

Kategorisierung von Kinect-Bewegungsdaten mittels deterministischer Algorithmen

Bachelorarbeit von
Laurenz Fuchs
1196307

Draft vom 8. März 2022

Universität der Bundeswehr München
Fakultät für Informatik

Kategorisierung von Kinect-Bewegungsdaten mittels deterministischer Algorithmen

Bachelorarbeit von
Laurenz Fuchs
1196307

Erstprüfer: Prof. Dr. Michael Koch
Zweitprüfer: Prof. Dr. Gunnar Teege
Betreuer: M. Sc. Julian Fietkau

Abgabetermin: 31. Mai 2022

Universität der Bundeswehr München
Fakultät für Informatik

Abstract

Hier steht eine kurze Zusammenfassung der Arbeit. Sie darf nicht länger als eine Seite sein, sollte aber mindestens eine halbe Seite umfassen.

Inhaltsverzeichnis

1	Einleitung und Motivation	1
1.1	Einleitung	1
1.2	Motivation	1
1.3	Kategorien der Interaktion mit Wandbildschirmen	2
1.4	Spezifikation der Kinect	3
1.5	Struktur des vorliegenden Datensatzes	4
2	Kategorisierung von Bewegungsdaten mittels deterministischer Algorithmen	7
2.1	Problembeschreibung	7
2.2	Hierarchisches Time-Series Clustering	8
2.3	Dynamic Time Warping Algorithmus	9
2.4	Veranschaulichendes Beispiel	10
2.5	Related Work Analyse	11
2.6	Einsatz im Kontext von Kinect-Bewegungsdaten	12
3	Konzeption	13
3.1	Anforderungsanalyse	13
3.2	Programmablauf	13
3.3	Teilsysteme	13
4	Implementierung	14
4.1	Grundlagen	14
4.1.1	Entwicklungsumgebung	14
4.1.2	Programmiersprache	14
4.2	Aufbau	14
4.3	Herausforderungen	14
4.3.1	Fehler in den Daten	14
4.4	Codebeschreibung	14
4.4.1	Verwendete Konzepte	14
4.4.2	Komplexität	14
5	Evaluation	15
5.1	Auswertung des Datensatzes	15

Inhaltsverzeichnis

5.2 Übertragbarkeit	15
6 Fazit	16
Abkürzungsverzeichnis	17
Abbildungsverzeichnis	18
Literaturverzeichnis	19

1 Einleitung und Motivation

1.1 Einleitung

1.2 Motivation

Von September 2021 bis August 2024 forschen die Hochschule für angewandte Wissenschaften in Hamburg und die Universität der Bundeswehr München im Rahmen des *HoPE-Projekts* zusammen an Effekten rund um die Steigerung der Aufmerksamkeit bei der Nutzung von großen, interaktiven Wandbildschirmen, sogenannten *Ambient Displays*. Dieser Bereich weist noch einen grundsätzlichen Forschungsbedarf auf. Im Projekt wird unter anderem der *Honeypot-Effekt* erforscht (UniBw, 2021). Dieser beschreibt in der Mensch-Computer-Interaktion (HCI) wie Menschen, die mit einem System interagieren, weitere Passanten anregen die Interaktion zu beobachten oder sogar an ihr teilzuhaben (Wouters et al., 2016). Der *Honeypot-Effekt* soll bei der Nutzung von *Ambient Displays* im öffentlichen und halb-öffentlichen Raum in Langzeit-Feldstudien analysiert werden. Dabei soll auch der Aspekt der Datenerhebung und -analyse weiter ausgebaut werden. Hierzu muss ein methodisches Rahmenwerk entwickelt werden, welches eine auf Sensordaten-basierende, automatische und zeitlich uneingeschränkte Evaluation von *Ambient Displays* ermöglicht (UniBw, 2021). Die Sensordaten werden von Body-Tracking-Kameras bereitgestellt. Konkret wurden die Wandbildschirme im vorliegenden Datensatz mit Microsoft Kinect v2 Kameras ausgestattet. Diese zeichnen die Interaktion von Nutzern mit den Displays auf, wodurch das Nutzerverhalten zu einem späteren Zeitpunkt ausgewertet werden kann. Die Daten liegen als Time-Series Data (TSD) vor. Bei diesem Datentyp handelt es sich um geordnete Sequenzen von Datenpunkten, die über eine gewisse Zeit hinweg aufgenommen werden (Ali et al., 2019). TSD enthalten oft wichtige Informationen die durch Mustererkennung entdeckt werden können. Eine gängige Methode zur Analyse sind hierbei *Clustering*-Verfahren (Aghabozorgi et al., 2015). Lässt sich eine Menge von Kategorien identifizieren, die etwas über das Verhalten von Menschen vor Wandbildschirmen aussagt? Ist die Einteilung in diese Kategorien automatisierbar? Aufgrund der Größe des vorliegenden Datensatzes ist eine manuelle Evaluation zur Beantwortung der Fragen nicht möglich. In dieser Bachelorarbeit wird daher versucht, dieses Problem mithilfe von deterministischen Algorithmen zu lösen. Wesentliches Ziel ist die Implementierung eines Systems zur Kategorisierung der vorliegenden Kinect-Bewegungsdaten. Dieses System wird detailliert beschrieben und anschließend evaluiert. Dabei soll *hierar-*

