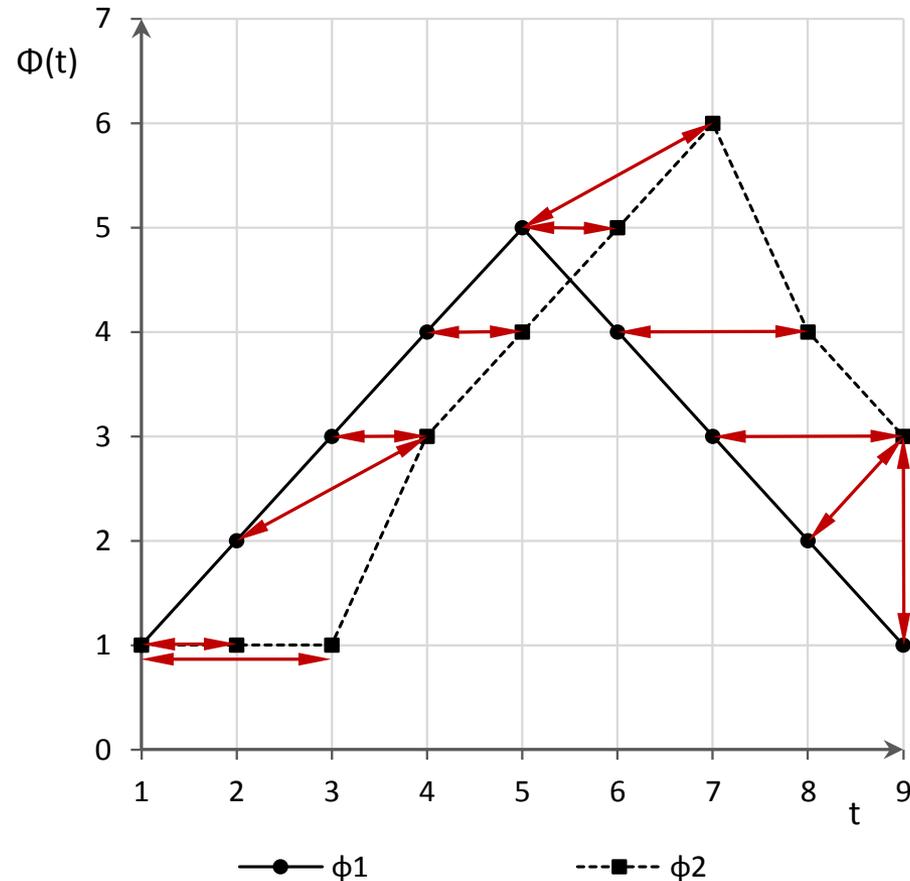
Schwächen der „klassischen“ Maße:

- MVS müssen gleiche Länge haben (lösbar über Resampling)
- Zuordnung der MV-Korrespondenzen ist festgelegt

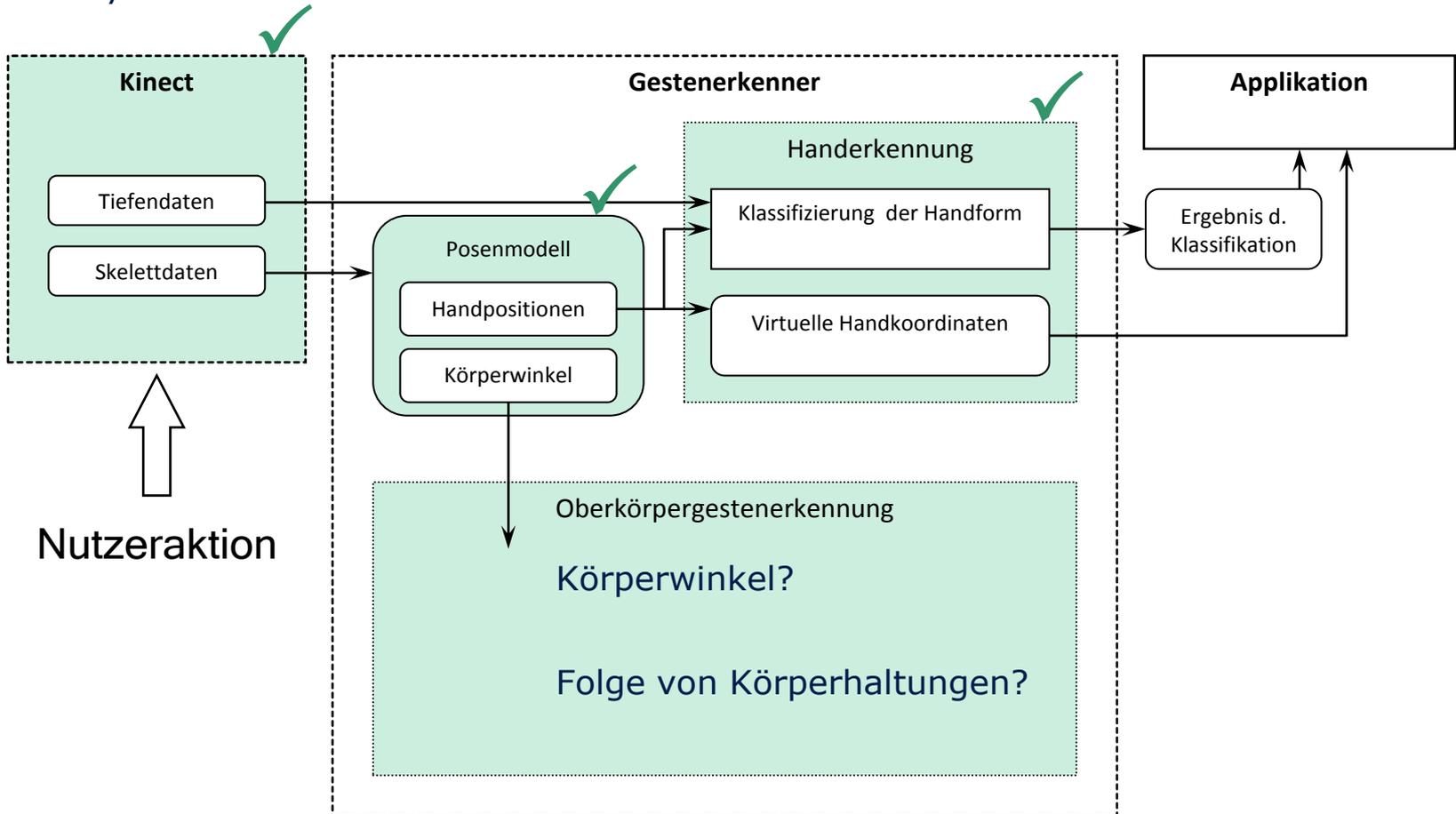
→ Vorteil DTW:

- lokale Anpassung der MV-Korrespondenzen

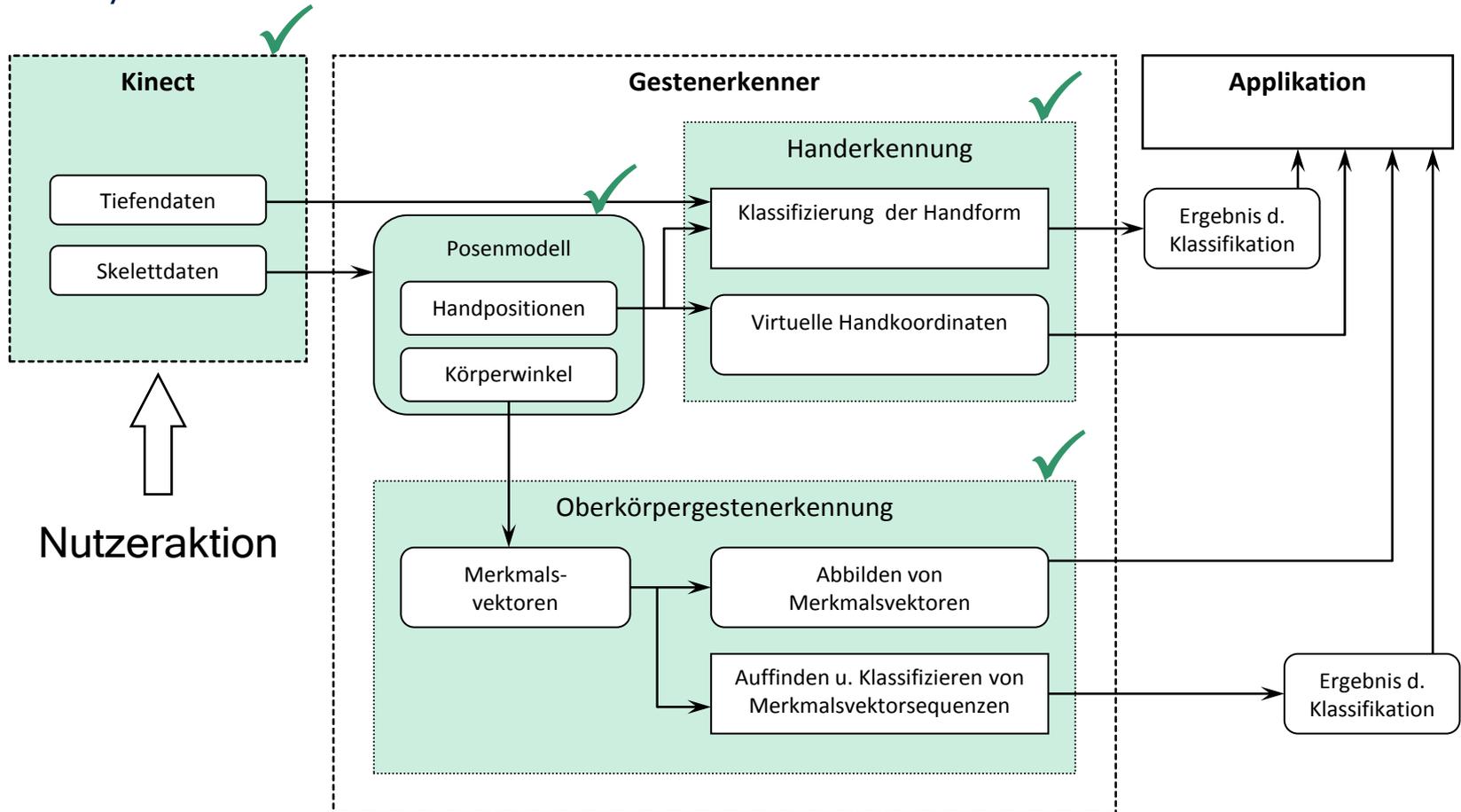
Bsp. 1D – Zeitreihen



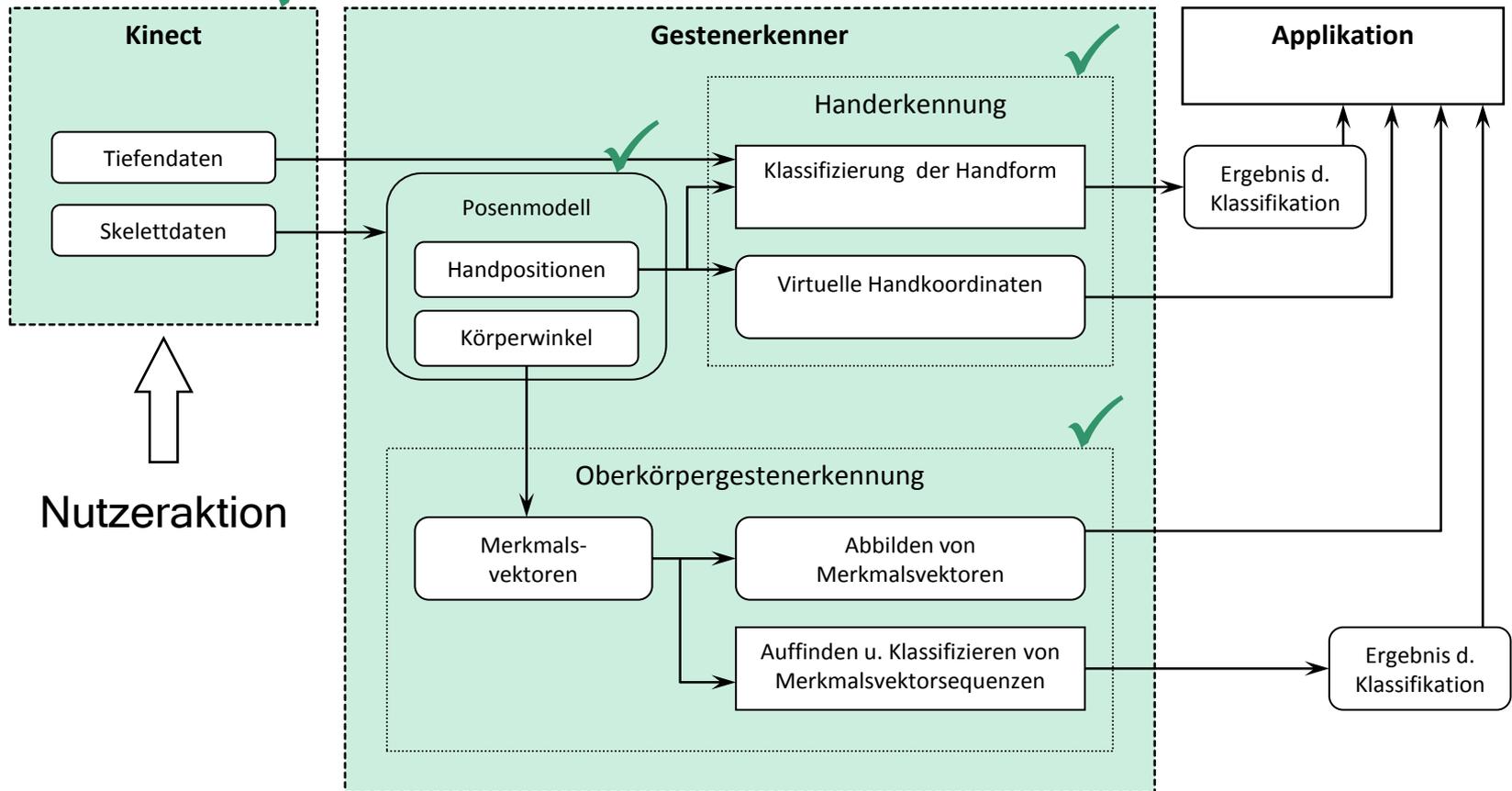
Gesamtsystem



Gesamtsystem



Gesamtsystem



3. Diskussion

Was wurde untersucht?

- Klassifikationsraten
- Geschwindigkeit

Klassifikation der Handform

Testdaten:

- 2500 Trainingsbeispiele, unterschiedliche Abstände zur Kamera, verschiedene Ausführungsgeschwindigkeiten, Hand frontal zur Kamera

Ergebnisse:

- Eingabe der geschlossenen Hand stets zu 100% korrekt klassifiziert
- korrekte Klassifizierung der geöffneten Hand fällt ab bestimmten Abstand stark ab
- Ø Rechenzeit pro Klassifikation: 0,37 ms

Klassifikation der Handform

	Distanzbereich	richtig erkannt	falsch erkannt	Genauigkeit
„geschlossene Hand“	1201 - 1600mm	100%	0%	100%
	800 - 1900mm	100%	0%	96,76%
	>1900mm	100%	0%	52,96%
„offene Hand“	1201 - 1600mm	100%	0%	100%
	800 - 1900mm	96,65%	3,35%	100%
	>1900mm	11,18%	88,82%	100%

Distanzbereich	Korrektklassifikationsrate	Falschklassifikationsrate
1201 - 1600mm	100%	0%
800 - 1900mm	98,33%	1,67%
>1900mm	55,59%	44,41%

$$\text{Genauigkeit (precision)} = \frac{\text{Klasse X richtig erkannt}}{\text{Klasse X erkannt}} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

$$\text{Korrektklassifikationsrate (accuracy)} = \frac{\text{Anz. der korrekt erkannten Gesten}}{\text{Gesamtheit der eingegeben Gesten}} = \frac{\text{true pos.} + \text{true neg.}}{\text{true pos.} + \text{true neg.} + \text{false pos.} + \text{false neg.}}$$

Klassifikation der Handform

	Distanzbereich	richtig erkannt	falsch erkannt	Genauigkeit
„geschlossene Hand“	1201 - 1600mm	100%	0%	100%
	800 - 1900mm	100%	0%	96,76%
	>1900mm	100%	0%	52,96%
„offene Hand“	1201 - 1600mm	100%	0%	100%
	800 - 1900mm	96,65%	3,35%	100%
	>1900mm	11,18%	88,82%	100%

Distanzbereich	Korrektklassifikationsrate	Falschklassifikationsrate
1201 - 1600mm	100%	0%
800 - 1900mm	98,33%	1,67%
>1900mm	55,59%	44,41%

$$\text{Genauigkeit (precision)} = \frac{\text{Klasse X richtig erkannt}}{\text{Klasse X erkannt}} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

$$\text{Korrektklassifikationsrate (accuracy)} = \frac{\text{Anz. der korrekt erkannten Gesten}}{\text{Gesamtheit der eingegeben Gesten}} = \frac{\text{true pos.} + \text{true neg.}}{\text{true pos.} + \text{true neg.} + \text{false pos.} + \text{false neg.}}$$

Klassifikation der MVS

Testdaten:

- 600 Trainings-Bsp., 10 Klassen → 2 Gruppen: 50 Test- u. 10 Referenz-Bsp. pro Klasse

Test 1: isolierte Klassifikation

Distanzmaß	k	R	F	N	KK	ØRZ
Eukl. Abstand	3	497	3	0	99,4%	1,35 ms
Mod. Eukl. Abstand	3	499	1	0	99,8%	1,44 ms
Pearson- Distanz	3	437	0	63	87,4%	3,33 ms
Dynamic Time Warping	3	482	18	0	96,4%	1,22 ms
FM-Distanz	3	498	2	0	99,6%	2,52 ms

k: Anzahl d. Nachbarn, R: richtig erkannt, F: falsch erkannt, N: nicht erkannt,
KK: Korrektklassifikationsrate, ØRZ: durchschnittliche Rechenzeit

Klassifikation der MVS

Testdaten:

- 600 Trainings-Bsp., 10 Klassen → 2 Gruppen: 50 Test- u. 10 Referenz-Bsp. pro Klasse

Test 1: isolierte Klassifikation

Distanzmaß	k	R	F	N	KK	ØRZ
Eukl. Abstand	3	497	3	0	99,4%	1,35 ms
Mod. Eukl. Abstand	3	499	1	0	99,8%	1,44 ms
Pearson- Distanz	3	437	0	63	87,4%	3,33 ms
Dynamic Time Warping	3	482	18	0	96,4%	1,22 ms
FM-Distanz	3	498	2	0	99,6%	2,52 ms

k: Anzahl d. Nachbarn, R: richtig erkannt, F: falsch erkannt, N: nicht erkannt,
KK: Korrektklassifikationsrate, ØRZ: durchschnittliche Rechenzeit

gut und schnell

Klassifikation der MVS

- gute Ergebnisse für alle Distanzmaße, Eignung im laufenden Betrieb?

