Gottfried Wilhelm
Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Institut für Praktische Informatik
Fachgebiet Software Engineering

Erkennung von Bewegungsmustern in Trajektorien zur Auswahl mobiler Dienste

Bachelorarbeit

im Studiengang Informatik

von

Raphael Voges

Prüfer: Prof. Dr. Joel Greenyer
Zweitprüfer: Prof. Dr.-Ing. habil. Monika Sester
Betreuer: Dipl.-Inform. Stefan Gärtner, M.Sc. Udo Feuerhake

Hannover, 28.08.2014

Abstract

Smartphones are an integral part of our society. However, many mobile users have difficulty finding the requested application, caused by the sheer amount of different applications they tend to install. To overcome this issue, mobile services are introduced that select the desired application for users based on their current context.

In this thesis, an algorithm is developed that supports mobile services by providing additional contextual information extracted from movement patterns. To derive the needs for these movement patterns, it is first shown how mobile services benefit from contextual information; secondly the mining of movement patterns from a huge database and how it can be used by software engineers during mobile service deployment. Furthermore, it is demonstrated how mobile services can be linked to ubiquitous computer systems by applying varying exploration kinds. The thesis concludes with considerations on the source data coverage of the developed algorithm and its sufficiency for mobile services in particular.

In this respect, it could be ascertained that about 60% of soccer data were covered by movement patterns after running the algorithm. Moreover, an intention was assigned to 56% of movement patterns, which resulted in achieving a gain of information for mobile services.

Zusammenfassung

Smartphones sind aus unserem Alltag kaum noch wegzudenken. Durch die Vielzahl der Anwendung wird der Nutzer jedoch vor das Problem gestellt, dass die gesuchte Anwendung schwer zu finden ist. Hier sollen mobile Dienste Abhilfe schaffen, indem sie basierend auf Kontextinformationen automatisch den richtigen Dienst für den Nutzer auswählen.

In dieser Arbeit wurde ein Verfahren entwickelt, das zusätzliche Kontextinformationen in Form von Bewegungsmustern sammelt. Dazu wird zunächst dargestellt, wie ein mobiler Dienst von diesen Informationen profitieren kann. Zudem wird gezeigt, wie mobile Dienste mit ubiquitären Computersystemen durch verschiedene Explorationsarten verknüpft werden können. Anschließend wird vorgestellt, wie das Verfahren Bewegungsmuster aus aufgenommenen Bewegungsdaten extrahieren kann, um Software-Entwickler bei der Bereitstellung von mobilen Diensten zu unterstützen. Die Arbeit schließt mit einer Diskussion, was für eine Abdeckung des Datensatzes durch das Verfahren erreicht werden kann und ob diese mobilen Diensten genügt.

Dabei wurde festgestellt, dass ca. 60% der Fußballdaten durch das Verfahren mit Bewegungsmustern abgedeckt wurden. Zudem konnte 56% der gefundenen Muster eine Intention zugeordnet werden, sodass ein Informationsgewinn für mobile Dienste erzielt wurde.

Inhaltsverzeichnis

1. Einleitung **1**
1.1. Motivation und Problemstellung . . . 1
1.2. Ziel der Arbeit . . . 3
1.3. Struktur der Arbeit . . . 3

2. Grundlagen **4**
2.1. Kontext . . . 4
2.2. Mobile Dienste . . . 5
2.3. Trajektorie-Analyse . . . 6

3. Verwandte Arbeiten **9**

4. Mobile Dienste **12**
4.1. Zusammenhang zwischen Intention und Bewegung . . . 12
4.2. Explorationsarten . . . 13
4.3. Zwei Ansätze zur Bestimmung des Kontextes . . . 16

5. Trajektorie-Analyse **18**
5.1. Voraussetzungen und Überblick . . . 18
5.2. Segmentierung . . . 20
5.2.1. Geschwindigkeit . . . 20
5.2.2. Bewegungsrichtung . . . 22
5.3. Sammeln von ähnlichen Segmenten . . . 24
5.4. Aufbereitung der Muster . . . 28