chisches Clustering mithilfe des *Dynamic Time Warping (DTW)*-Algorithmus eingesetzt werden. Außerdem wird die Frage beantwortet, welches Vorwissen benötigt wird, um das Verfahren einsetzen zu können und welche Attribute zur Analyse interessant sind (z.B. Laufpfade oder Engaged-Werte).

1.3 Kategorien der Interaktion mit Wandbildschirmen

Interaktive digitale Medien sind in der Öffentlichkeit immer präsenter. Deshalb wird es für Wandbildschirme immer schwieriger die Aufmerksamkeit von Passanten zu erregen und sie zur Interaktion zu animieren. Diese Herausforderungen können nicht einfach durch verbesserte Hardware oder attraktivere Displays gelöst werden. Stattdessen muss ein besseres Verständnis von Menschen und deren Technologienutzung geschaffen werden (Wouters et al., 2016). *Ambient Displays* sind große, interaktive Bildschirme im (halb-) öffentlichen Raum, mit denen Nutzer interagieren können. Es handelt sich meist um ästhetisch ansprechende Displays die Personen mit Informationen versorgen (Mankoff et al., 2003). Eine Kategorisierung der Interaktionen von Personen mit solchen Wandbildschirmen kann das Verständnis des Verhaltens verbessern. Dazu existieren verschiedenste *Audience Behaviour-Interaktionsmodelle*, wovon im Folgenden zwei näher beschrieben werden.

Das *Audience Funnel Modell* beschreibt, wie Menschen sich um ein großes öffentliches Display versammeln und von Beobachtern zu Interagierenden mit dem System, und anschließend wieder zu Beobachtern werden. Menschen neigen dazu verschiedene Phasen der Interaktivität zu durchlaufen, bevor sie direkt mit dem System interagieren (Wouters et al., 2016; Mai and Hußmann, 2018). Die einzelnen Phasen des *Audience Funnel* werden in Abbildung 1.1 gezeigt. Eine der Aufgaben eines Wandbildschirms ist es also Aufmerksamkeit auf sich zu ziehen und den Nutzer zu motivieren mit dem System zu interagieren (Mai and Hußmann, 2018). Mai and Hußmann (2018) verweisen darauf, dass *Ambient Displays* in der Öffentlichkeit nicht unbedingt der zentrale Punkt der Aufmerksamkeit sind, da vorbeigehende Personen eigene intrinsische Ziele verfolgen. Die Herausforderung für Entwickler ist es die Systeme so zu gestalten, dass sie Aufmerksamkeit erregen, sich aber gleichzeitig nicht gezwungen in den Mittelpunkt stellen.

Ein zweites Modell wird durch den bereits erwähnten *Honeypot-Effekt* beschrieben. Er zeigt, dass Individuen unabhängig von Belohnungen, Bestrafungen oder sozialem Wettbewerb von der reinen Präsenz oder den Aktivitäten anderer beeinflusst werden. In der HCI wird dies meist erkennbar, indem Passanten sich einem System nähern und überlegen, ob sie mit ihm interagieren sollen, nachdem sie anderen Menschen dabei zugesehen haben (Wouters et al., 2016).

Eine Einordnung der Bewegungsdaten in die Kategorien solcher Modelle und eine weiterführende Analyse der Bewegungen kann Aufschluss über das Verhalten von Menschen

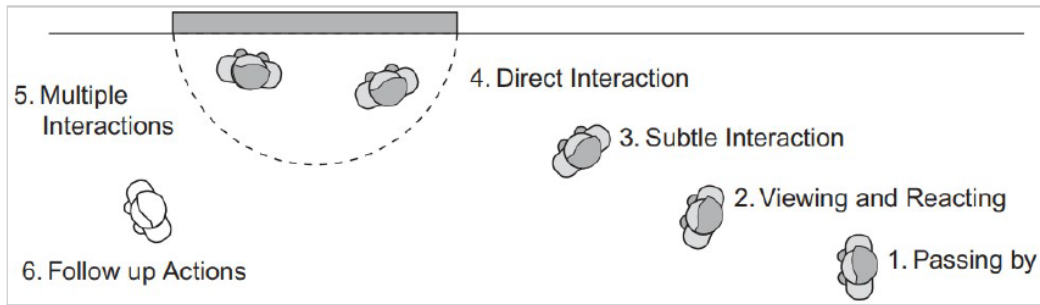


Abbildung 1.1: Audience Funnel Framework. Abbildung aus Mai and Hußmann (2018).

vor interaktiven Bildschirmen geben. Wie bereits erwähnt ist eine Kategorisierung des vorliegenden Kinect-Datensatzes aufgrund der großen Datenmenge nicht manuell realisierbar. Im Folgenden werden die Struktur des Datensatzes und die Grundlagen der verwendeten Sensorik beschrieben. Aufbauend darauf werden Überlegungen angestellt, wie eine Implementierung zur Automatisierung der Kategorisierung aussehen kann.