Test 2: Klassifikation zur Laufzeit

- Ausführung der 10 Gesten in unterschiedl. Reihenfolge, insgesamt 600 Ausführungen
- On-The-Fly-Erkennung durch das parallele System von Finden und Klassifizieren
- nur DTW-Verfahren bringt hier gute Ergebnisse
- bei den anderen: häufig fälschliche Klassifikationen von Nicht-Gesten (false positives)

Test 2: Klassifikation zur Laufzeit

- Gesamtergebnis (DTW):
 - Anzahl der Gesten: 600
 - richtig erkannt: 574
 - falsch erkannt: 7
 - nicht erkannt: 19
 - Korrektklassifikationsrate: 95,67%
 - \emptyset Genauigkeit: 98,85%

- Geschwindigkeitsbetrachtung:
 - Festlegung: Gesten $< 3s$, Suchfenstergröße 12
 - \emptyset RZ = 84 ms (pro Fenster) < 267 ms (\emptyset Markierungsintervall),
 - \emptyset Latenz ca. 1-4 Markierungsintervalle (267ms-1067ms), maximal 3,17s

Handerkennung

Handposition

- + Ruhige Zeigerführung, Latenz vertretbar, Details bleiben erhalten
- + nichtlineare Skalierung erlaubt Zielgenauigkeit und gleichzeitig großen Arbeitsbereich
- kein vollständiges Ausgleichen von Tracking-Ungenauigkeiten

Handform

- + gute Erkennung in bestimmten Distanzbereich
- + schnell (ca. 1% von Frame-Dauer)
- kleiner nutzbarer Tiefenbereich durch Mindestabstand d. Nutzers zur Kinect

Erkennung von OKG

- + für überschaubare Anzahl selbstdefinierter Gesten gute Ergebnisse
- + schnell
- systembedingte maximale Verzögerung

Erweiterungspotenzial:

- Verbesserung d. Handformerkennung unter Einbeziehung d. Videobildes
- Erkennung von Fingerstellungen
- mehrere Kinects synchronisieren → größerer effektiver Nutzerbereich
- effektiveres Training durch halbautomatisches Labeling

...

4. Demonstration

Details

Beobachtung: stabilster Punkt ist \underline{s}_1

1. Annahme: Nutzer bleibt an Ort und Stelle

→ Glättung von \underline{s}_1 :

$$\tilde{s}_{1,t} := (1 - \beta) \cdot \underline{s}_{1,t} + \beta \cdot \tilde{s}_{1,t-1}$$

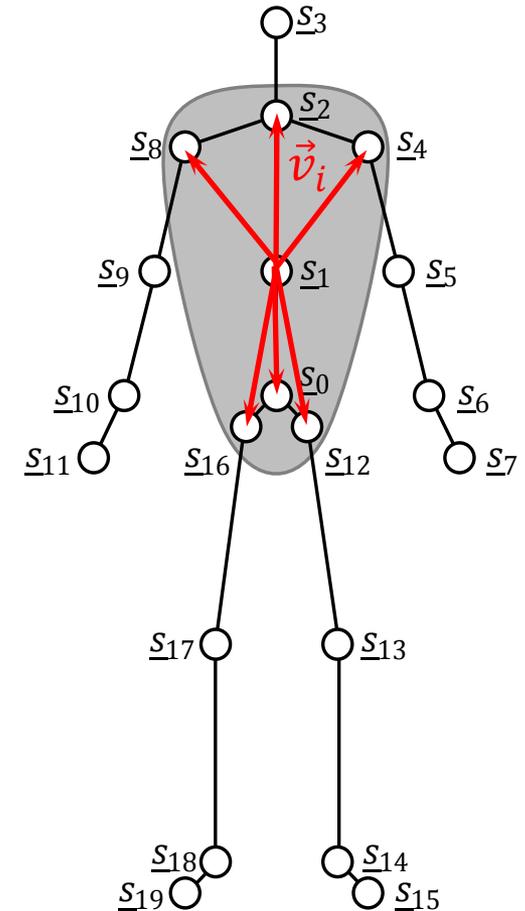
2. Annahme: Torso ist starr

Stützvektoren $\vec{v}_{i,t} := \underline{s}_{i,t} - \tilde{s}_{1,t}$ für $i \in \{0,2,4,8,12,16\}$

werden temporal geglättet: $\tilde{v}_{i,t} := (1 - \alpha) \cdot \vec{v}_{i,t} + \alpha \cdot \tilde{v}_{i,t-1}$

Berechnung rekonstruierter Torsopunkte $\underline{s}'_{i,t}$

$$\underline{s}'_{i,t} := \tilde{s}_{1,t} + \tilde{v}_{i,t} \quad \text{für } i \in \{0,2,4,8,12,16\}$$



Übersetzung von Hand- zu Zeigerposition:

SENSITIVITÄT := $\frac{\text{Änderung der ausgehenden Zeigerposition}}{\text{Änderung der eingehenden Handposition}}$

$$\sigma_t := \frac{\Delta \underline{p}_t^{L|R}}{\Delta \tilde{x}_t^{L|R}}$$

$$\rightarrow \Delta \underline{p}_t^{L|R} = \sigma_t \cdot \Delta \tilde{x}_t^{L|R} \quad \text{mit } \Delta \tilde{x}_t^{L|R} = \tilde{x}_t^{L|R} - \tilde{x}_{t-1}^{L|R} \quad \text{und } \Delta \underline{p}_t^{L|R} = \underline{p}_t^{L|R} - \underline{p}_{t-1}^{L|R}$$

- einfacher Ansatz: $\sigma_t = \sigma_0 = \text{const.}$
- Problem: genaue Positionierung bei kleinem Wert, großer Bildschirmbereich bei großem Wert

→ wünschenswert: BEIDES!

Nicht-lineare Sensitivitäts-Skalierung (vgl. Mauszeigerbeschleunigung):

SENSITIVITÄT = Skalierungskoeffizient \times Betrag der Positionsänderung

$$\sigma_t^\epsilon := \epsilon \cdot \left\| \Delta \tilde{x}_t^{L|R} \right\|$$

- großes Verhältnis bei schnellen, kleines Verhältnis bei langsamen Hand-Bewegungen

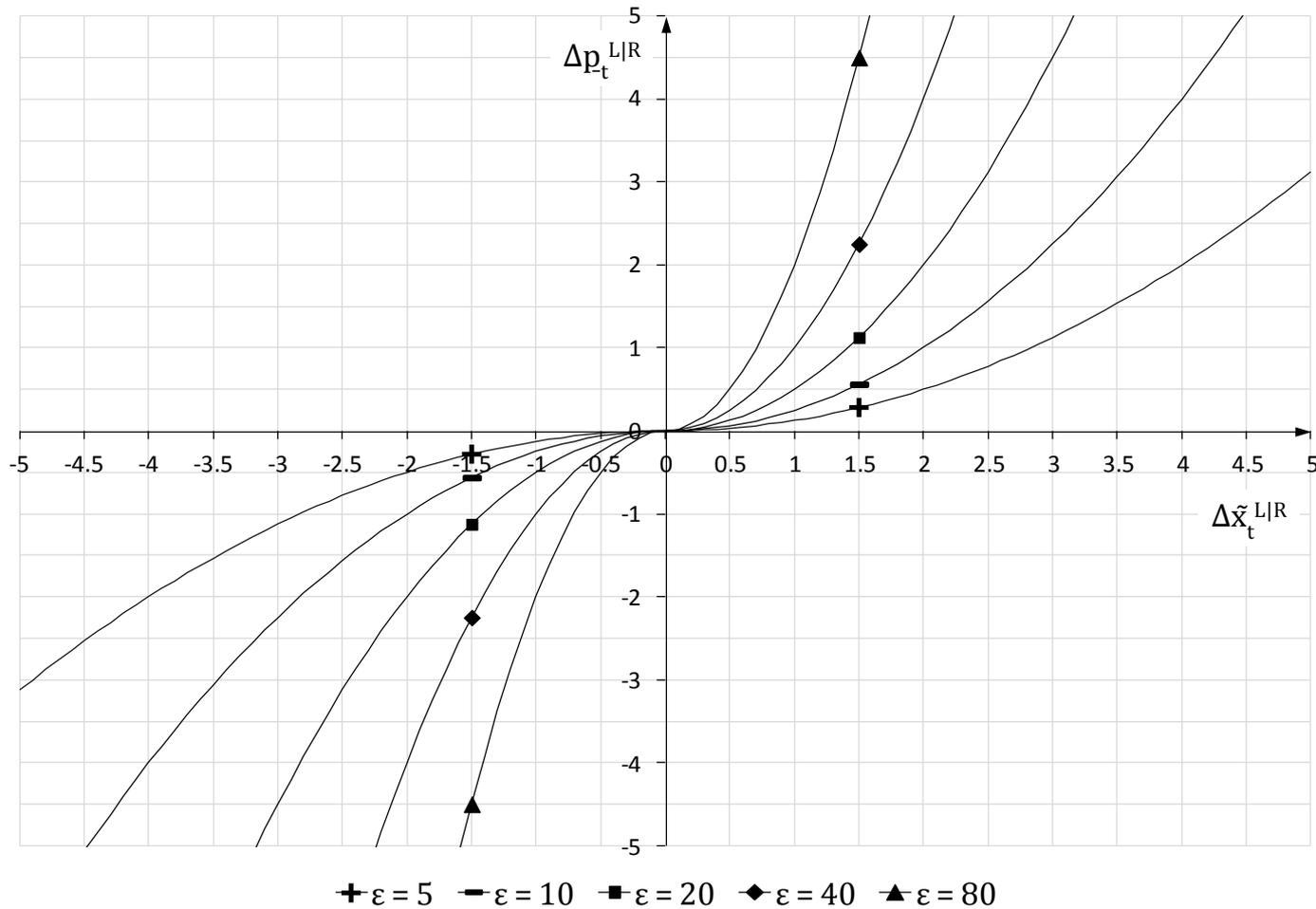
$$\rightarrow \Delta \underline{p}_t^{L|R} = \epsilon \cdot \left\| \Delta \tilde{x}_t^{L|R} \right\| \cdot \Delta \tilde{x}_t^{L|R}$$

$$\rightarrow \underline{p}_t^{L|R} = \epsilon \cdot \left\| \Delta \tilde{x}_t^{L|R} \right\| \cdot \Delta \tilde{x}_t^{L|R} + \underline{p}_{t-1}^{L|R}$$

- Wahl von ϵ empirisch, resultiert in 1:1 - Übersetzung bei einer „mittelschnellen“ Positionsänderung, entspricht 0,025 m pro Frame ($\sim 2,7$ km/h):

$$1 = \epsilon \cdot \left\| \Delta \tilde{x}_m^{L|R} \right\| \rightarrow \epsilon = \frac{1}{\left\| \Delta \tilde{x}_m^{L|R} \right\|} = \frac{1}{0,025} = 40$$

Nicht-lineare Sensitivitäts-Skalierung (vgl. Mauszeigerbeschleunigung):

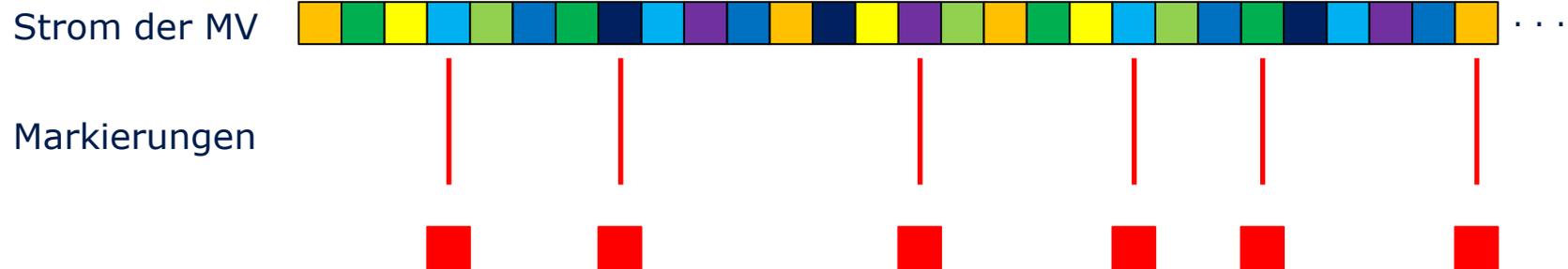


Vorgehen (II) – Suchfensteralgorithmus

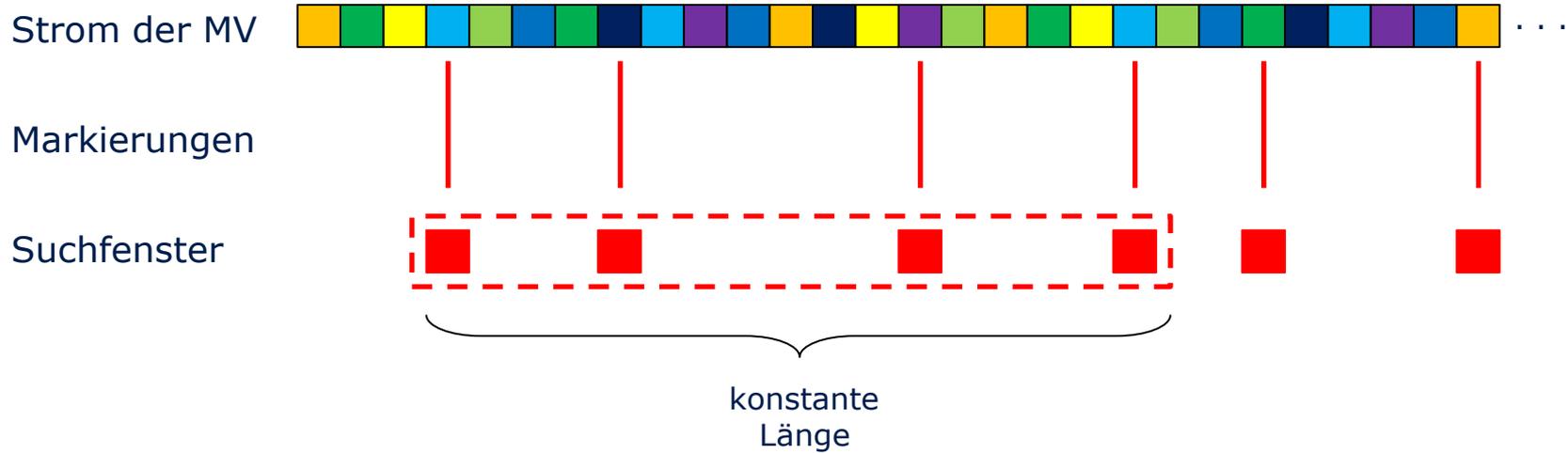
Nutzerinteraktion zur Laufzeit

Strom der MV  ...