6. Evaluation und Diskussion **30**
6.1. Ziel der Evaluation . . . 30
6.2. Vorbereitung und Durchführung . . . 31
6.3. Ergebnis . . . 32
6.3.1. Maximierung der Punkt-Abdeckung . . . 32
6.3.2. Maximierung der Segment-Abdeckung . . . 35
6.3.3. Parameter für das Sammeln ähnlicher Segmente . . . 37
6.3.4. Qualitative Analyse der gefundenen Muster . . . 40
6.4. Diskussion . . . 41

7. Zusammenfassung und Ausblick **43**
7.1. Zusammenfassung . . . 43
7.2. Ausblick . . . 44

A. Beispiel zur Maximierung der Punkt-Abdeckung **51**

B. Beispiel zur Maximierung der Segment-Abdeckung **55**

C. Negativbeispiel für falsche Parameterwahl **59**

D. Qualitative Auswertung der gefundenen Muster **62**

E. Inhalt der CD **64**

1. Einleitung

1.1. Motivation und Problemstellung

Heutzutage ist ein Smartphone aus dem Leben vieler Menschen kaum noch wegzudenken. Wie die Statistik in Abbildung 1.1 belegt, ist die Anzahl der Smartphone-Nutzer in Deutschland seit 2009 stark gestiegen und hat sich mehr als versechsfacht. Dadurch, dass fast jeder Nutzer zusätzlich eine mobile Datenflatrate besitzt, können permanent auch unterwegs zahlreiche Informationen vom Handy abgerufen werden. Hinzu kommt, dass die Entwicklung der Smartphones immer weiter voranschreitet und moderne Exemplare mittlerweile kleine Alleskönner für den Alltag sind. Aufgrund dessen hat sich die Lebensqualität vieler Nutzer verbessert, da mit einem Smartphone viele mobile Informationsdienste genutzt werden können, beispielsweise die Navigation in fremden Städten.

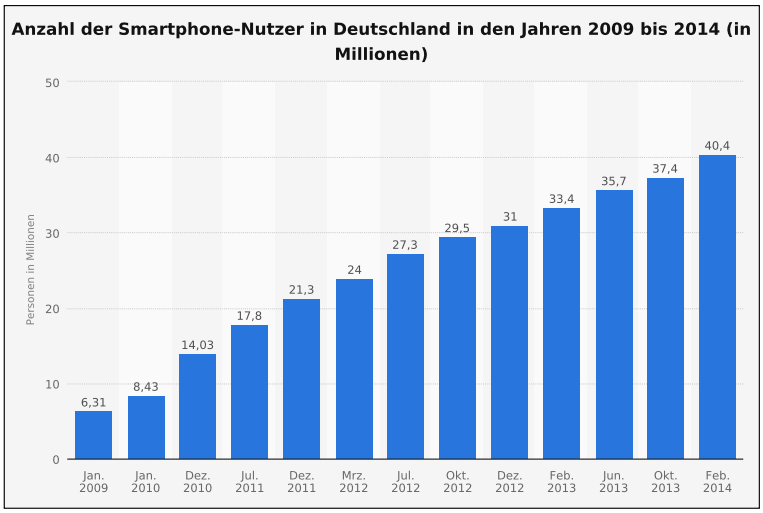

Abbildung 1.1.: Anzahl der Smartphone-Nutzer in Deutschland[1]

Dies stellt jedoch nur einen kleinen Teil der Möglichkeiten eines Smartphones dar. Deren weitreichenden Funktionalitäten machen die Entwicklung von Apps für jedermann interessant - vor allem jedoch für Unternehmen, die ihre Kunden mit Informationen versorgen oder Geld mit der Entwicklung verdienen möchten. Dabei kann zwischen vielen verschiedenen Arten von Apps unterschieden werden. Eine Art von Apps umfasst diejenigen, die der Unterhaltung dienen. Hierzu zählen zum Beispiel Spiele oder Musik-Player. Eine andere Art von Apps sind solche, mit denen Informationen abgerufen und andere Services für den Alltag verwendet werden können, wie beispielsweise

Anwendungen, die Stadtkarten bereitstellen und eine optimale Route für den Nutzer berechnen können.