1.4 Spezifikation der Kinect

Der Xbox 360 Kinect Sensor war eine Revolution im Bereich der erschwinglichen 3D-Erkennungssensorik. Ursprünglich war er für die Videospiele-Industrie gedacht. Schon bald wurde er aber auch für wissenschaftliche Experimente genutzt. In späteren Jahren folgten weitere Iterationen der Kinect (Tölgyessy et al., 2021). Im vorliegenden Datensatz des *HoPE-Projekts* kam die Kinect v2 für Xbox One zum Einsatz. Diese Sensorik stellt Farbbilder einer Rot-Grün-Blau (RGB) Kamera, Tiefenbilder einer Tiefenkamera und Audiodateien verschiedener Mikrofone zur Verfügung (Microsoft, 2014). Besonders die Tiefenkamera hilft zuverlässige Ergebnisse bei der Erkennung von Menschen vor *Ambient Displays* zu erzielen. Li et al. (2014) fassen es wie folgt zusammen. Die kompakte Größe, die Benutzungsfreundlichkeit, die stark vereinfachte Hintergrund-Subtraktion im Vergleich zu anderer Sensorik, sowie die hohe Genauigkeit und die hohe Bildrate machen Tiefenkameras zu einer attraktiven Lösung für ein breites Spektrum an Anwendungen. Die Kinect v2 verwendet dabei den Ansatz der kontinuierlichen Wellenintensitätsmodulation, der häufig bei Time-of-Flight (ToF)-Tiefenkameras zum Einsatz kommt. Dabei wird das Licht einer Lichtquelle von Objekten im Sichtfeld der Kamera zurückgestreut und die Phasenverzögerung zwischen dem emittierten und dem reflektierten Licht gemessen. Diese Phasendifferenz wird für jedes Pixel im Bildfeld in einen Entfernungswert umgerechnet (Tölgyessy et al., 2021). Der Sensor kann Tiefenbilder mit einer Auflösung von 512 x 424 Pixeln und gewöhnliche Farbbilder mit 1920 x 1080 Pixeln aufnehmen (Marin et al., 2019). Bei der Kinect v2 können bis zu sechs Personen erfasst werden. Dabei wird die Lage von 25 Skelettpunkten, sowie verschiedene Gesichtsattribute

erfasst (Microsoft, 2014). Abbildung 1.2 zeigt eine Übersicht dieser Punkte.

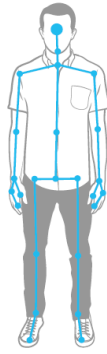


Abbildung 1.2: Skelettpunkte der Kinect v2. Abbildung aus Microsoft (2014).

1.5 Struktur des vorliegenden Datensatzes

Zur Evaluation des implementierten System dient der Kinect-Datensatz des *HoPE-Projekts*. Hierfür wurden im Jahr 2017 für 18 Wochen zwei Kinect v2 Systeme an einem *Ambient Display* angebracht. Der Versuchsaufbau kann Abbildung 1.3 entnommen werden.