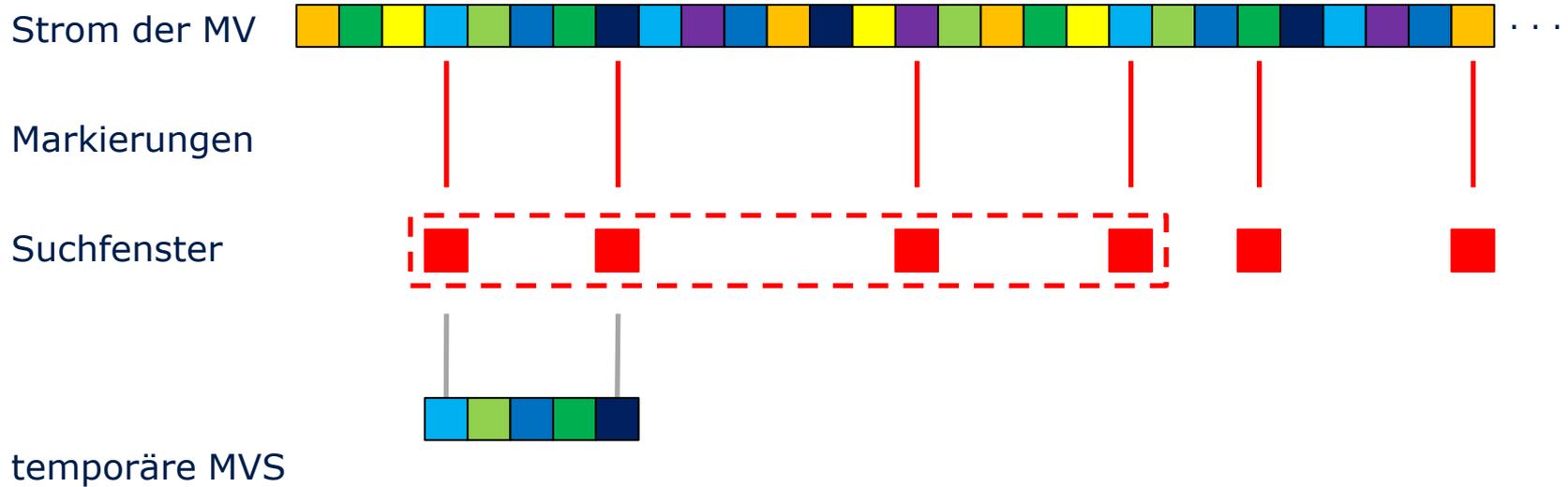
Vorgehen (II) – Suchfensteralgorithmus



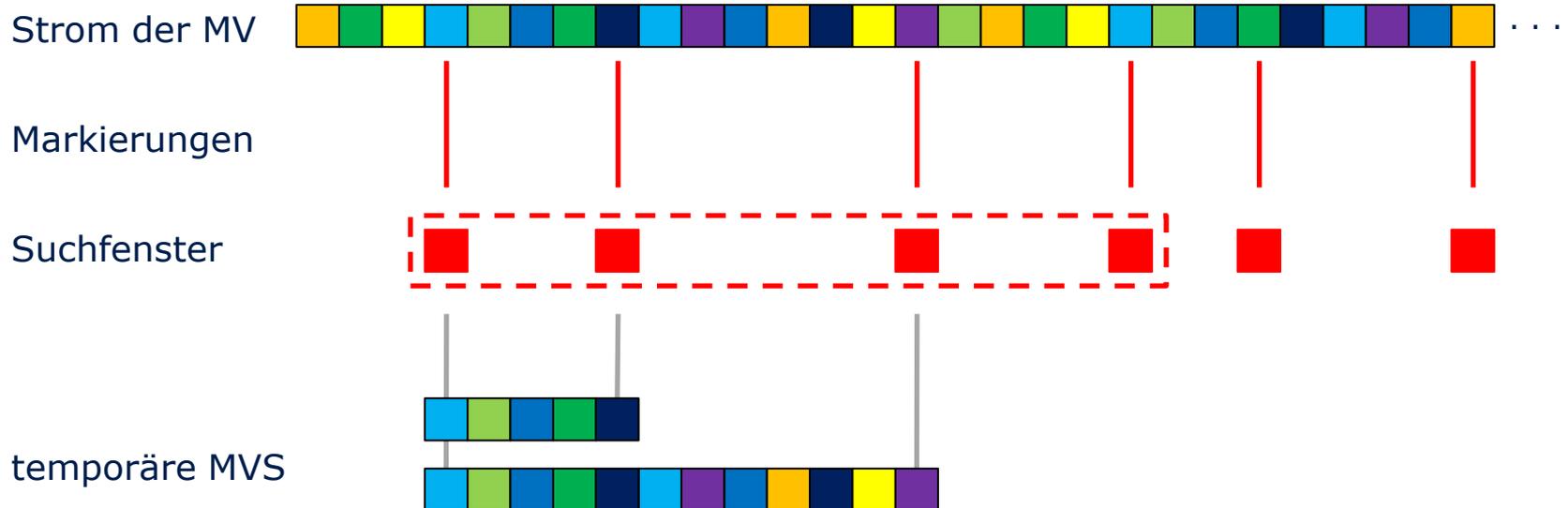
Vorgehen (II) – Suchfensteralgorithmus



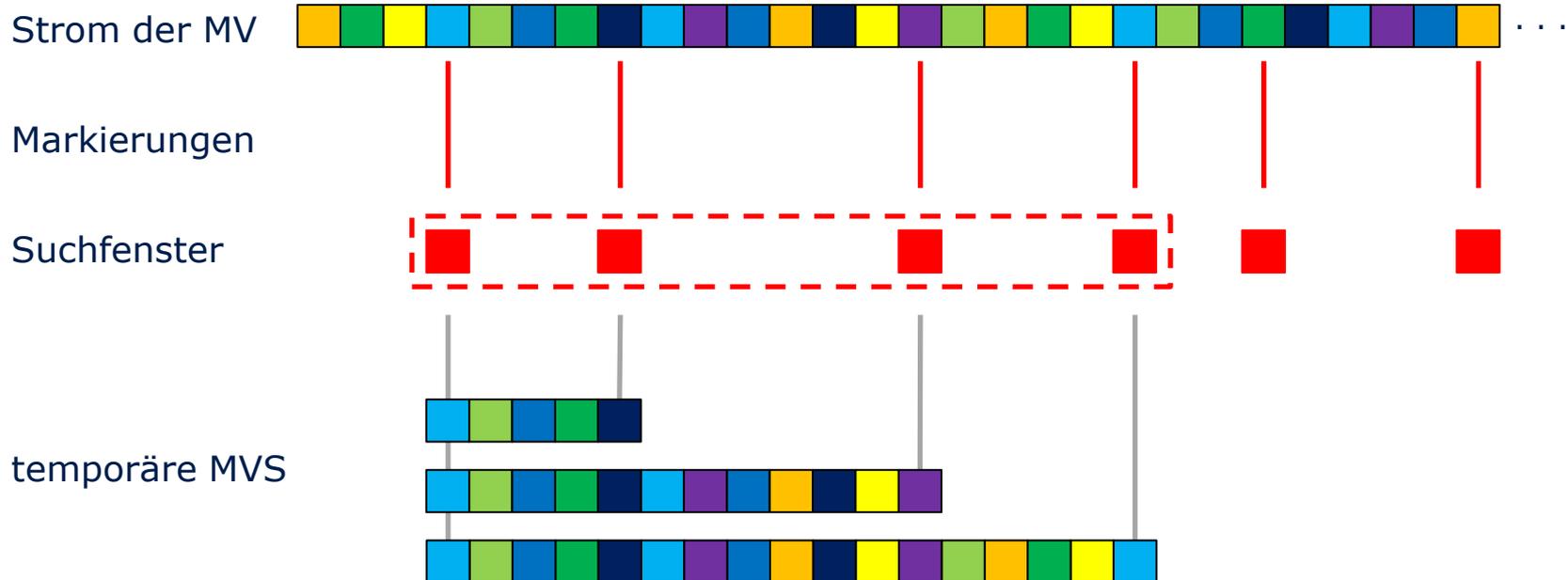
Vorgehen (II) – Suchfensteralgorithmus



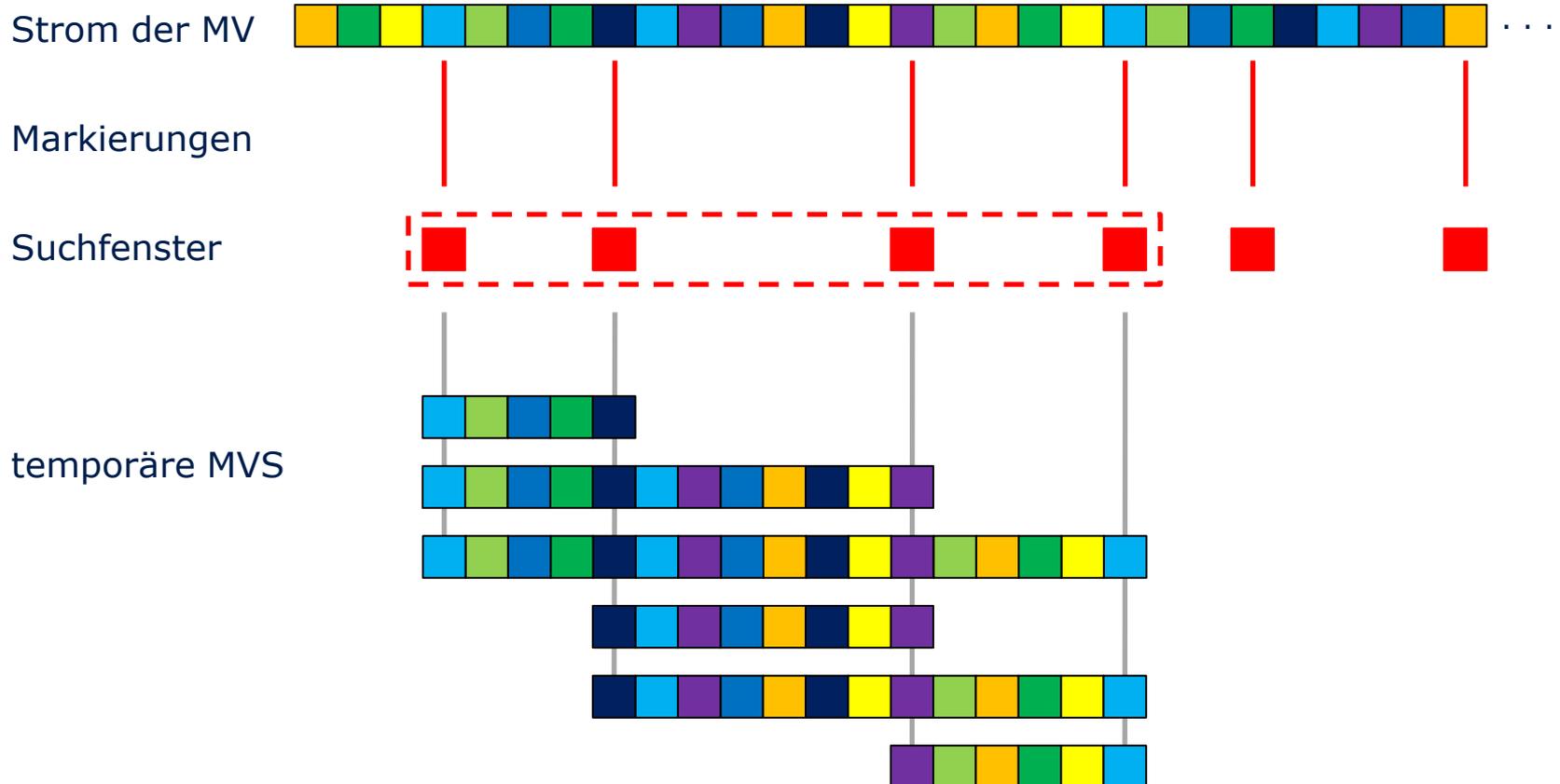
Vorgehen (II) – Suchfensteralgorithmus



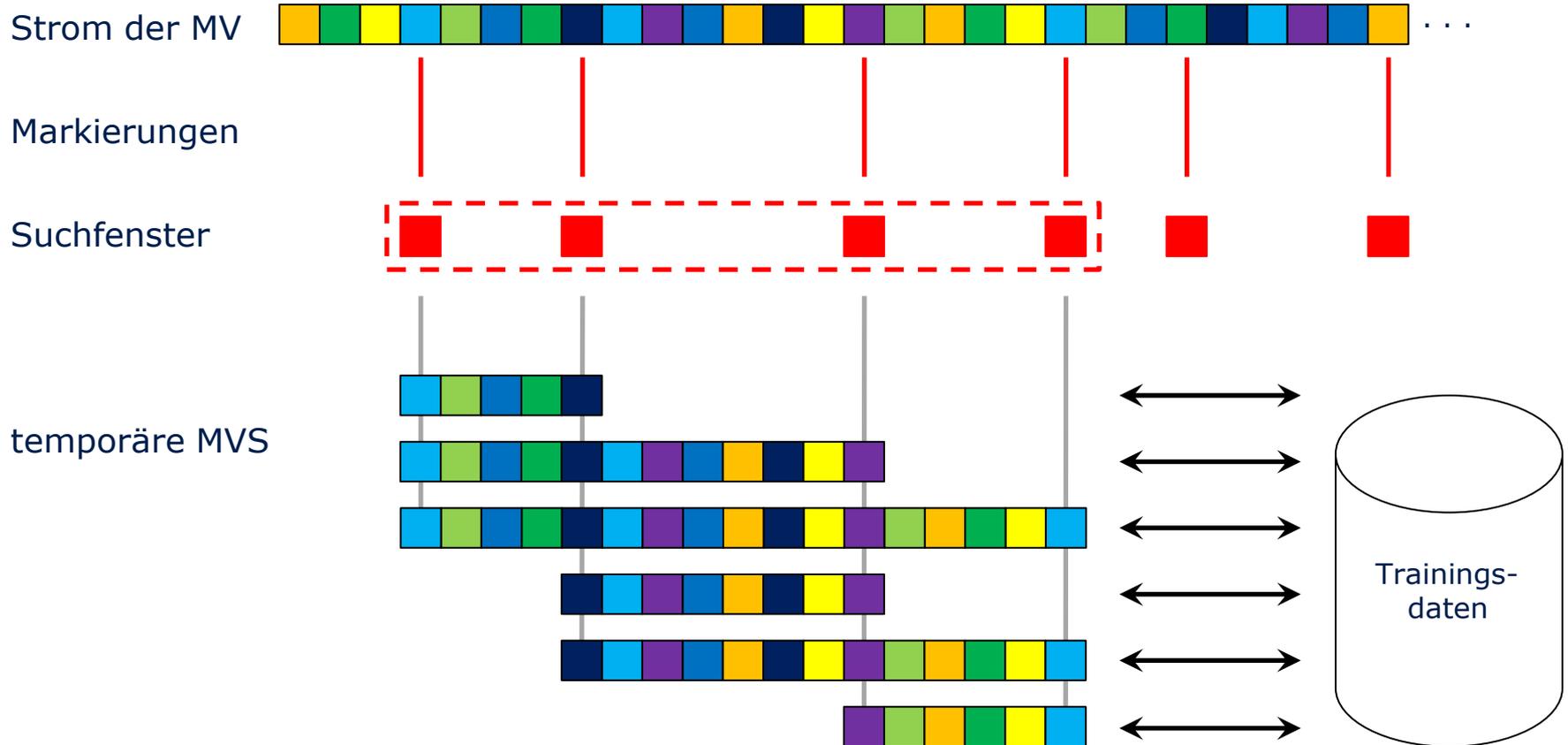
Vorgehen (II) – Suchfensteralgorithmus



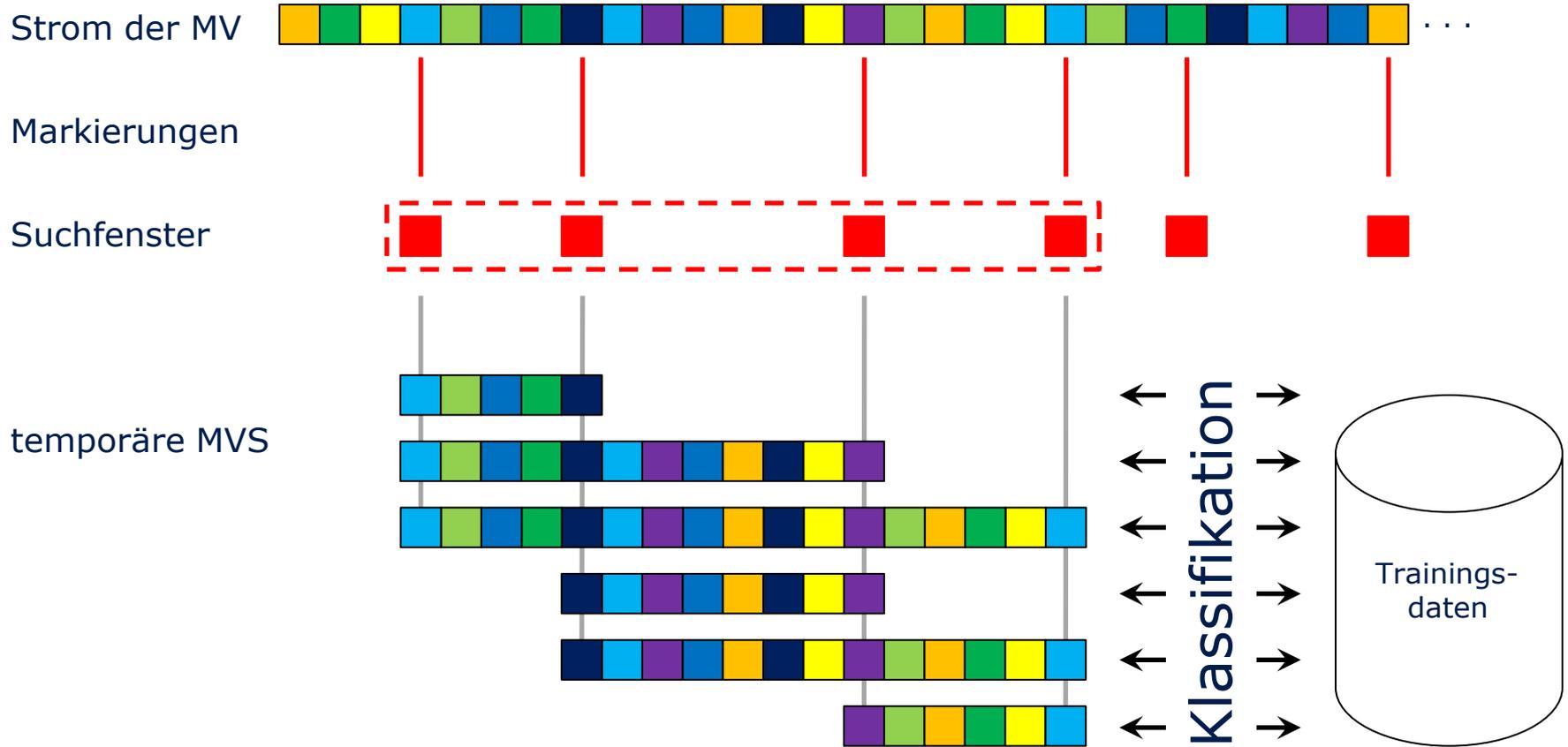
Vorgehen (II) – Suchfensteralgorithmus



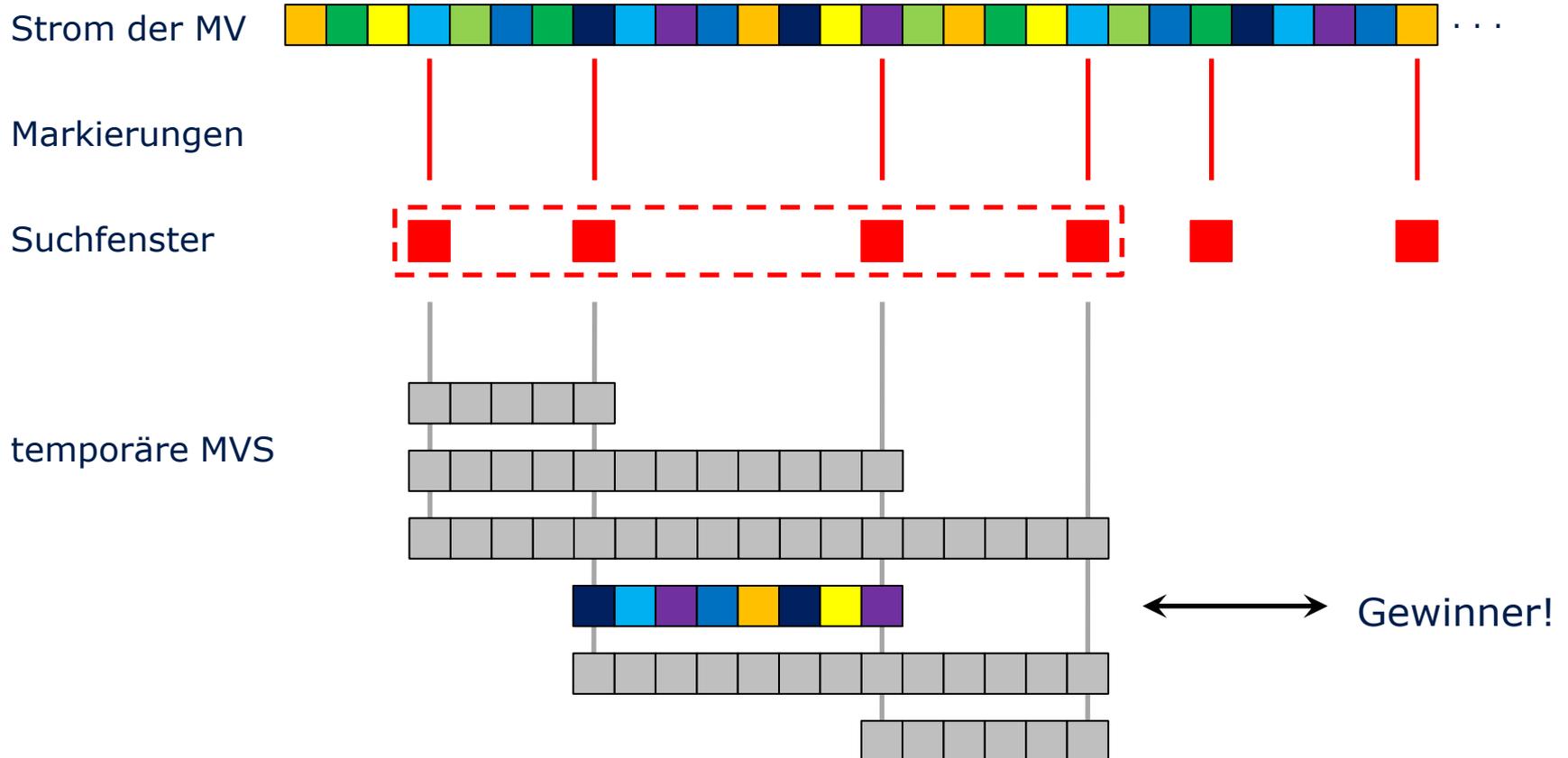
Vorgehen (II) – Suchfensteralgorithmus



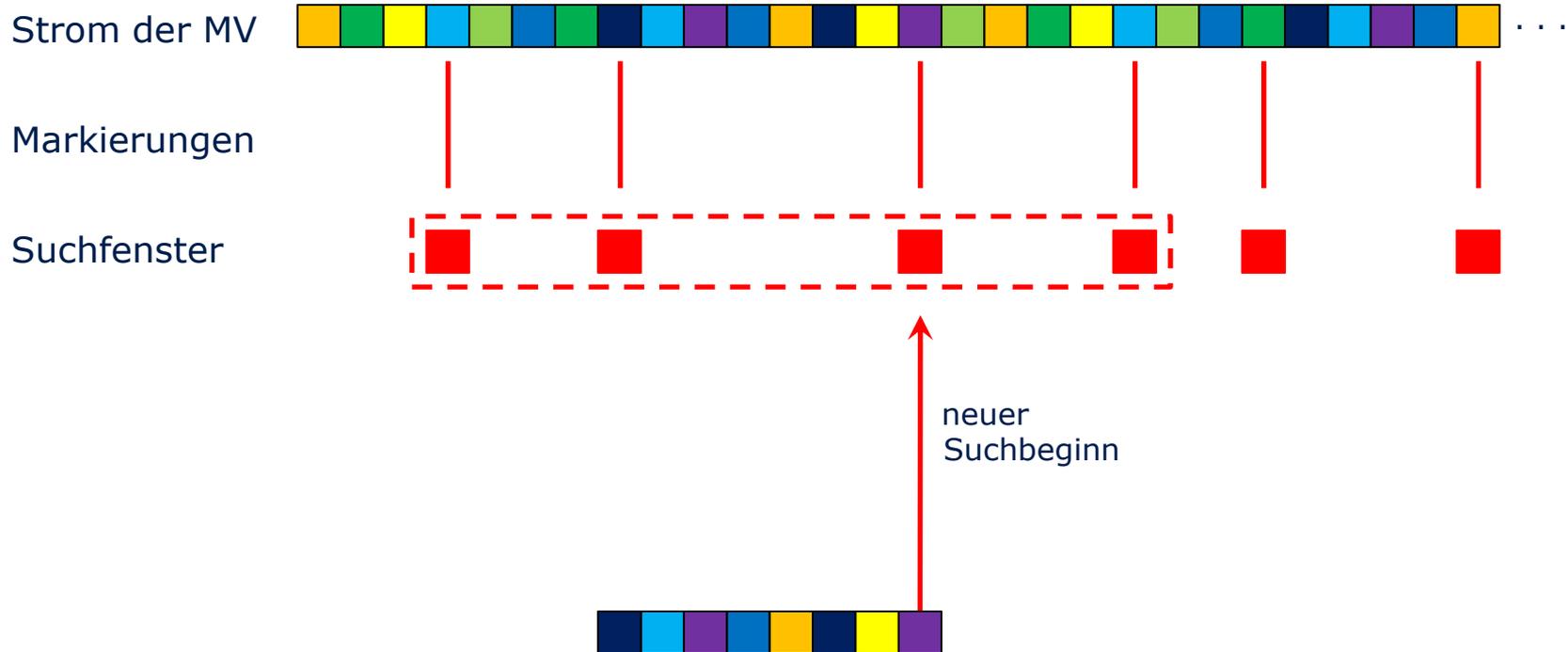
Vorgehen (II) – Suchfensteralgorithmus



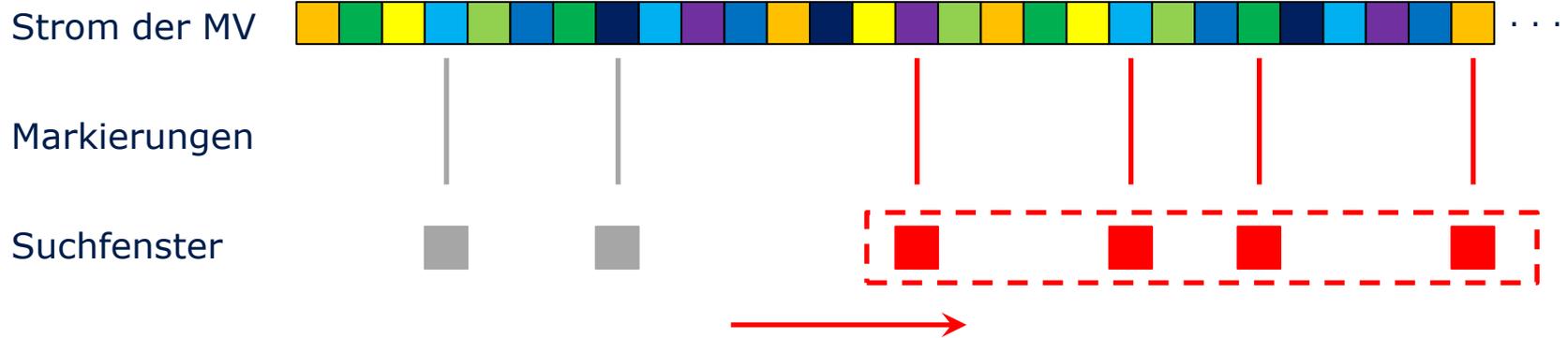
Vorgehen (II) – Suchfensteralgorithmus



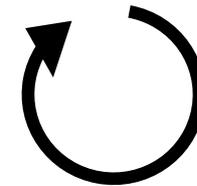
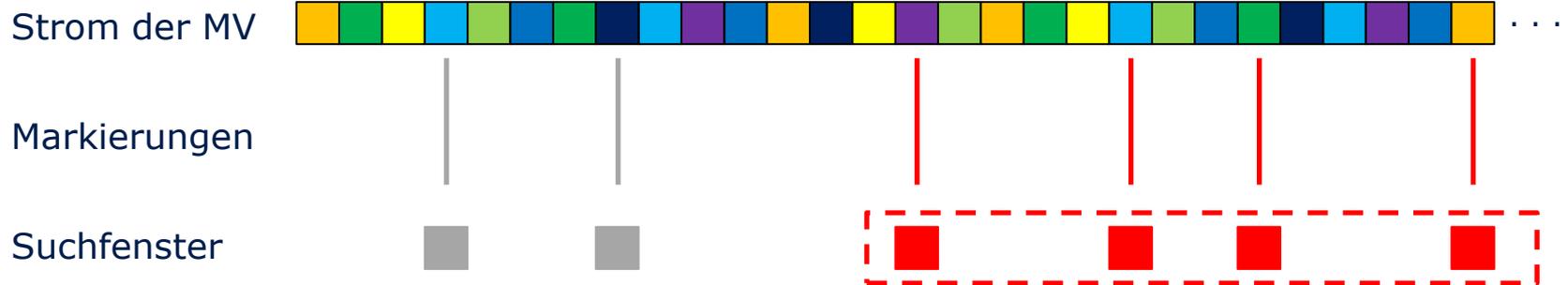
Vorgehen (II) – Suchfensteralgorithmus

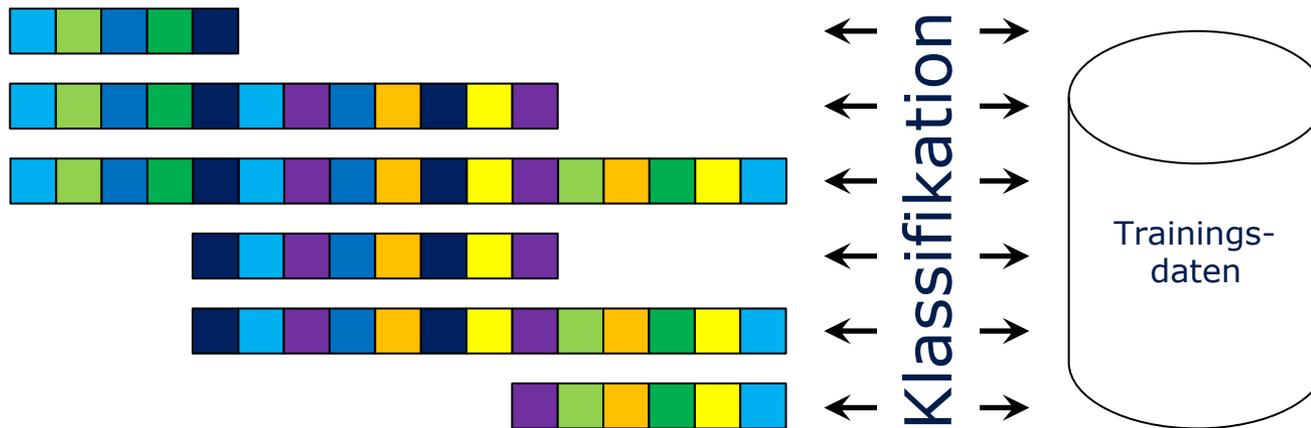


Vorgehen (II) – Suchfensteralgorithmus



Vorgehen (II) – Suchfensteralgorithmus





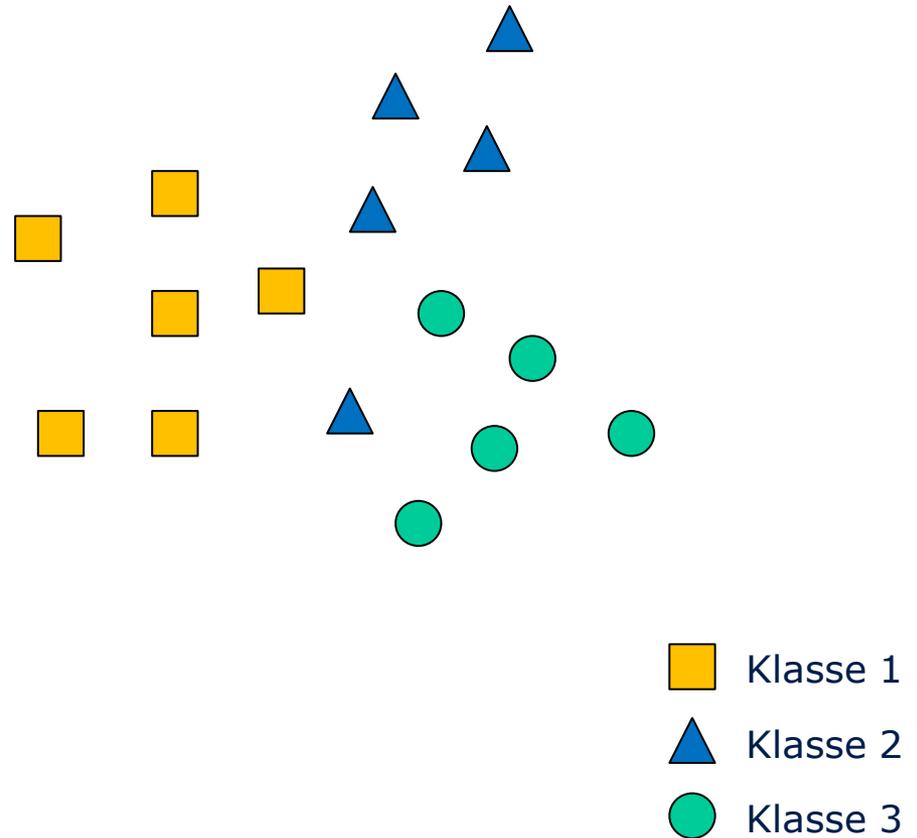
Klassifikation von MVS

k-Nearest-Neighbour-Algorithmus

Klassifikation von MVS

k-Nearest-Neighbour-Algorithmus

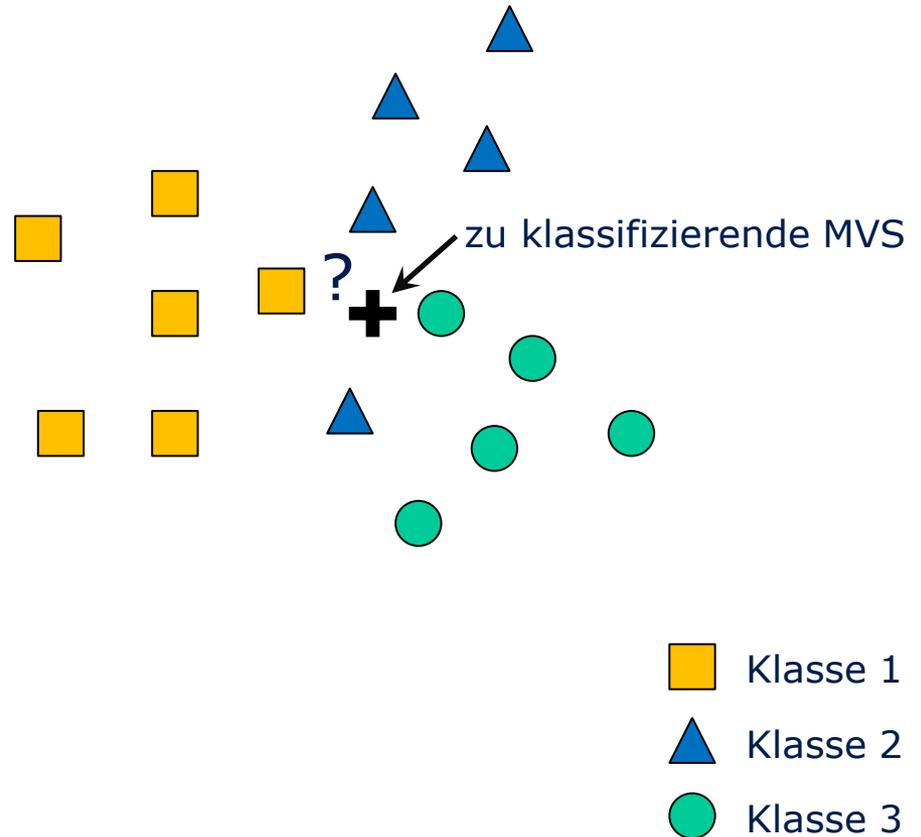
Raum der MVS



Klassifikation von MVS

k-Nearest-Neighbour-Algorithmus

Raum der MVS

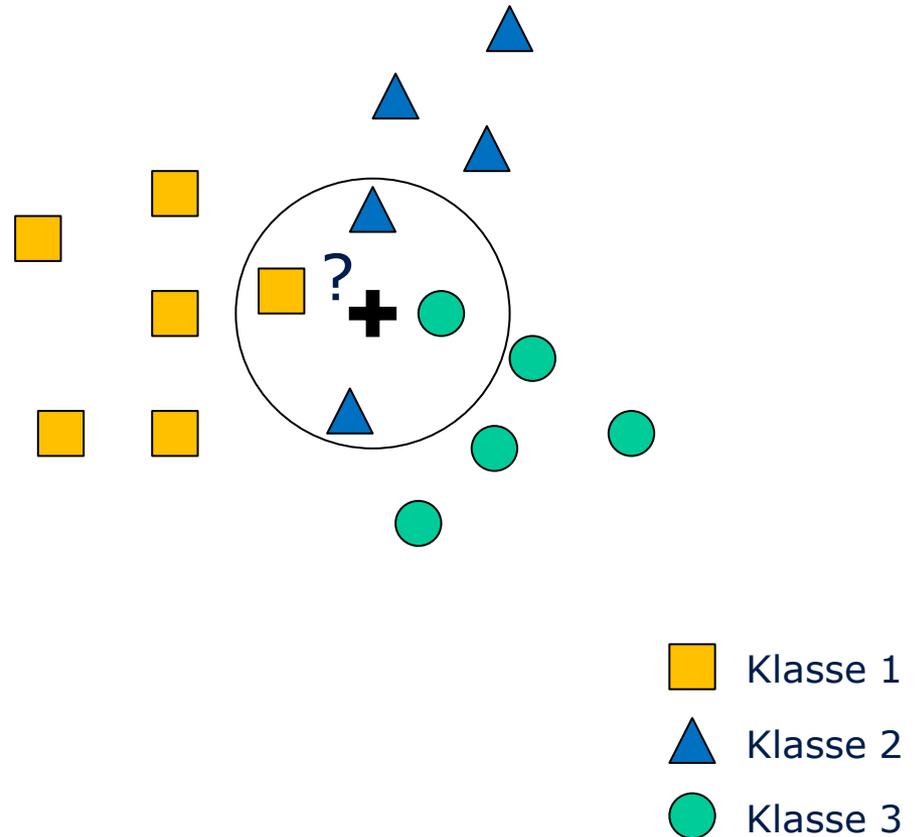


Klassifikation von MVS

k-Nearest-Neighbour-Algorithmus

- suche die k nächstliegenden Nachbarn

Raum der MVS

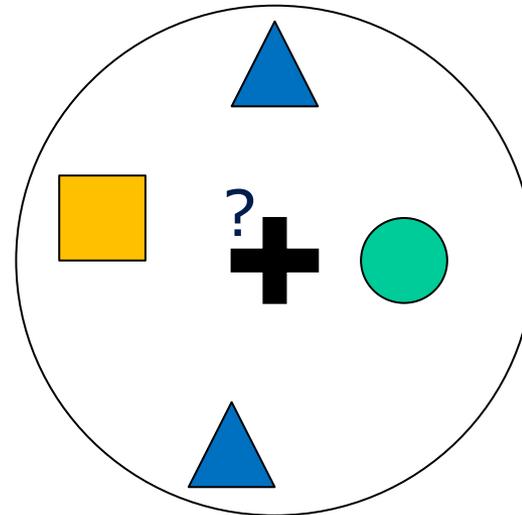


Klassifikation von MVS

k-Nearest-Neighbour-Algorithmus

- suche die k nächstliegenden Nachbarn

Raum der MVS



-  Klasse 1
-  Klasse 2
-  Klasse 3

Klassifikation von MVS

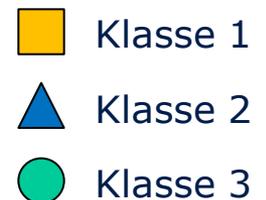
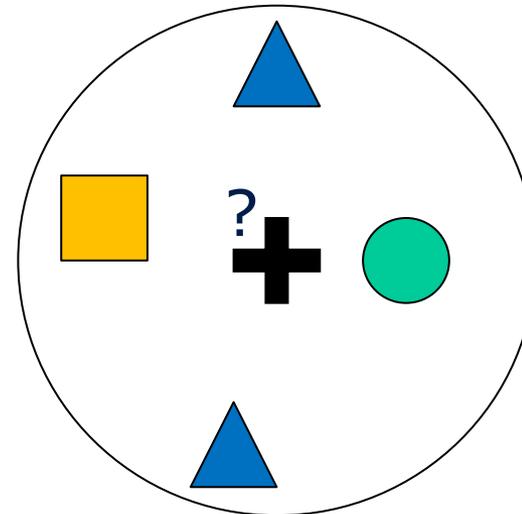
k-Nearest-Neighbour-Algorithmus

- suche die k nächstliegenden Nachbarn

- naive Klassifikation: Zählen

→ ▲ gewinnt

Raum der MVS



Klassifikation von MVS

k-Nearest-Neighbour-Algorithmus

Raum der MVS

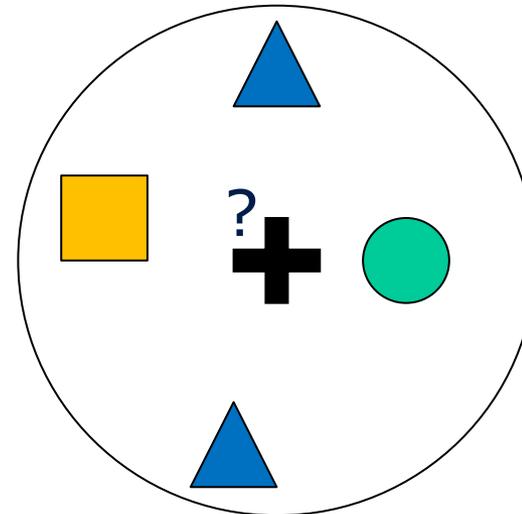
- suche die k nächstliegenden Nachbarn

- naive Klassifikation: Zählen

→  gewinnt

- besser: Gewichtete Summen

$$\text{Punkte der Klasse } X = \sum_{\text{MVS} \in X} \frac{1}{\text{Distanz zu MVS}}$$



-  Klasse 1
-  Klasse 2
-  Klasse 3

Klassifikation von MVS

k-Nearest-Neighbour-Algorithmus

Raum der MVS

- suche die k nächstliegenden Nachbarn

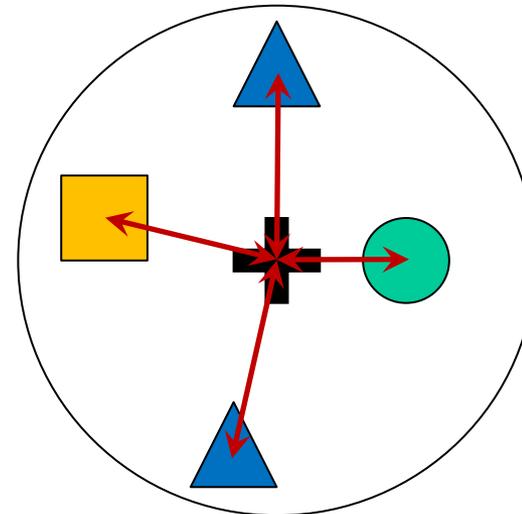
- naive Klassifikation: Zählen

→  gewinnt

- besser: Gewichtete Summen

$$\text{Punkte der Klasse } X = \sum_{\text{MVS} \in X} \frac{1}{\text{Distanz zu MVS}}$$

→  gewinnt

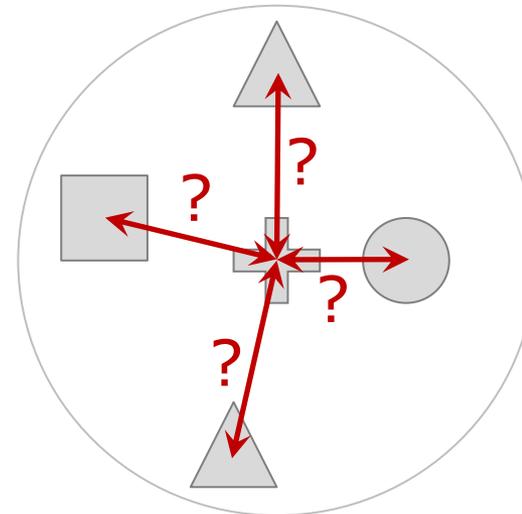


-  Klasse 1
-  Klasse 2
-  Klasse 3

Klassifikation von MVS

Raum der MVS

Wie werden die Distanzen berechnet?