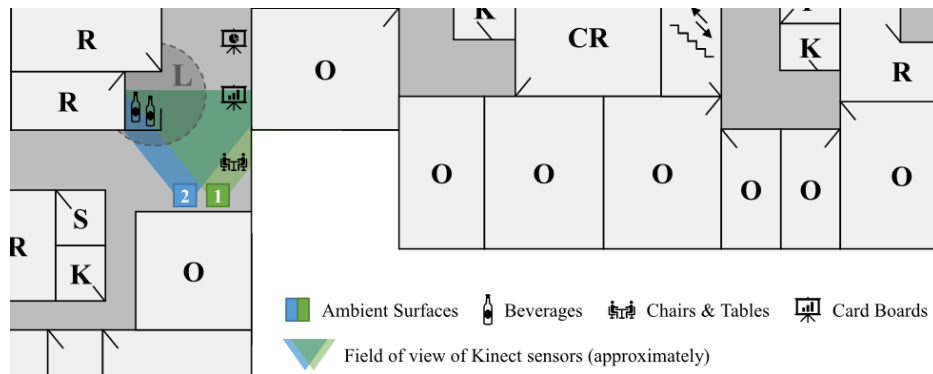


Abbildung 1.3: Kinect-Setup des Datensatzes. Abbildung von Jan Schwarzer ToDo

Eine Auswertung des Datensatzes ergab, dass dieser 97.626 Datenpunkte beinhaltet (Temiz, 2022). Jeder Datenpunkt enthält mehrere sogenannte Frames. Dabei handelt es sich um Momentaufnahmen der Kinect-Sensorik. In einem Frame befindet sich immer nur eine Person. Falls sich bei einem Datenpunkt mehrere Menschen gleichzeitig im Sensorbereich befinden, erhält jede Person eine eindeutige Identifikationsnummer und der Datenpunkt wird in mehrere Frames unterteilt. Dabei entspricht jeder Frame der Momentaufnahme einer Person. Zu jedem Datenpunkt liegen verschiedene Dateien vor, die jeweils den Zeitstempel in Kombination mit einem geeigneten Postfix als Dateinamen tragen:

- timestamp.txt
- timestamp_bodies.txt
- timestamp_summary.txt
- timestamp.xef

Die Datei *timestamp.txt* enthält folgende Attributwerte:

Attributname	Beschreibung
Zeitstempel	Zeitpunkt der Aufnahme
KinectId	(Frage an Julian und Jan: Was genau ist die KinectId?)
RecordId	Identifikationsnummer des Records
BodyIndex	Identifikationsnummer der Person
BodyCount	Anzahl der erfassten Personen
Happy	Person ist glücklich
Engaged	Person zeigt Interesse
WearingGlasses	Person trägt eine Brille
LeftEyeClosed	Person hat das linke Auge geschlossen
RightEyeClosed	Person hat das rechte Auge geschlossen
MouthOpen	Person hat den Mund geöffnet
MouthMoved	Person bewegt den Mund
LookingAway	Person schaut nicht zum Kinect-Sensor
Body.HandLeftState	Zustand der linken Hand
Body.HandRightState	Zusand der rechten Hand
X	x-Koordinate des Skelettpunkts SpineShoulder
Y	y-Koordinate des Skelettpunkts SpineShoulder
Z	z-Koordinate des Skelettpunkts SpineShoulder
Distance	Distanz zwischen Kinect-Sensor und Skelettpunkt SpineShoulder

Diese Attribute werden in der Textdatei durch drei Rauten ('###') voneinander abgegrenzt. Ein konkreter Frame aus dem Datensatz kann Abbildung 1.4 entnommen werden.

```
2017-04-10 07:41:53.943 +02:00 ### 72057594038063128 ### 1341053376 ###
### 1 ### No ### Maybe ### Unknown ### No ### No ###
No ### Yes ### Maybe ### Unknown ### Closed ### 0,1074784
### 0,2451882 ### 4,18441 ### 4,19296454679429
```

Abbildung 1.4: Frame aus dem Datensatz.

Die Datei *timestamp_bodies.txt* enthält für jeden Frame die Position aller 25 Skelettpunkte. *timestamp_summary.txt* zeigt eine kurze Zusammenfassung des Datenpunkts, welche gut menschenlesbar ist. *timestamp.xef* kann genutzt werden, um den Datenpunkt in der Anwendung Kinect Studio zu visualisieren. Letzlich sind die Einträge sogenannte *TSD*. Bei diesem Datentyp handelt es sich um geordnete Sequenzen von Datenpunkten, die über eine gewisse Zeit hinweg aufgenommen werden. Oft in regelmäßigen Abständen (Ali et al., 2019). Jeder Datenpunkt beinhaltet durchschnittlich 355 Frames, wodurch eine Gesamt-Frameanzahl von 34.687.630 entsteht (Temiz, 2022). Die Größe bestätigt erneut die Notwendigkeit eines Software-Tools zur Auswertung.

2 Kategorisierung von Bewegungsdaten mittels deterministischer Algorithmen

Nach den einführenden Bemerkungen in Kapitel 1 werden nun die grundlegenden Punkte beschrieben, die bei der Umsetzung eines Kategorisierungssystems für TSD zu beachten sind. Anschließend wird das eingesetzte Vorgehen erläutert und an einem Beispiel veranschaulicht. Ein Einblick in verwandte Arbeiten (Abschnitt 2.5) vertieft das Verständnis weiter. Abschließend wird auf zu beachtende Besonderheiten des Kinect-Datensatzes eingegangen.