-  Klasse 1
-  Klasse 2
-  Klasse 3

Anmerkungen:

- grundsätzlich keine Normalisierung der MV nötig, da Winkel in der gleichen Größenordnung liegen
- Studentisierung birgt die Gefahr, dass Rauschanteile deutlicheren Einfluss auf das Klassifikationsergebnis haben
- aber Skalierung entlang der Anzahl der MV zur Vergleichbarkeit unterschiedlich langer Gesten

Def. Merkmalsvektorsequenz von Zeitpunkt i bis Zeitpunkt j :

$$\Phi^{(i,j)} := (\tilde{\phi}_i, \tilde{\phi}_{i+1}, \dots, \tilde{\phi}_{j-1}, \tilde{\phi}_j) \quad \text{mit } i < j$$

$\tilde{\phi}_i \dots$ geglätteter
Merkmalsvektor

Metrik:

$$d(\vec{\phi}_i, \vec{\phi}_j) = \sqrt{\sum_{n=1}^8 (\phi_{n,i} - \phi_{n,j})^2}$$

$\Phi_k^{(i,j)}$... k-ter Eintrag in
der MVS

Euklidischer Abstand

$$\mathcal{D}_{\text{Euklid}}(\Phi^{(i,j)}, \Phi^{(g,h)}) = \frac{1}{N} \sqrt{\sum_{k=1}^N \left(d\left(\Phi_k^{(i,j)}, \Phi_k^{(g,h)}\right) \right)^2}$$

$$N := |\Phi^{(i,j)}| = |\Phi^{(g,h)}|$$

Pearson-Distanz

Korrelationskoeffizienten pro Merkmalsdimension m:

$\Phi_{m,k}^{(i,j)}$... k-ter Eintrag in
der m-ten
Dimension
einer MVS

$$\rho_m(\Phi^{(i,j)}, \Phi^{(g,h)}) := \frac{\sum_{k=1}^N (\Phi_{m,k}^{(i,j)} - \overline{\Phi_{m,k}^{(i,j)}}) (\Phi_{m,k}^{(g,h)} - \overline{\Phi_{m,k}^{(g,h)}})}{\sqrt{\sum_{k=1}^N (\Phi_{m,k}^{(i,j)} - \overline{\Phi_{m,k}^{(i,j)}})^2 (\Phi_{m,k}^{(g,h)} - \overline{\Phi_{m,k}^{(g,h)}})^2}}$$

$$N := |\Phi^{(i,j)}| = |\Phi^{(g,h)}|$$

$$\overline{\Phi_{m,k}^{(i,j)}} := \frac{1}{N} \sum_{k=1}^N \Phi_{m,k}^{(i,j)}$$

$$\rightarrow \mathcal{D}_{\text{Pearson}}(\Phi^{(i,j)}, \Phi^{(g,h)}) := \frac{1}{N} \sum_{m=1}^8 (1 - \rho_m(\Phi^{(i,j)}, \Phi^{(g,h)}))$$

FM-Distanz [OFH08]

Kinetische Energie pro Merkmalsdimension m:

$$E_{m,k} := m_m v_{m,k}^2 \quad \text{mit} \quad v_{m,k} = \frac{\Phi_{m,k}^{(i,j)} - \Phi_{m,k-1}^{(i,j)}}{\Delta t}$$

$\Phi_{m,k}^{(i,j)}$... k-ter Eintrag in der m-ten Dimension einer MVS

lässt sich vereinfachen zu:

$$E_{m,k} := \left(\Phi_{m,k}^{(i,j)} - \Phi_{m,k-1}^{(i,j)} \right)^2$$

mittlere Energie der m-ten Merkmalszeitreihe:

$$\overline{E}_m := \frac{1}{N-1} \sum_{k=2}^N E_{m,k} \quad \rightarrow \text{logarithmische Skalierung: } e_m := \log(\overline{E}_m + 1)$$

Energievektoren

$$\vec{e} := (e_1, e_2, \dots, e_8) \quad \rightarrow \mathcal{D}_{\text{FM}}(\Phi^{(i,j)}, \Phi^{(g,h)}) := \sqrt{\sum_{k=1}^8 \left(e_k^{(i,j)} - e_k^{(g,h)} \right)^2}$$

03 Gestenerkennung – Gesture Recognition

DYNAMIC TIME WARPING

- Voraussetzung: lokales Distanzmaß (auch Kostenfunktion genannt)

$$d: \vec{\phi} \times \vec{\phi} \rightarrow \mathbb{R}^+, \quad d(\vec{\phi}_i, \vec{\phi}_j) = \|\vec{\phi}_i - \vec{\phi}_j\|$$

- Input: zwei Merkmalsvektor-Zeitreihen, die aus den Posen einer Geste gewonnen werden

$$\boldsymbol{\phi}^{(ij)} = (\vec{\phi}_i, \vec{\phi}_{i+1}, \dots, \vec{\phi}_k, \dots, \vec{\phi}_{j-1}, \vec{\phi}_j) \text{ mit } \vec{\phi}_k = \varphi(P_k) \text{ für } P_k \in G^{(ij)}$$

$$\boldsymbol{\phi}_k^{(ij)} = \vec{\phi}_{i+k-1} \text{ mit } k \in [1, j - i + 1]$$

- Ziel des Algorithmus: Verknüpfen von zusammengehörenden Merkmalsvektorkopien aus beiden Zeitreihen, sodass die akkumulierte Distanz minimal wird

03 Gestenerkennung – Gesture Recognition

DYNAMIC TIME WARPING – Grundlagen

1. lokale Kostenmatrix C aller paarweisen Distanzen:

$$C \in \mathbb{R}^{M \times N}, C_{mn} := d\left(\phi_m^{(gh)}, \phi_n^{(ij)}\right), m \in [1, M], n \in [1, N]$$

2. Warping-Pfad Q : ist eine Folge von Indexpaaren (stellvertretend für die Merkmalsvektorpaare):

$$Q = (q_1, q_2, \dots, q_L) \text{ mit } q_l = (m_l, n_l) \text{ für } l \in [1, L]$$

• Distanz zwischen beiden Zeitreihen entspricht den akkumulierten Kosten entlang des Pfades Q :

$$\text{Distanz}_Q\left(\phi^{(gh)}, \phi^{(ij)}\right) = \sum_{l=1}^L d\left(\phi_{m_l}^{(gh)}, \phi_{n_l}^{(ij)}\right)$$

03 Gestenerkennung – Gesture Recognition

DYNAMIC TIME WARPING – Grundlagen

Warping-Pfad muss folgende Bedingungen erfüllen:

- Boundary – Kriterium: $q_1 = (1,1)$ und $q_L = (M, N)$
(Start- und Endindexpaar des Pfades muss Start- und Endindizes der beiden Zeitreihen entsprechen)
- Monotonie – Kriterium: $m_0 \leq m_1 \leq \dots \leq m_L$ und $n_0 \leq n_1 \leq \dots \leq n_L$
(Zeitordnung muss erhalten bleiben)
- Schrittgrößen – Kriterium: $q_{l+1} - q_l \in \{(0,1), (1,0), (1,1)\}$ für $l \in [1, L - 1]$
(Es dürfen keine Sprünge entlang des Pfades auftreten)

03 Gestenerkennung – Gesture Recognition

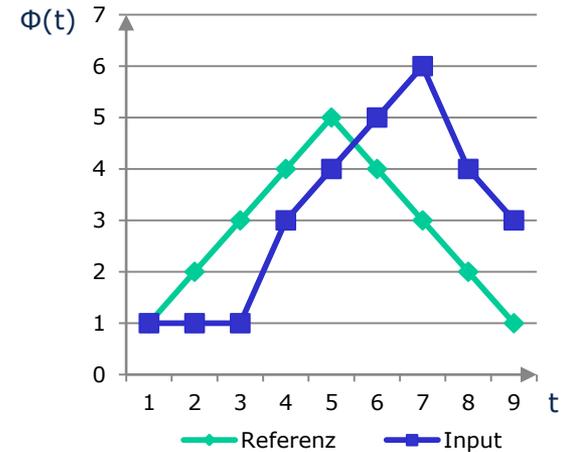
DYNAMIC TIME WARPING – Vorgehen

- Unter der Menge aller möglichen Pfade ist derjenige mit den minimalen Gesamtkosten gesucht
- Definition einer globalen Kostenmatrix $D \in \mathbb{R}^{M \times N}$:
- 1. Zeile: $D_{1,n} = \sum_{k=1}^n d(\phi_1^{(gh)}, \phi_k^{(ij)})$ für $n \in [1, N]$
- 1. Spalte: $D_{m,1} = \sum_{k=1}^m d(\phi_k^{(gh)}, \phi_1^{(ij)})$ für $m \in [1, M]$
- Sonst: $D_{m,n} = d(\phi_m^{(gh)}, \phi_n^{(ij)}) + \min\{D_{m-1,n-1}, D_{m-1,n}, D_{m,n-1}\}$ für $m \in [2, M]$ und $n \in [2, N]$
- Der Eintrag $D_{M,N}$ enthält die Gesamtkosten des optimalen Pfades (dieser kann explizit durch Backtracking ermittelt werden)
- liegen sie unter einem definierten Schwellwert, so werden die Gesten als übereinstimmend erkannt

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
	1	1	1	1	3	4	5	6	4	3



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

1. Zeile und 1. Spalte setzen

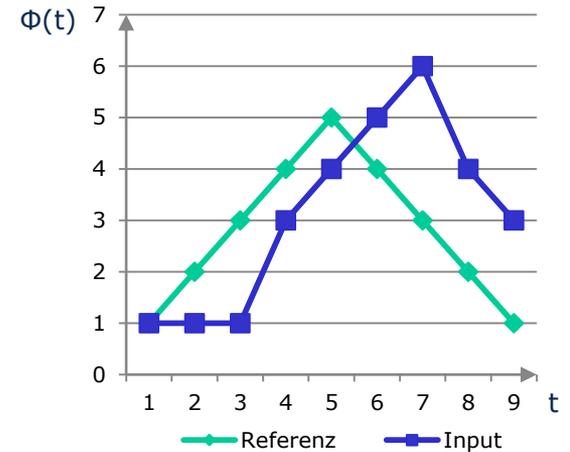
03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2	
	2	1	1	1	1	2	3	4	2	1	
	3	2	2	2	0	1	2	3	1	0	
	4	3	3	3	1	0	1	2	0	1	
	5	4	4	4	2	1	0	1	1	2	
	4	3	3	3	1	0	1	2	0	1	
	3	2	2	2	0	1	2	3	1	0	
	2	1	1	1	1	2	3	4	2	1	
	1	0	0	0	0	2	3	4	5	3	2
		1	1	1	3	4	5	6	4	3	
										Input	

Globale Distanzmatrix D:

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0



Distanz herkömmlich berechnet: d=13

1. Zeile und 1. Spalte setzen

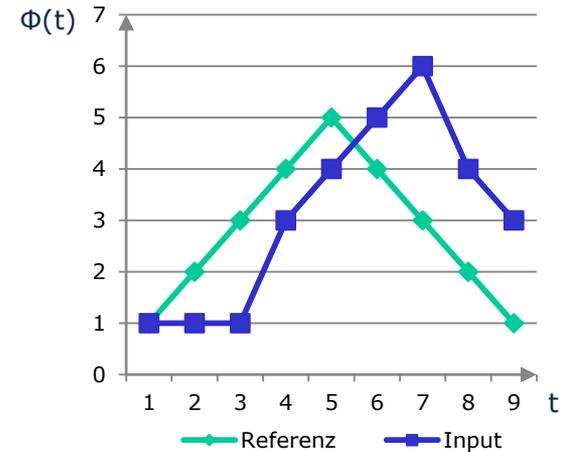
03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
		Input	1	1	1	3	4	5	6	4

Globale Distanzmatrix D:

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0



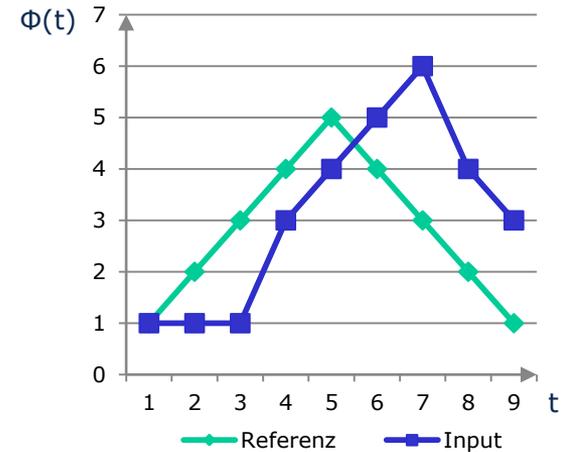
Distanz herkömmlich berechnet: d=13

1. Zeile und 1. Spalte setzen

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

	1	2	3	4	5	6	7	8	9
1	0	0	0	2	3	4	5	3	2
2	1	1	1	1	2	3	4	2	1
3	2	2	2	0	1	2	3	1	0
4	3	3	3	1	0	1	2	0	1
5	4	4	4	2	1	0	1	1	2
4	3	3	3	1	0	1	2	0	1
3	2	2	2	0	1	2	3	1	0
2	1	1	1	1	2	3	4	2	1
1	0	0	0	2	3	4	5	3	2
	1	1	1	3	4	5	6	4	3



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

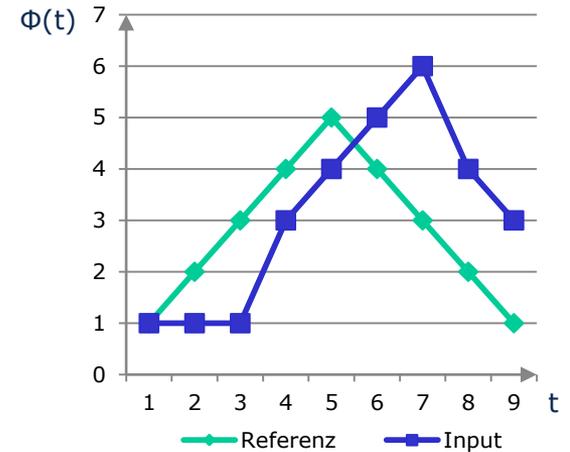
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	2	0	0	0	0	0	0

1. Zeile und 1. Spalte setzen

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2	
	2	1	1	1	1	2	3	4	2	1	
	3	2	2	2	0	1	2	3	1	0	
	4	3	3	3	1	0	1	2	0	1	
	5	4	4	4	2	1	0	1	1	2	
	4	3	3	3	1	0	1	2	0	1	
	3	2	2	2	0	1	2	3	1	0	
	2	1	1	1	1	2	3	4	2	1	
	1	0	0	0	0	2	3	4	5	3	2
		Input	1	1	1	3	4	5	6	4	3



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	2	5	0	0	0	0	0

1. Zeile und 1. Spalte setzen

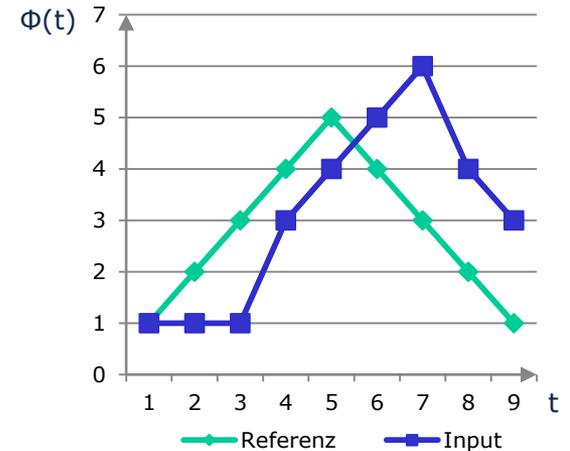
03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
		Input	1	1	1	3	4	5	6	4

Globale Distanzmatrix D:

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	2	5	9	0	0	0	0



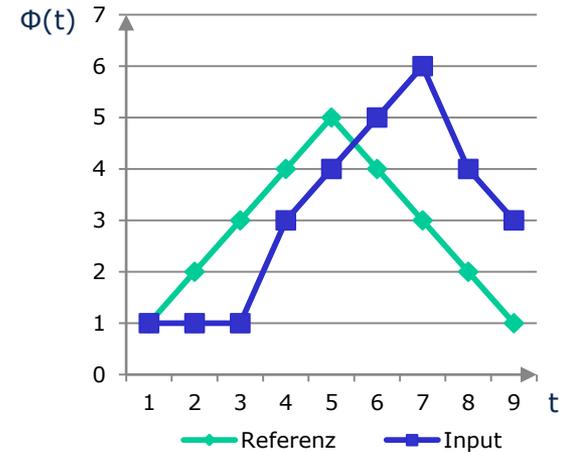
Distanz herkömmlich berechnet: $d=13$

1. Zeile und 1. Spalte setzen

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
		Input	1	1	1	3	4	5	6	4



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	2	5	9	14	0	0	0

1. Zeile und 1. Spalte setzen

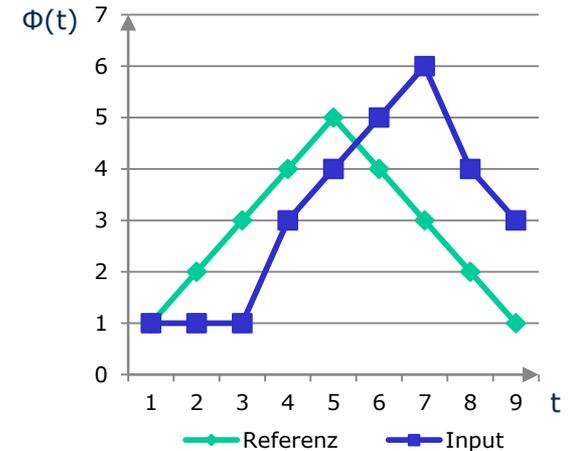
03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
		Input	1	1	1	3	4	5	6	4

Globale Distanzmatrix D:

0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	2	5	9	14	17	0	0



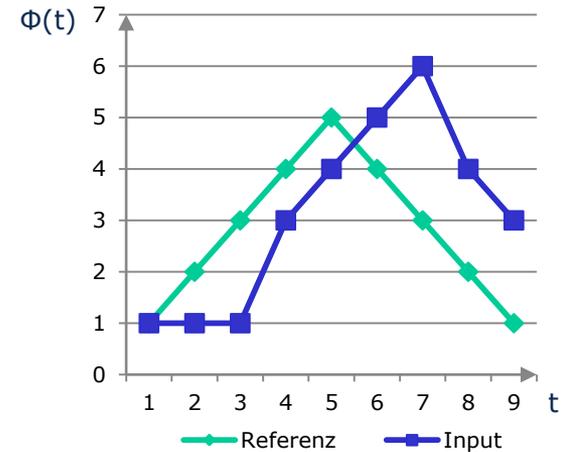
Distanz herkömmlich berechnet: d=13

1. Zeile und 1. Spalte setzen

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
	Input	1	1	1	3	4	5	6	4	3



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

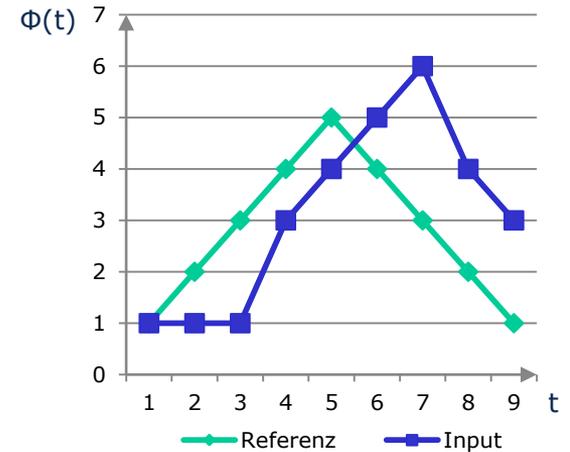
16	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
0	0	0	2	5	9	14	17	19

1. Zeile und 1. Spalte setzen

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
	1	1	1	1	3	4	5	6	4	3
		Input								