2.1 Problembeschreibung

Beim vorliegenden Kinect-Datensatz sind zwei wesentliche Punkte zu beachten. Zum einen handelt es sich um eine große Datenmenge. Die einzelnen Datenpunkte müssen daher automatisiert bearbeitet werden. Ziel der Verarbeitung ist es möglichst sinnvolle *Cluster* zu erhalten. Um die Gemeinsamkeiten zweier Datenpunkte zu berechnen, wird eine geeignete Vergleichsfunktion benötigt. Die Wahl dieser Funktion ist wichtig für den Erfolg des *Clusterings* (Warren Liao, 2005). Zu beachten ist zudem, dass die Bewegungsaufnahmen durch *TSD* beschrieben werden. Bei diesem Datentyp handelt es sich um geordnete Sequenzen von Datenpunkten, die über eine gewisse Zeit hinweg aufgenommen werden. Oft in regelmäßigen Abständen (Ali et al., 2019). Werden hier herkömmliche Distanzmetriken, wie die Euklidische-Distanz verwendet kann dies zu Problemen führen und die Aussagekraft des Vergleichs verringern. Es kann vorkommen, dass verschiedene Personen die gleiche Bewegung unterschiedlich schnell ausführen. Gegebenenfalls weisen die Datenreihen bei der gleichen Bewegung daher sogar eine unterschiedliche Anzahl an Frames auf. Abbildung 2.1 zeigt ein exemplarisches Szenario. Die Kurven weisen ähnliche Segmente auf. Sie sind allerdings unterschiedlich lang und die Ähnlichkeiten treten zu unterschiedlichen Zeitpunkten auf. Mit der bekannten euklidischen Distanz ist das Ergebnis hier nicht aussagekräftig und der letzte Punkt der Reihe kann gar nicht zugeordnet werden. Um dieses Problem zu lösen ist eine elastische Metrik nötig, die besser mit zeitlichen Verschiebungen umgehen kann (Aghabozorgi et al., 2015).

Das Clustering soll mithilfe von *hierarchischem Clustern* durchgeführt werden, welches in Abschnitt 2.2 beschrieben wird. Als geeignete Distanz-Metrik wird dabei *DTW* verwendet (Abschnitt 2.3). Hier ist eine Mehrfachzuordnung von Punkten möglich. Diese können so

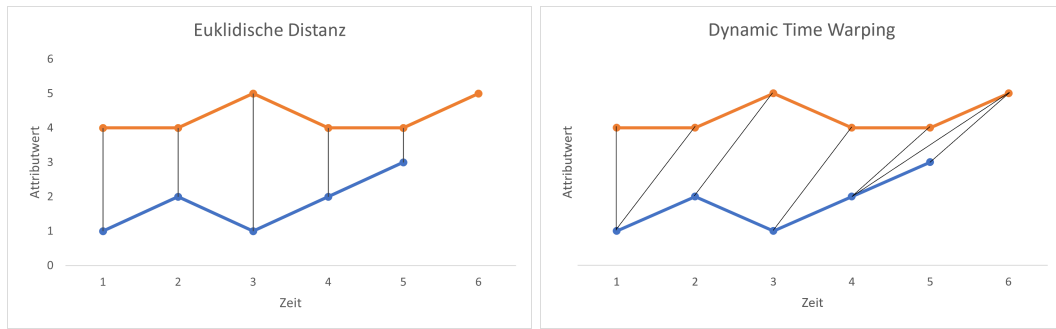


Abbildung 2.1: Zuordnung der Messpunkte zweier Datenreihen.

verbunden werden, dass die Kosten minimal sind. Dies erlaubt eine Zuordnung ähnlicher Muster in den Daten auch wenn sie zeitlich verschoben sind.

2.2 Hierarchisches Time-Series Clustering

Clustering kann genutzt werden, um große Datensätze zu kategorisieren, wenn keine Informationen zu den Kategorien vorliegen (Aghabozorgi et al., 2015). Ziel ist es, eine Struktur in den vorliegenden Daten zu finden, indem Elemente anhand eines festgelegten Vorgehens in homogene Gruppen einsortiert werden. Dabei sollen die Unterschiede bei Elementen der gleichen Gruppe möglichst gering und bei Elementen verschiedener Gruppen möglichst groß sein (Aghabozorgi et al., 2015; Warren Liao, 2005). Bei n Datenpunkten erzeugt ein *Clustering* k Partitionen des Datensatzes. Jede Partition entspricht dabei einem *Cluster*, dass mindestens ein Objekt enthält. Zum *Clustering* sind besonders die Wahl des Clustering-Algorithmus und der Methode zur Bestimmung der Ähnlichkeit von zwei Datenreihen von Bedeutung (Warren Liao, 2005). Letzteres wird DTW gelöst und in Abschnitt 2.3 beschrieben. Als Clustering-Vorgehen kann das sogenannte *hierarchical Clustering* genutzt werden. Dieses funktioniert, indem Datenobjekte in einen Baum von Clustern eingeteilt werden. Bei agglomerativen Vorgehen wird ein *bottom-up* Ansatz verfolgt. Jedes Objekt ist daher zunächst ein eigenes Cluster. Dann werden schrittweise Cluster zusammengeführt, bis alle Objekte im selben Cluster sind, oder eine Terminierungsbedingung erfüllt ist (Warren Liao, 2005).