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

16	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0
0	0	0	0	2	5	9	14	17	19

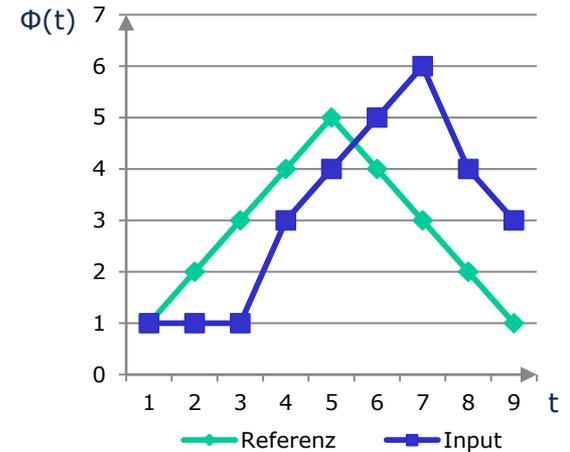
Restliche Einträge:

$$\text{Red} = \text{Yellow} + \min(\text{Green}, \text{Blue}, \text{Purple})$$

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
	1	1	1	1	3	4	5	6	4	3
		Input								



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

16	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0
0	0	0	2	5	9	14	17	19	

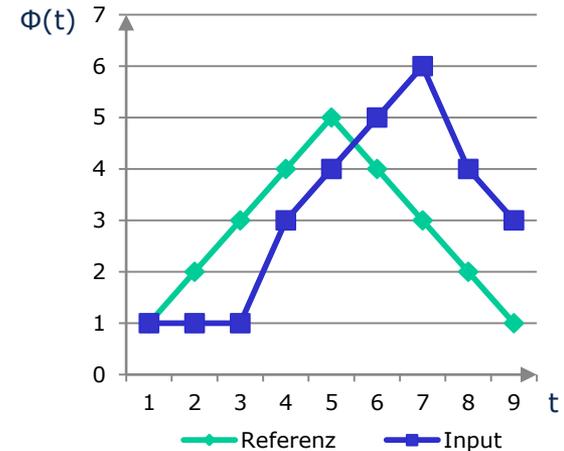
Restliche Einträge:

$$\text{Red} = \text{Yellow} + \min(\text{Green}, \text{Blue}, \text{Purple})$$

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2	
	2	1	1	1	1	2	3	4	2	1	
	3	2	2	2	0	1	2	3	1	0	
	4	3	3	3	1	0	1	2	0	1	
	5	4	4	4	2	1	0	1	1	2	
	4	3	3	3	1	0	1	2	0	1	
	3	2	2	2	0	1	2	3	1	0	
	2	1	1	1	1	2	3	4	2	1	
	1	0	0	0	0	2	3	4	5	3	2
	1	1	1	1	3	4	5	6	4	3	
		Input									



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

16	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
1	1	1	1	3	6	0	0	0	0
0	0	0	2	5	9	14	17	19	

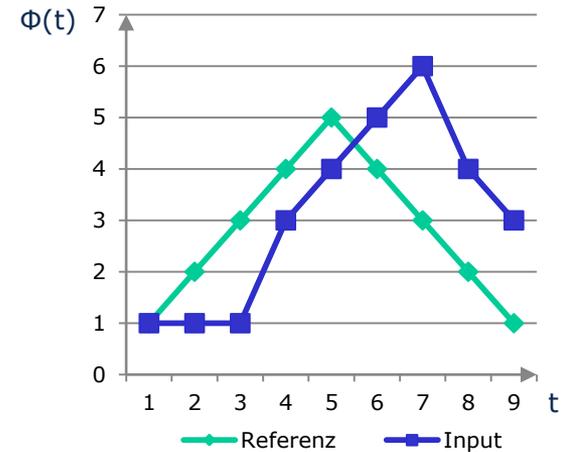
Restliche Einträge:

$$\text{Red} = \text{Yellow} + \min(\text{Green}, \text{Blue}, \text{Purple})$$

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2	
	2	1	1	1	1	2	3	4	2	1	
	3	2	2	2	0	1	2	3	1	0	
	4	3	3	3	1	0	1	2	0	1	
	5	4	4	4	2	1	0	1	1	2	
	4	3	3	3	1	0	1	2	0	1	
	3	2	2	2	0	1	2	3	1	0	
	2	1	1	1	1	2	3	4	2	1	
	1	0	0	0	0	2	3	4	5	3	2
	1	1	1	1	3	4	5	6	4	3	
		Input									



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

16	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
1	1	1	1	3	6	10	12	13	
0	0	0	2	5	9	14	17	19	

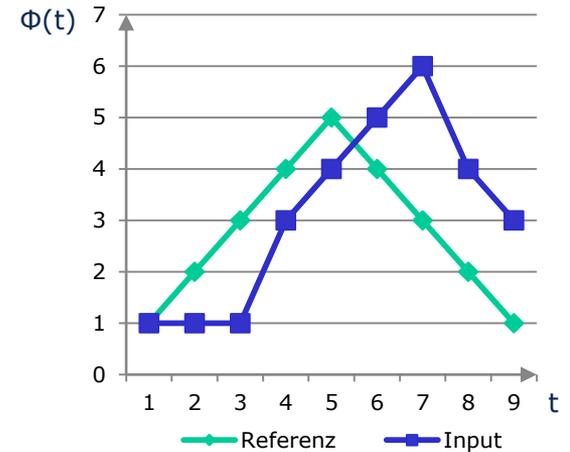
Restliche Einträge:

$$\text{Red} = \text{Yellow} + \min(\text{Green}, \text{Blue}, \text{Purple})$$

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
		Input	1	1	1	3	4	5	6	4



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

16	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
6	6	0	0	0	0	0	0	0	0
3	3	3	1	2	4	7	8	8	8
1	1	1	1	3	6	10	12	13	13
0	0	0	2	5	9	14	17	19	19

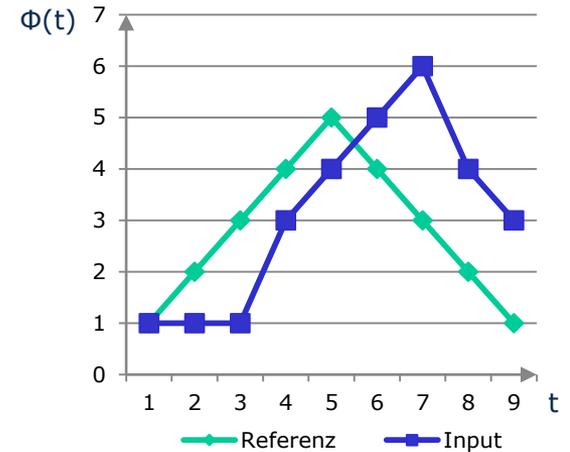
Restliche Einträge:

$$\text{Red} = \text{Yellow} + \min(\text{Green}, \text{Blue}, \text{Purple})$$

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
		1	1	1	3	4	5	6	4	3
		Input								



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

16	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
6	6	6	2	1	2	4	4	5	
3	3	3	1	2	4	7	8	8	
1	1	1	1	3	6	10	12	13	
0	0	0	2	5	9	14	17	19	

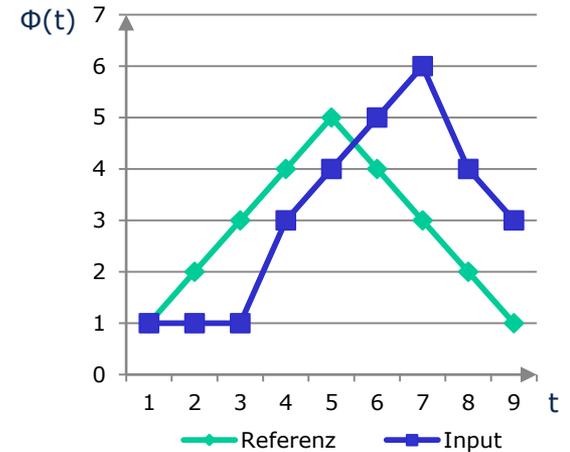
Restliche Einträge:

$$\text{Red} = \text{Yellow} + \min(\text{Green}, \text{Blue}, \text{Purple})$$

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
		Input	1	1	1	3	4	5	6	4



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

16	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0
15	15	0	0	0	0	0	0	0
13	13	13	5	2	2	3	2	3
10	10	10	4	2	1	2	3	5
6	6	6	2	1	2	4	4	5
3	3	3	1	2	4	7	8	8
1	1	1	1	3	6	10	12	13
0	0	0	2	5	9	14	17	19

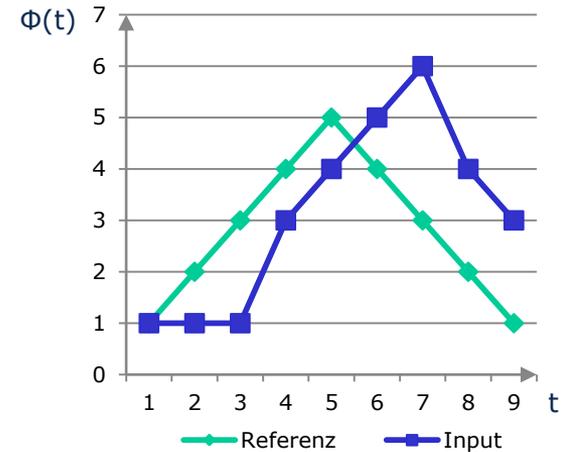
Restliche Einträge:

$$\text{Red} = \text{Yellow} + \min(\text{Green}, \text{Blue}, \text{Purple})$$

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
		1	1	1	3	4	5	6	4	3
										Input



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

16	16	16	8	8	9	11	8	5
16	16	16	6	5	6	8	5	3
15	15	15	5	3	4	5	3	2
13	13	13	5	2	2	3	2	3
10	10	10	4	2	1	2	3	5
6	6	6	2	1	2	4	4	5
3	3	3	1	2	4	7	8	8
1	1	1	1	3	6	10	12	13
0	0	0	2	5	9	14	17	19

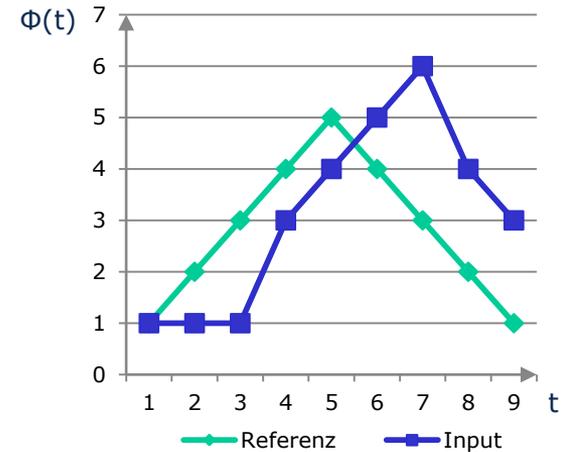
Restliche Einträge:

$$\text{Red} = \text{Yellow} + \min(\text{Green}, \text{Blue}, \text{Purple})$$

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
		Input	1	1	1	3	4	5	6	4



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

16	16	16	8	8	9	11	8	5
16	16	16	6	5	6	8	5	3
15	15	15	5	3	4	5	3	2
13	13	13	5	2	2	3	2	3
10	10	10	4	2	1	2	3	5
6	6	6	2	1	2	4	4	5
3	3	3	1	2	4	7	8	8
1	1	1	1	3	6	10	12	13
0	0	0	2	5	9	14	17	19

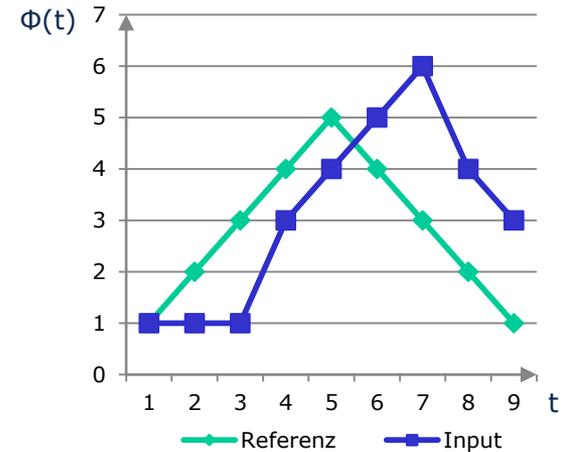
Backtracking:

= min (, ,)

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
		Input	1	1	1	3	4	5	6	4



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

16	16	16	8	8	9	11	8	5
16	16	16	6	5	6	8	5	3
15	15	15	5	3	4	5	3	2
13	13	13	5	2	2	3	2	3
10	10	10	4	2	1	2	3	5
6	6	6	2	1	2	4	4	5
3	3	3	1	2	4	7	8	8
1	1	1	1	3	6	10	12	13
0	0	0	2	5	9	14	17	19

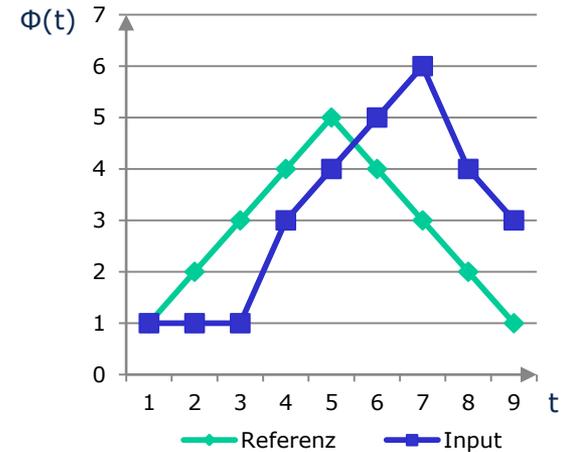
Backtracking:

= min (, ,)

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
		Input	1	1	1	3	4	5	6	4



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

16	16	16	8	8	9	11	8	5
16	16	16	6	5	6	8	5	3
15	15	15	5	3	4	5	3	2
13	13	13	5	2	2	3	2	3
10	10	10	4	2	1	2	3	5
6	6	6	2	1	2	4	4	5
3	3	3	1	2	4	7	8	8
1	1	1	1	3	6	10	12	13
0	0	0	2	5	9	14	17	19

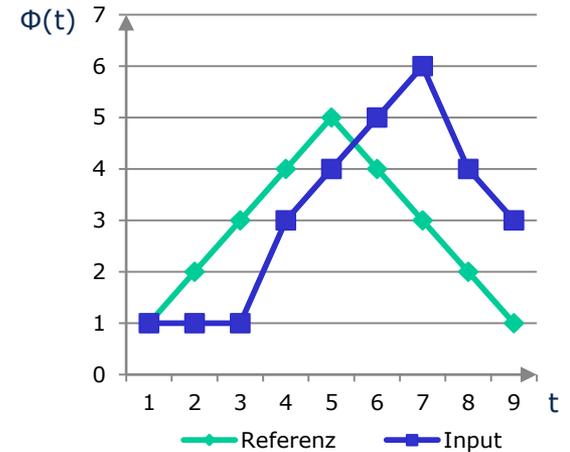
Backtracking:

= min (, ,)

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
		Input	1	1	1	3	4	5	6	4



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

16	16	16	8	8	9	11	8	5
16	16	16	6	5	6	8	5	3
15	15	15	5	3	4	5	3	2
13	13	13	5	2	2	3	2	3
10	10	10	4	2	1	2	3	5
6	6	6	2	1	2	4	4	5
3	3	3	1	2	4	7	8	8
1	1	1	1	3	6	10	12	13
0	0	0	2	5	9	14	17	19

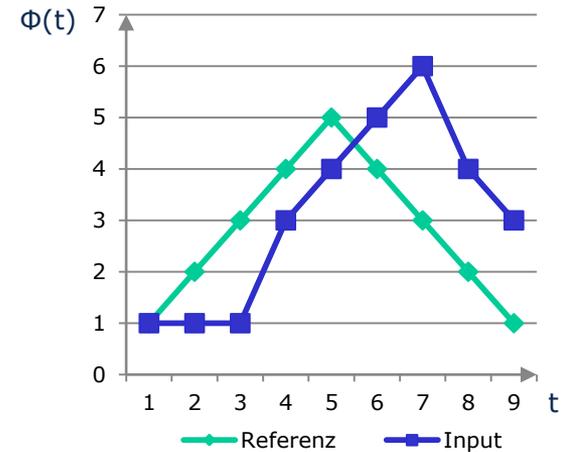
Backtracking:

= min (, ,)

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
		Input	1	1	1	3	4	5	6	4



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

16	16	16	8	8	9	11	8	5
16	16	16	6	5	6	8	5	3
15	15	15	5	3	4	5	3	2
13	13	13	5	2	2	3	2	3
10	10	10	4	2	1	2	3	5
6	6	6	2	1	2	4	4	5
3	3	3	1	2	4	7	8	8
1	1	1	1	3	6	10	12	13
0	0	0	2	5	9	14	17	19

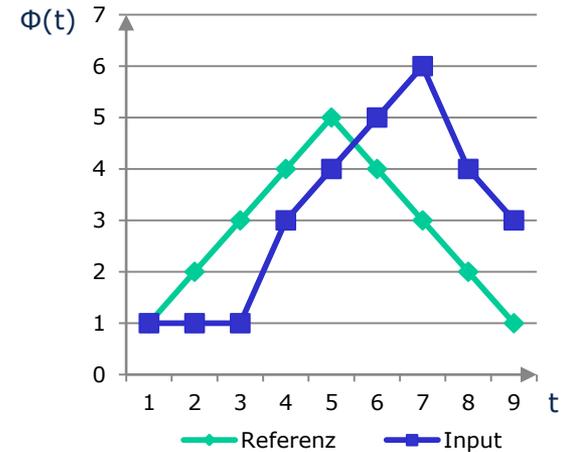
Backtracking:

= min (, ,)

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
		Input	1	1	1	3	4	5	6	4



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

16	16	16	8	8	9	11	8	5
16	16	16	6	5	6	8	5	3
15	15	15	5	3	4	5	3	2
13	13	13	5	2	2	3	2	3
10	10	10	4	2	1	2	3	5
6	6	6	2	1	2	4	4	5
3	3	3	1	2	4	7	8	8
1	1	1	1	3	6	10	12	13
0	0	0	2	5	9	14	17	19

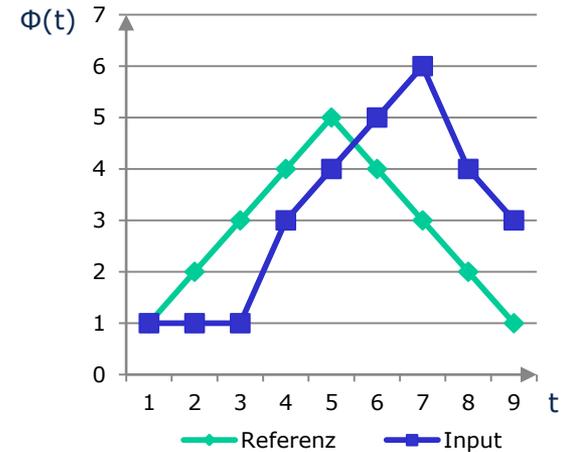
Backtracking:

= min (, ,)

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
		Input	1	1	1	3	4	5	6	4



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

16	16	16	8	8	9	11	8	5
16	16	16	6	5	6	8	5	3
15	15	15	5	3	4	5	3	2
13	13	13	5	2	2	3	2	3
10	10	10	4	2	1	2	3	5
6	6	6	2	1	2	4	4	5
3	3	3	1	2	4	7	8	8
1	1	1	1	3	6	10	12	13
0	0	0	2	5	9	14	17	19

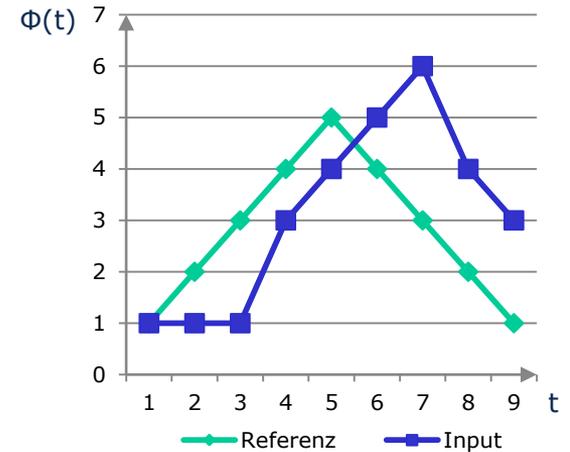
Backtracking:

= min (, ,)