Im Fall des Kinect-Datensatzes liegen die Daten als TSD vor. TSD zu kategorisieren findet in vielen wissenschaftlichen Gebieten Anwendung, um Muster zu finden. Sie erlauben es wichtige Informationen aus großen Datensätzen zu extrahieren. TSD sind dynamische Daten, weil sich die Attributwerte mit der Zeit verändern. Diese Sammlungen von Werten können aber auch als ein einziges Objekt betrachtet werden. Das Kategorisieren von derartigen Objekten kann bei der Detektion relevanter Muster helfen (Aghabozorgi et al., 2015). Die meisten bestehenden Cluster-Algorithmen können auf diesen Datentypen zugeschnitten

werden. So auch das hierarchische Clustering. Der größte Unterschied ist wie bereits erwähnt, dass die Distanz-/ Ähnlichkeitsmetrik angepasst werden muss, um sinnvoll bei TSD genutzt werden zu können. Ein Vorteil ist, dass hierarchisches Clustern bei der Verwendung von DTW nicht auf Datenreihen der gleichen Länge beschränkt ist (Warren Liao, 2005).

2.3 Dynamic Time Warping Algorithmus

Der *DTW Algorithmus* kann als Distanz-Metrik beim hierarchischen Clustern genutzt werden. Hier sind *one-to-many* oder *one-to-none* Beziehungen zwischen Datenpunkten möglich. Man bezeichnet das Vorgehen daher auch als eine *elastische Metrik*. Sie kann gut mit zeitlichen Verschiebungen und unterschiedlich langen Datenreihen umgehen (Aghabozorgi et al., 2015).

In der Literatur findet man viele Erläuterungen des DTW-Algorithmus. Beispielsweise in Mohammadzade et al. (2021), Warren Liao (2005), Aghabozorgi et al. (2015) oder Yu and Xiong (2019). Im Folgenden soll der Algorithmus ebenfalls erläutert werden. Das Ziel ist die Zuordnung verschiedener Signale, die im Verlauf der Zeit ähnliche Muster aufweisen. Diese Muster treten zu unterschiedlichen Zeiten oder in unterschiedlichen Raten auf. Um die Zuweisung zueinander zu erreichen muss jedes Signal gegebenenfalls lokal gestreckt oder gestaucht werden. Durch *dynamisches Programmieren* kann eine Korrespondenz zwischen den Zeitindizes zweier Signale gefunden werden, sodass die Summe der Distanzen zwischen den beiden Signalen minimal ist (Mohammadzade et al., 2021).

Seien $Q = q_1, q_2, \dots, q_i, \dots, q_n$ und $R = r_1, r_2, \dots, r_j, \dots, r_n$ zwei Signale die durch DTW ausgerichtet werden sollen, sodass die Differenz zwischen ihnen minimal ist. Dafür wird eine $n \times m$ Matrix M gebildet, wobei das Element (i, j) die Distanz $d(q_i, r_j)$ zwischen zwei Punkten q_i und r_j enthält. Hierfür wird im Normalfall die euklidische Distanz genutzt (Warren Liao, 2005). In der Zelle $(0, 0)$ enthält die Matrix den Wert 0 und für die übrigen Zellen der Spalte und Zeile 0 wird jeweils der Wert unendlich angenommen. Das Verfahren kann auf einfache Weise auf mehrdimensionale TSD angepasst werden. In diesem Fall sind die zu minimierenden Kosten die Summe der Distanzen zwischen den korrespondierenden Werten der beiden Datenreihen (Mohammadzade et al., 2021). Ein *Warping Pfad* $W = w_1, w_2, \dots, w_k, \dots, w_K$ in dem $\max(m, n) \leq K \leq m + n - 1$ gilt, ist eine Reihe von Matrixelementen die drei Bedingungen erfüllen:

- Grenzbedingungen
- Kontinuität
- Monotonität

Die Grenzbedingungen fordern, dass die Startzelle des Warping Pfads in der Matrix eine andere ist als die Zielzelle. Dabei gilt: $w_1 = (m, n)$ und $w_k = (0, 0)$. Aufgrund der Kontinuität

sind nur Schritte in angrenzende Zellen erlaubt. Die Monotonitätsbedingung zwingt den Warping Pfad dazu, sich monoton in der Zeit zu bewegen.

Mit Hilfe von dynamischer Programmierung können die Matrixeinträge von $M_{(0,0)}$ ausgehend berechnet werden, indem wiederholt die folgende Gleichung ausgewertet wird. $M_{(i,j)} = d(q_i, r_j) + \min \{M_{(i-1,j-1)}, M_{(i-1,j)}, M_{(i,j-1)}\}$ Sie definiert den kumulativen Abstand als die Summe aus dem Abstand des aktuellen Elementes und dem Minimum der kumulativen Abstände der benachbarten Elemente (Warren Liao, 2005). Der Pfad mit der minimalen Distanz ist von Bedeutung. Er kann mit $d_{DTW} = \min(\frac{\sum_{k=1}^K w_k}{K})$ berechnet werden.

2.4 Veranschaulichendes Beispiel

Zur Veranschaulichung soll nun das Vorgehen anhand von vier Datenreihen durchgeführt werden. Seien $R_1 = (0, 1, 1, 2, 2, 1, 1)$, $R_2 = (0, 1, 1, 1, 2, 2, 1, 1)$, $R_3 = (1, 1, 0, 1, 2, 2, 1, 1)$ und $R_4 = (2, 2, 1, 2, 2, 2)$ gegeben. Die Kosten können berechnet werden, indem die Kostenmatrix aufgestellt und der Warping Pfad berechnet wird (Abschnitt 2.3). Für die Berechnung der Kosten zwischen R_1 und R_2 ergibt sich beispielsweise folgende Matrix mit hervorgehobenem Warping Pfad.

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

Außerdem ergeben sich folgende gerundete Kosten zwischen den einzelnen Reihen:

Reihen	Kosten
R_1, R_2	0
R_1, R_3	1.56
R_1, R_4	2.88
R_2, R_3	2.89
R_3, R_4	2.67

Da R_1 und R_2 zusammen die geringsten Kosten haben bilden sie das erste Cluster C_1 . Um dessen Wertreihe zu erhalten werden die Attributwerte paarweise addiert und die Summe anschließend halbiert. Dabei kann anhand des Warping Pfads abgelesen werden, welche Werte zusammengehören. So ergibt sich $C_1 = (0, 1, 1, 1, 2, 2, 1, 1)$. Es sollte ein *Threshold*

definiert werden, der angibt, ab wann die Kosten zu groß sind, um die beiden Datenreihen zu einem Cluster zusammenzuführen. Ein sinnvoller Wert hängt dabei vom vorliegenden Datensatz ab. Sei dieser Threshold im Folgenden $t = 2$. Nun werden die Schritte wiederholt, wobei C_1 als eine einzelne neue Datenreihe betrachtet wird. Dabei ergeben sich folgende Kosten:

Reihen	Kosten
C_1, R_3	1.56
C_1, R_4	2.89
R_3, R_4	2.67

Nun haben C_1 und R_3 die geringsten Kombinationskosten. Da die Kosten geringer sind als der Threshold kann ein neues Cluster gebildet werden. Es ergibt sich $C_2 = (0.5, 1, 0.5, 1, 2, 2, 1, 1)$. Damit können die neuen Kosten berechnet werden:

Reihen	Kosten
C_2, R_4	2.78

Da die berechneten Kosten den zuvor definierten Threshold übersteigen wird kein neues Cluster mehr gebildet und wir sind am Ende unserer Berechnungen angekommen. Abbildung 2.2 veranschaulicht die gebildeten Cluster.

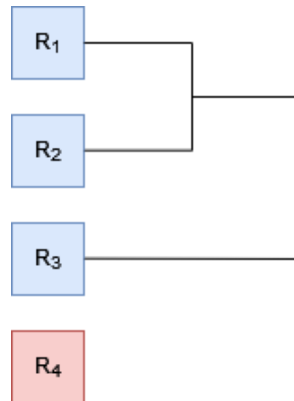


Abbildung 2.2: Graphische Veranschaulichung der gebildeten Cluster.

2.5 Related Work Analyse

ToDo

2.6 Einsatz im Kontext von Kinect-Bewegungsdaten

Wie bereits erwähnt handelt es sich bei den vorliegenden Daten um TSD. Warren Liao (2005) und Aghabozorgi et al. (2015) weisen darauf hin, dass besonders die verwendete Distanz-/Ähnlichkeitsmetrik entscheidend ist, um diesen Datentyp sinnvoll in Kategorien einteilen zu können. Hierarchisches Clustering in Kombination mit DTW scheint ein vielversprechender Ansatz zu sein. Da die Daten in Textdateien vorliegen ist eine Komponente im System nötig, die diese Informationen ausliest und in geeigneten Objekten der genutzten Programmiersprache abspeichert. Besonders von Bedeutung ist zudem die Wahl geeigneter Attribute. Die Daten enthalten viele Werte die mehr oder weniger interessant für die Kategorisierung sein können. Die Laufwege oder Engagement-Werte sind beispielsweise relevanter als die Tatsache, ob eine Person Brillenträger ist oder nicht. Idealerweise sollte die Wahl der genutzten Attribute vom Nutzer der Anwendung definiert werden können. So kann das Tool effektiv genutzt werden, um nach wiederkehrenden Abläufen in den Daten zu suchen. Zudem ist die große Menge der Daten zu erwähnen. Aghabozorgi et al. (2015) weist darauf hin, dass DTW weniger performant ist als herkömmliche Metriken wie die euklidische Distanz. Gegebenenfalls müssen die Daten vor der Nutzung geeignet gefilter werden. Diese Filterung soll allerdings nicht Bestandteil dieser Bachelorarbeit sein.