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
		Input	1	1	1	3	4	5	6	4



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

16	16	16	8	8	9	11	8	5
16	16	16	6	5	6	8	5	3
15	15	15	5	3	4	5	3	2
13	13	13	5	2	2	3	2	3
10	10	10	4	2	1	2	3	5
6	6	6	2	1	2	4	4	5
3	3	3	1	2	4	7	8	8
1	1	1	1	3	6	10	12	13
0	0	0	2	5	9	14	17	19

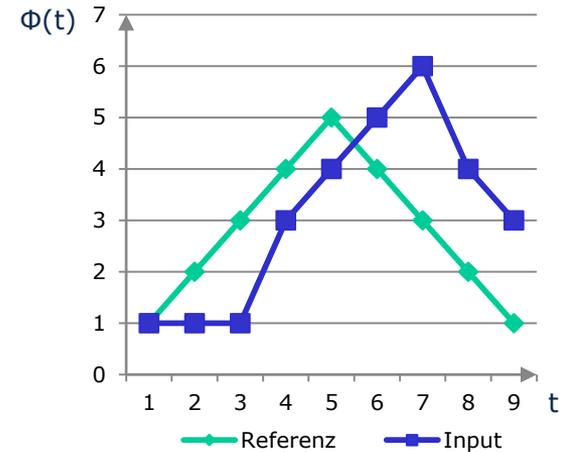
Backtracking:

= min (, ,)

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
		Input	1	1	1	3	4	5	6	4



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

16	16	16	8	8	9	11	8	5
16	16	16	6	5	6	8	5	3
15	15	15	5	3	4	5	3	2
13	13	13	5	2	2	3	2	3
10	10	10	4	2	1	2	3	5
6	6	6	2	1	2	4	4	5
3	3	3	3	1	2	4	7	8
1	1	1	1	1	3	6	10	12
0	0	0	2	5	9	14	17	19

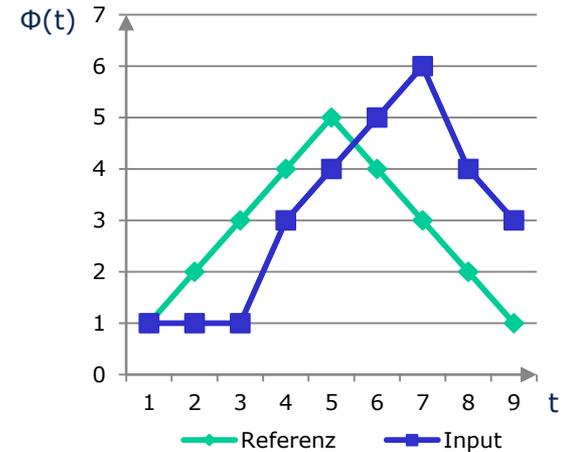
Backtracking:

= min (, ,)

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
	1	1	1	1	3	4	5	6	4	3
		Input								



Distanz herkömmlich berechnet: $d=13$

Globale Distanzmatrix D:

16	16	16	8	8	9	11	8	5
16	16	16	6	5	6	8	5	3
15	15	15	5	3	4	5	3	2
13	13	13	5	2	2	3	2	3
10	10	10	4	2	1	2	3	5
6	6	6	2	1	2	4	4	5
3	3	3	1	2	4	7	8	8
1	1	1	1	3	6	10	12	13
0	0	0	2	5	9	14	17	19

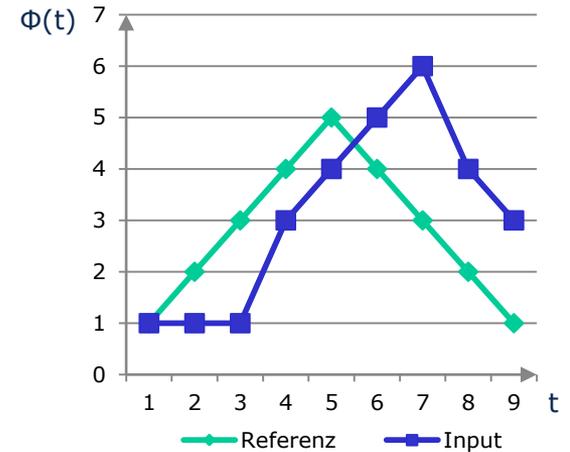
Backtracking:

= min (, ,)

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
		Input	1	1	1	3	4	5	6	4



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

16	16	16	8	8	9	11	8	5
16	16	16	6	5	6	8	5	3
15	15	15	5	3	4	5	3	2
13	13	13	5	2	2	3	2	3
10	10	10	4	2	1	2	3	5
6	6	6	2	1	2	4	4	5
3	3	3	1	2	4	7	8	8
1	1	1	1	3	6	10	12	13
0	0	0	2	5	9	14	17	19

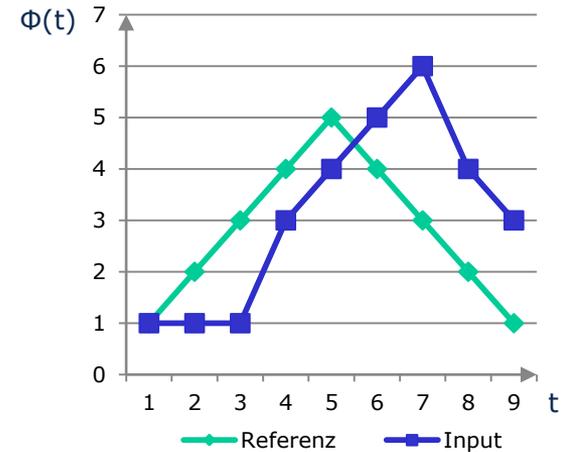
Backtracking:

= min (, ,)

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
		Input	1	1	1	3	4	5	6	4



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

16	16	16	8	8	9	11	8	5
16	16	16	6	5	6	8	5	3
15	15	15	5	3	4	5	3	2
13	13	13	5	2	2	3	2	3
10	10	10	4	2	1	2	3	5
6	6	6	2	1	2	4	4	5
3	3	3	1	2	4	7	8	8
1	1	1	1	3	6	10	12	13
0	0	0	2	5	9	14	17	19

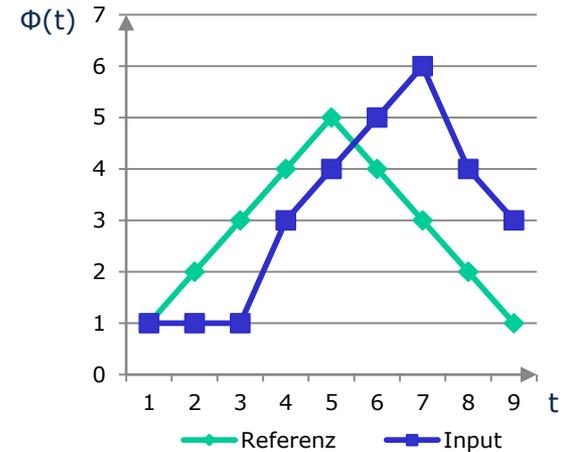
Backtracking:

= min (, ,)

03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
		Input	1	1	1	3	4	5	6	4



Distanz herkömmlich berechnet: d=13

Globale Distanzmatrix D:

16	16	16	8	8	9	11	8	5
16	16	16	6	5	6	8	5	3
15	15	15	5	3	4	5	3	2
13	13	13	5	2	2	3	2	3
10	10	10	4	2	1	2	3	5
6	6	6	2	1	2	4	4	5
3	3	3	1	2	4	7	8	8
1	1	1	1	3	6	10	12	13
0	0	0	2	5	9	14	17	19

- Alternativer Pfad

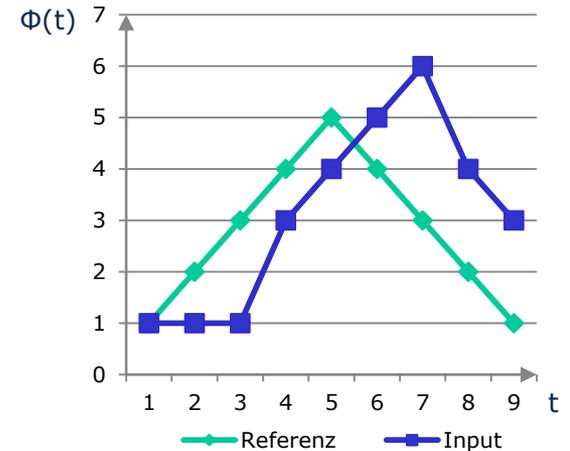
03 Gestenerkennung – Bsp.

Lokale Distanzmatrix C:

Referenz	1	0	0	0	2	3	4	5	3	2
	2	1	1	1	1	2	3	4	2	1
	3	2	2	2	0	1	2	3	1	0
	4	3	3	3	1	0	1	2	0	1
	5	4	4	4	2	1	0	1	1	2
	4	3	3	3	1	0	1	2	0	1
	3	2	2	2	0	1	2	3	1	0
	2	1	1	1	1	2	3	4	2	1
	1	0	0	0	2	3	4	5	3	2
		Input	1	1	1	3	4	5	6	4

Globale Distanzmatrix D:

16	16	16	8	8	9	11	8	5
16	16	16	6	5	6	8	5	3
15	15	15	5	3	4	5	3	2
13	13	13	5	2	2	3	2	3
10	10	10	4	2	1	2	3	5
6	6	6	2	1	2	4	4	5
3	3	3	1	2	4	7	8	8
1	1	1	1	3	6	10	12	13
0	0	0	2	5	9	14	17	19



Distanz herkömmlich berechnet: d=13

- Distanz der beiden Zeitreihen über Dynamic Time Warping: d= 5

Bestimmung des Oberkörpergestentyps

- über Varianzen in den MVS, unterscheiden *statisch* (kleine akkumulierte Varianz) von *dynamisch* (große akkumulierte Varianz)
- Einteilung in links-/rechts-/beidseitige Ausführung (in Trainingsdaten gespeichert)

Optimierung

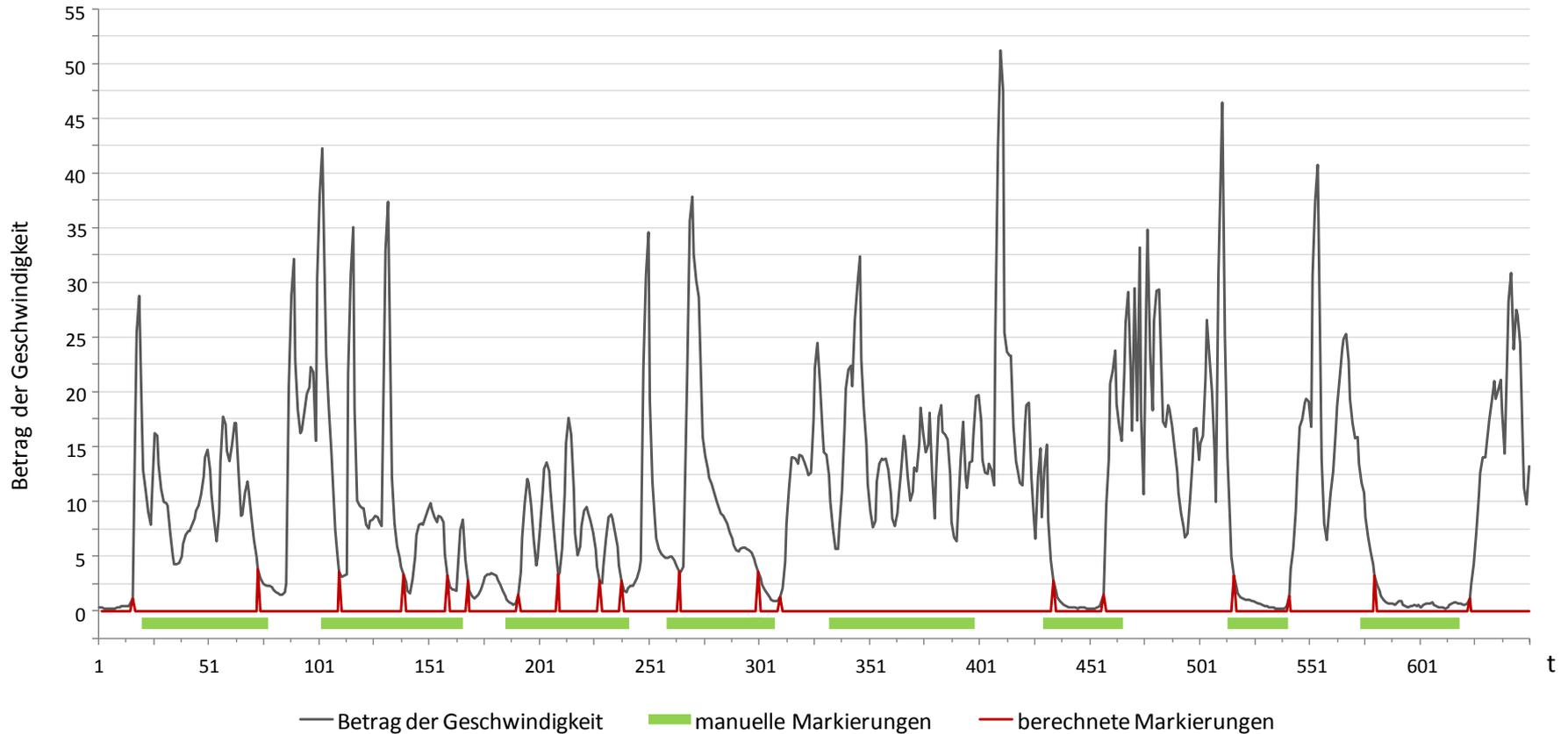
- Mindestlänge für temporäre MVS in gleitendem Suchfenster
- statisch Gesten ausgedrückt durch ihren Mittelwertsvektor → Distanz reduziert sich zu euklidischem Abstand zweier MV

Auffinden von Oberkörpergesten

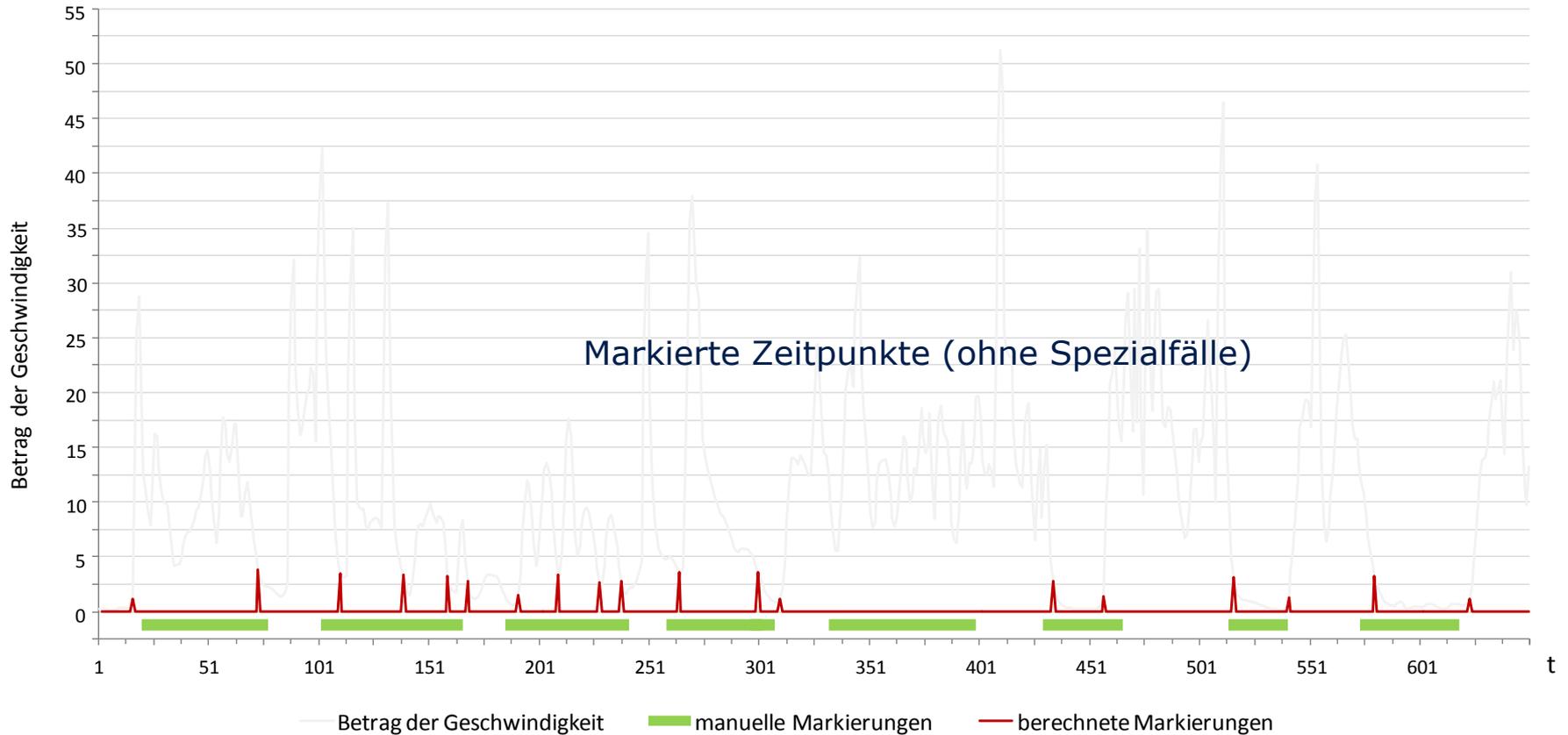
Untersuchung zur Laufzeit:

- Bewegungsausführung, *manuelles* Markieren der Gesten zur Laufzeit mit Fußtaster
- Vergleichen, ob *automatisch* markierte Zeitpunkte auf Anfang/Ende einer Geste fallen
- koordinativ schwierig, keine umfassende Untersuchung

Ergebnisse



Ergebnisse





(a)



(b)