3 Konzeption

3.1 Anforderungsanalyse

3.2 Programmablauf

3.3 Teilsysteme

4 Implementierung

4.1 Grundlagen

4.1.1 Entwicklungsumgebung

4.1.2 Programmiersprache

4.2 Aufbau

4.3 Herausforderungen

4.3.1 Fehler in den Daten

4.4 Codebeschreibung

4.4.1 Verwendete Konzepte

4.4.2 Komplexität

5 Evaluation

5.1 Auswertung des Datensatzes

5.2 Übertragbarkeit

6 Fazit

Abkürzungsverzeichnis

HCI	Mensch-Computer-Interaktion
RGB	Rot-Grün-Blau
ToF	Time-of-Flight
DTW	Dynamich Time Warping
TSD	Time-Series Data

Abbildungsverzeichnis

1.1	Audience Funnel Framework. Abbildung aus Mai and Hußmann (2018). . . .	3
1.2	Skelettpunkte der Kinect v2. Abbildung aus Microsoft (2014).	4
1.3	Kinect-Setup des Datensatzes. Abbildung von Jan Schwarzer ToDo	4
1.4	Frame aus dem Datensatz.	5
2.1	Zuordnung der Messpunkte zweier Datenreihen.	8
2.2	Graphische Veranschaulichung der gebildeten Cluster.	11

Literaturverzeichnis

- Aghabozorgi, S., Seyed Shirkhorshidi, A., and Ying Wah, T. (2015). Time-series clustering – a decade review. *Information Systems*, pages 16–38.
- Ali, M., Alqahtani, A., Jones, M. W., and Xie, X. (2019). Clustering and classification for time series data in visual analytics: A survey. *IEEE Access*, pages 181314–181338. Conference Name: IEEE Access.
- Li, L. et al. (2014). Time-of-flight camera—an introduction. *Technical white paper*. SLOA190B.
- Mai, C. and Hußmann, H. (2018). The audience funnel for head-mounted displays in public environments. In *2018 IEEE 4th Workshop on Everyday Virtual Reality (WEVR)*, page 5.
- Mankoff, J., Dey, A. K., Hsieh, G., Kientz, J., Lederer, S., and Ames, M. (2003). Heuristic evaluation of ambient displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 169–176.
- Marin, G., Agresti, G., Minto, L., and Zanuttigh, P. (2019). A multi-camera dataset for depth estimation in an indoor scenario. *Data in Brief*, page 104619.
- Microsoft (2014). Human interface guidelines v2.0.
- Mohammadzade, H., Hosseini, S., Rezaei-Dastjerdehei, M. R., and Tabejamaat, M. (2021). Dynamic time warping-based features with class-specific joint importance maps for action recognition using kinect depth sensor. *IEEE Sensors Journal*, pages 9300–9313. Conference Name: IEEE Sensors Journal.
- Temiz, J. (2022). Konzeption und implementierung eines datenanalyse-werkzeugs für body-tracking-kameras. Bachelor’s thesis, Universität der Bundeswehr München.
- Tölgyessy, M., Dekan, M., Chovanec, L., and Hubinský, P. (2021). Evaluation of the azure kinect and its comparison to kinect v1 and kinect v2. *Sensors*, page 413.
- UniBw (2021). Honeypot-effekt an interaktiven ambient displays (HoPE) — inf2.
- Warren Liao, T. (2005). Clustering of time series data—a survey. *Pattern Recognition*, pages 1857–1874.

- Wouters, N., Downs, J., Harrop, M., Cox, T., Oliveira, E., Webber, S., Vetere, F., and Vande Moere, A. (2016). Uncovering the honeypot effect: How audiences engage with public interactive systems. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, pages 5–16.
- Yu, X. and Xiong, S. (2019). A dynamic time warping based algorithm to evaluate kinect-enabled home-based physical rehabilitation exercises for older people. *Sensors*, page 2882.

Hiermit versichere ich, die vorliegende Arbeit selbständig und ohne fremde Hilfe verfasst, die Zitate ordnungsgemäß gekennzeichnet und keine anderen, als die im Literatur/Schriftenverzeichnis angegebenen Quellen und Hilfsmittel benutzt zu haben.

Ferner habe ich vom Merkblatt über die Verwendung von studentischen Abschlussarbeiten Kenntnis genommen und räume das einfache Nutzungsrecht an meiner Bachelorarbeit der Universität der Bundeswehr München ein.

München, den 31. Mai 2022

.....
(*Unterschrift*)