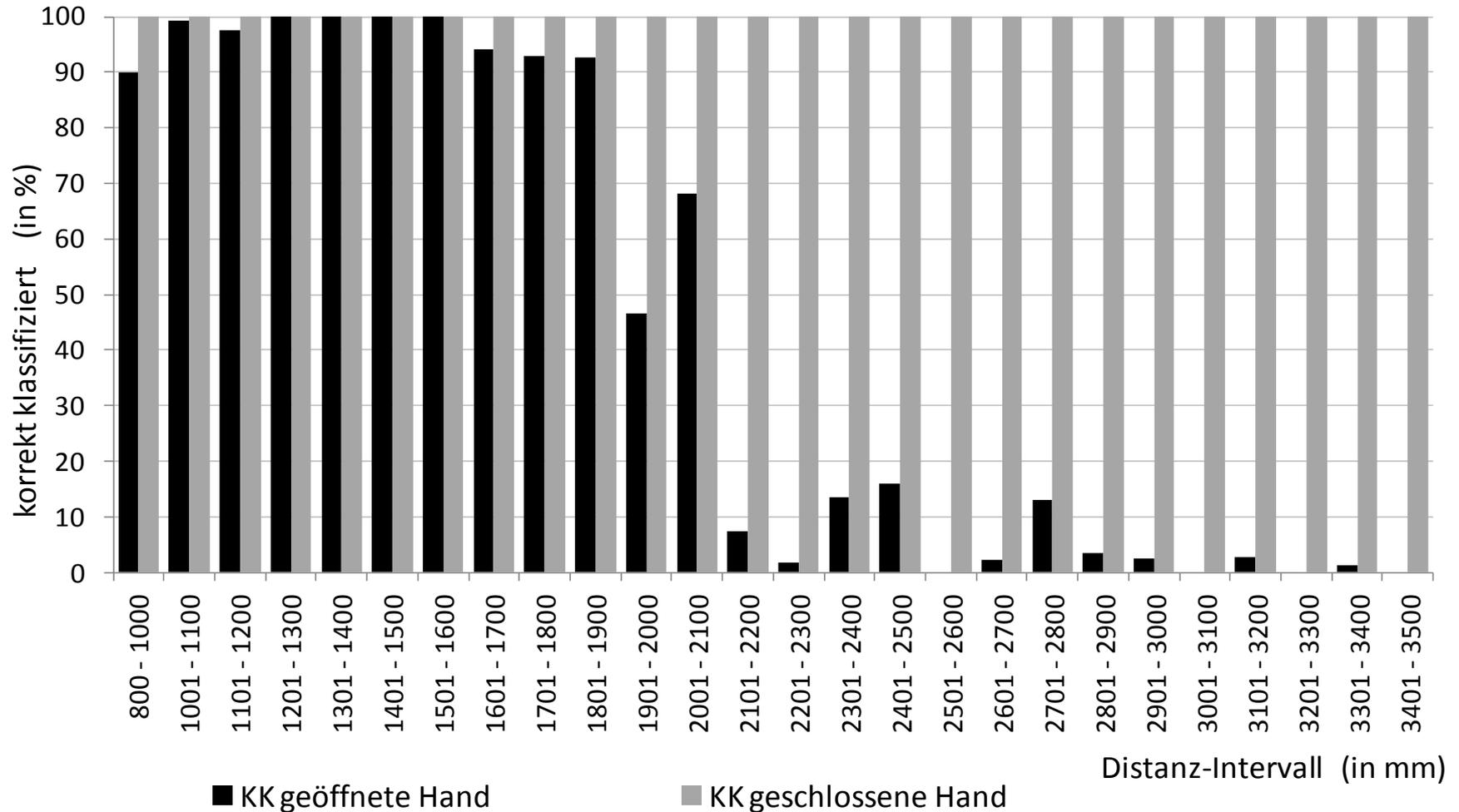


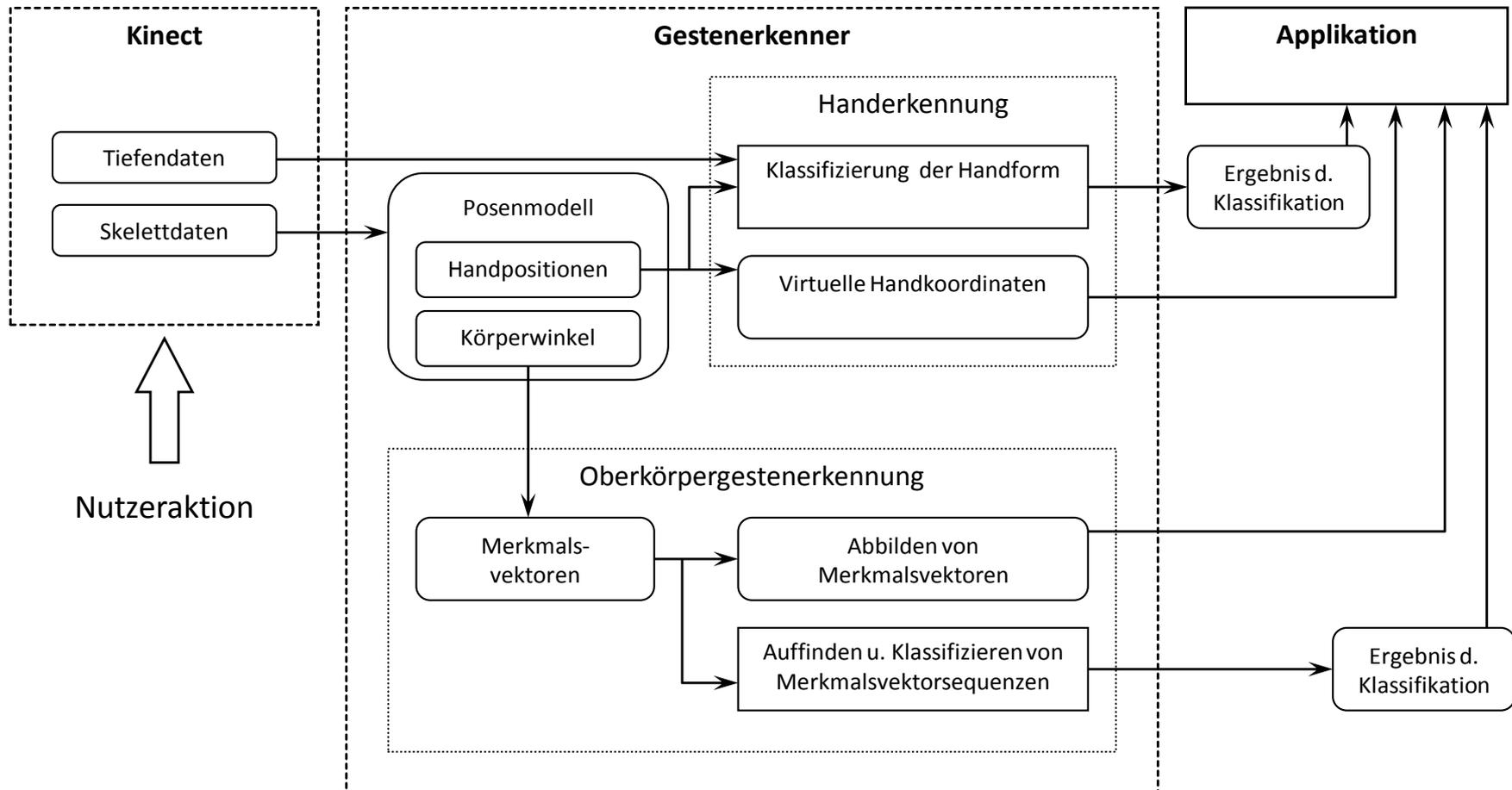
(c)

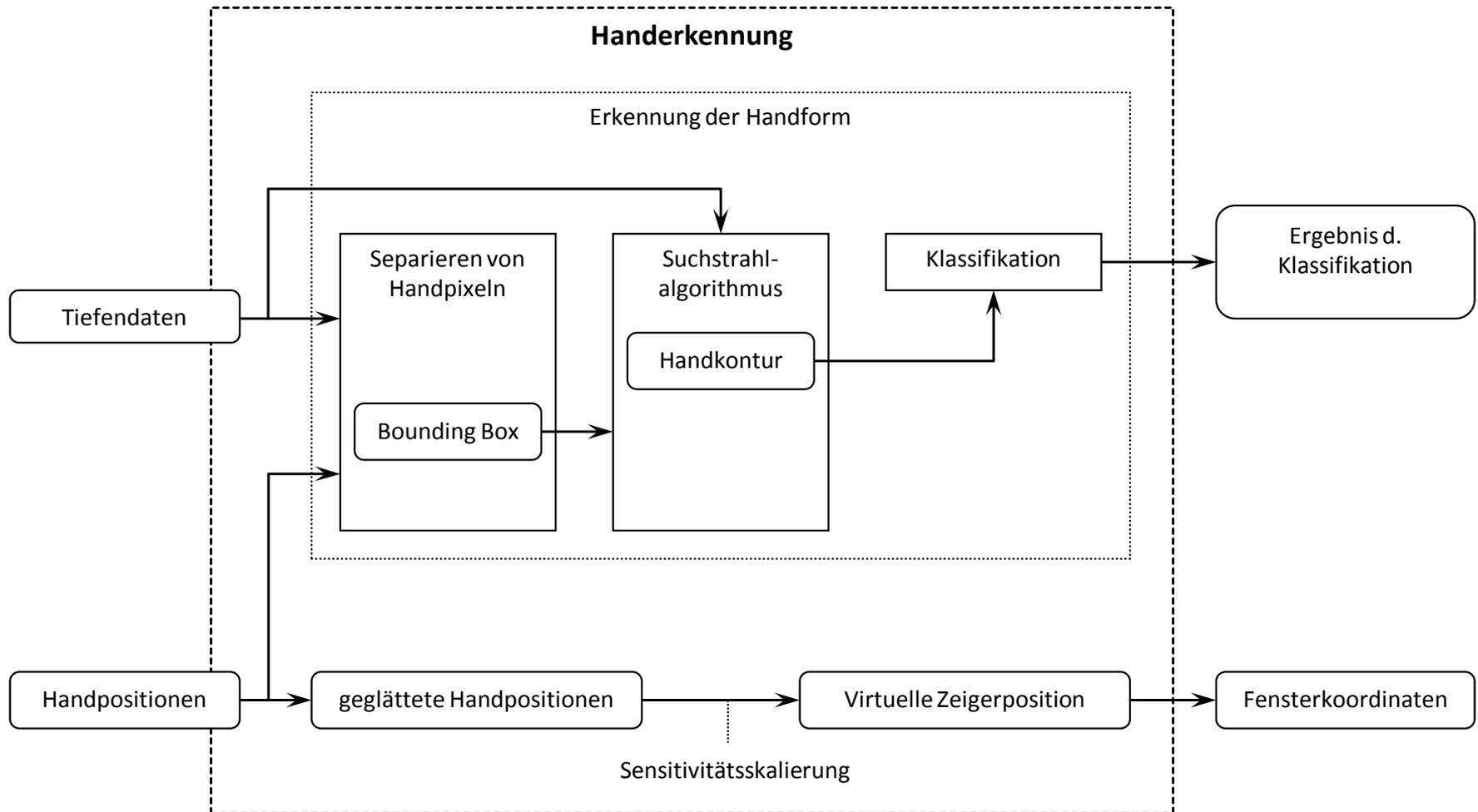


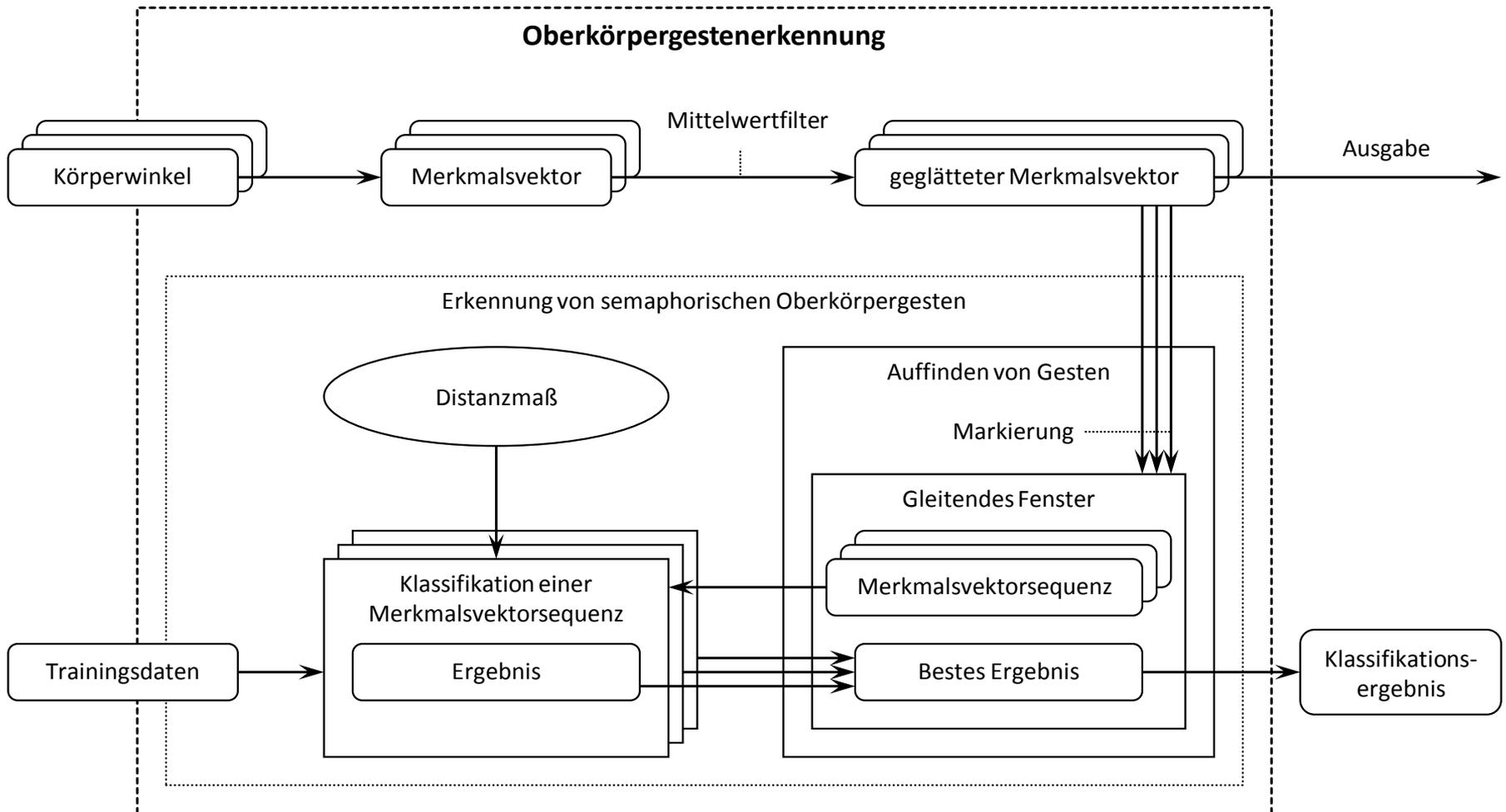
(d)

(a) ~ 1000 mm, (b) ~ 1500 mm, (c) ~ 2000 mm, (d) ~ 3000 mm









- Wahl der Gesten hat großen Einfluss
- größere Nutzerstudie fehlt

praxisnahe Herausforderungen:

- Wahl der System-Parameter
 - 13 Werte bei der Handerkennung
 - 16 Werte bei der OKG-Erkennung
- überwiegend empirisch ermittelt

- [BB11] BISWAS, K. K. ; BASU, S. K.: Gesture recognition using Microsoft Kinect. In: Automation, Robotics and Applications (ICARA), 2011 5th International Conference on, IEEE, Dezember 2011, S. 100–103
- [Boo78] DE BOOR, Carl Wilhelm R.: A Practical Guide to Splines. New York : Springer Verlag, 1978
- [BP+04] BARBIC, Jernej ; ; PAN, Jia-Yu ; FALOUTSOS, Christos ; HODGINS, Jessica K. ; POLLARD, Nancy: Segmenting Motion Capture Data into Distinct Behaviors. In: In Proceedings of Graphics Interface 2004, 2004, S. 185 – 194
- [BWK11] BATISTA, Gustavo ; WANG, Xiaoyue ; KEOGH, Eamonn J.: A Complexity-Invariant Distance Measure for Time Series. In: SDM, SIAM / Omnipress, 2011, S. 699–710
- [DSM+11] DOLIOTIS, Paul ; STEFAN, Alexandra ; MCMURROUGH, Christopher ; ECKHARD, David; ATHITSOS, Vassilis: Comparing gesture recognition accuracy using color and depth information. In: Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments. New York, NY, USA : ACM, 2011 (PETRA '11), S.20:1–20:7
- [GGNZ06] GUYON, Isabelle ; GUNN, Steve ; NIKRAVESH, Masoud ; ZADEH, Lotfi A.: Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing). Bd. 207. 1. Springer, August 2006
- [HS03] HÄRDLE, Wolfgang ; SIMAR, Léopold: Applied multivariate statistical analysis. Berlin : Springer, 2003
- [Int96] INTERNATIONAL STANDARD OF ORGANISATION. Basic human body measurements for technological design - ISO 7250. 1996
- [Ize08] IZENMAN, Alan J.: Modern Multivariate Statistical Techniques: Regression, Classification and Manifold Learning. 1. Springer Publishing Company, Incorporated, 2008
- [KGP02] KOVAR, Lucas ; GLEICHER, Michael ; PIGHIN, Frédéric: Motion graphs. In: SIGGRAPH'02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques Bd. 21. New York, NY, USA : ACM, Juli 2002, S. 473–482
- [KLJ04] KANG, Hyun ; LEE, Chang W. ; JUNG, Keechul: Recognition-based gesture spotting in video games. In: Pattern Recogn. Lett. 25 (2004), November, Nr. 15, S. 1701–1714
- [KS05] KARAM, Maria ; SCHRAEFEL, M. C.: A Taxonomy of Gestures in Human Computer Interactions/ University of Southampton. 2005. – Technical Report

- [Li12] LI, Yi: Multi-scenario gesture recognition using Kinect. In: MEHDI, Quasim H. (Hrsg.); ELMAGHRABY, Adel (Hrsg.) ; MARSHALL, Ian (Hrsg.) ; MORETON, Robert (Hrsg.) ; RAGADE, Rammohan K. (Hrsg.) ; ZAPIRAIN, Begona G. (Hrsg.) ; CHARIKER, Julia (Hrsg.); EL-SAID, Mostafa M. (Hrsg.) ; YAMPOLSKIY, Roman V. (Hrsg.) ; ZHIGIANG, Nickola L.(Hrsg.): CGAMES, IEEE, 2012, S. 126–130
- [LK99] LEE, Hyeon-Kyu ; KIM, Jin H.: An HMM-Based Threshold Model Approach for Gesture Recognition. In: IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (1999), S. 961–973
- [Mic12] Microsoft Corporation: Kinect For Windows SDK Documentation. 2012
- [OFH08] ONUMA, Kensuke ; FALOUTSOS, Christos ; HODGINS, Jessica K.: FMDistance: A fast and effective distance function for motion capture data. In: in Short Papers Proceedings of EUROGRAPHICS, 2008
- [PL06] PARK, A-Youn ; LEE, Seong-Whan: Gesture spotting in continuous whole body action sequences using discrete hidden markov models. In: Proceedings of the 6th international conference on Gesture in Human-Computer Interaction and Simulation. Berlin, Heidelberg: Springer-Verlag, 2006 (GW'05), S. 100–111
- [RKH11] RAPTIS, Michalis ; KIROVSKI, Darko ; HOPPE, Hugues: Real-time classification of dancegestures from skeleton animation. In: Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. New York, NY, USA : ACM, 2011 (SCA '11), S. 147–156
- [RVCB03] RAMAMOORTHY, Aditya ; VASWANI, Namrata ; CHAUDHURY, Santanu ; BANERJEE, Subhashis. Recognition of dynamic hand gestures. 2003
- [RYZ11] REN, Zhou ; YUAN, Junsong ; ZHANG, Zhengyou: Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera. In: Proceedings of the 19th ACM international conference on Multimedia. New York, NY, USA : ACM, 2011 (MM '11), S. 1093–1096
- [SC90] SAKOE, Hiroaki ; CHIBA, Seibi: Readings in speech recognition. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1990, Kapitel Dynamic programming algorithm optimization for spoken word recognition, S. 159–165
- [Sen08] SENIN, Pavel: Dynamic TimeWarping Algorithm Review / Department of Information and Computer Sciences, University of Hawaii, Honolulu, Hawaii 96822. 2008 (CSDL-08-04). – Forschungsbericht

- [SRT07] STIEFMEIER, Thomas ; ROGGEN, Daniel ; TRÖSTER, Gerhard: Gestures are strings: efficient online gesture spotting and classification using string matching. In: Proceedings of the ICST 2nd international conference on Body area networks. ICST, Brussels, Belgium : ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007 (BodyNets '07), S. 16:1–16:8
- [SSS+05] SCHÜNKE, M. ; SCHULTE, E. ; SCHUMACHER, U. ; VOLL, M. ; WESKER, K.: Prometheus, LernAtlas der Anatomie, Allgemeine Anatomie und Bewegungssystem. Stuttgart : Thieme, 2005
- [Tan11] TANG, Matthew. Recognizing Hand Gestures with Microsoft's Kinect. 2011
- [TLB12] TRINDADE, Pedro ; LOBO, Jorge ; BARRETO, Joao P.: Hand gesture recognition using color and depth images enhanced with hand angular pose data. In: Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on, 2012, S. 71 –76
- [WM97] WILSON, D. R. ; MARTINEZ, Tony R.: Improved heterogeneous distance functions. In: J. Artif. Int. Res. 6 (1997), Januar, Nr. 1, S. 1–34
- [WNDS99] WOO, Mason ; NEIDER, Jackie ; DAVIS, Tom ; SHREINER, Dave: OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.2. 3rd. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 1999
- [YPL07] YANG, Hee-Deok ; PARK, A-Yeon ; LEE, Seong-Whan: Gesture Spotting and Recognition for Human-Robot Interaction. In: IEEE Transactions on Robotics 23 (2007), Nr. 2, S. 256–270