Da viele Unternehmen ihre eigenen Anwendungen entwickeln, ist die Anzahl an mobilen Apps enorm gestiegen, wie eine weitere Statistik in Abbildung 1.2 belegt. Hierbei

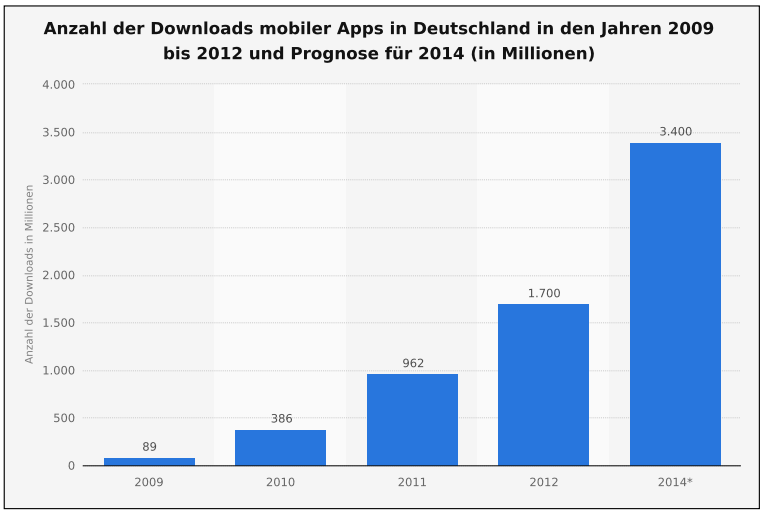

Abbildung 1.2.: Anzahl der Downloads mobiler Apps in Deutschland[2]

ergibt sich nun für den Nutzer das Problem, dass er eine große Zahl verschiedenster Anwendungen auf seinem Smartphone installieren muss, um an alle gewünschten Informationen und Funktionen zu gelangen. Unter dieser großen Zahl leidet jedoch die Übersichtlichkeit auf dem Smartphone, wodurch es dem Nutzer erschwert wird, auf Anhieb die gewünschte App zu finden. Um diesem Problem entgegenzuwirken, existieren mobile Dienste, die sich zum Ziel gesetzt haben, möglichst viele Anwendungen in sich zu vereinen. Ziel dieser Arbeit ist, diese mobilen Dienste dahingehend zu optimieren, dass die Auswahl des richtigen Diensts automatisch erfolgt. Hierfür werden Bewegungsmuster als Kontext eingeführt.

Zu diesem Thema wurde 2012 eine Bachelorarbeit von Markus Hettig mit dem Titel „Grafisches Werkzeug zur Bereitstellung von kontextbezogenen Diensten für mobile Nutzer“ verfasst[3]. In deren Rahmen hat der Autor ein Tool entwickelt, das basierend auf zusätzlichen Informationen stets die vom Nutzer benötigte Anwendung zur Verfügung stellt. Diese zusätzlichen Informationen umfassen bislang die Präferenzen und den aktuellen Standort der Person.

Um den entwickelten Ansatz mit weiteren Kontextinformationen zu erweitern, müssen so viele Informationen wie möglich über den Nutzer gesammelt und ausgewertet werden, unter anderem auch dessen Bewegung und die in ihr enthaltenen Muster. Bewegungsdaten stellen eine neue Art von Kontext dar, die bisher nicht berücksichtigt wurde, der wertvoll ist, da er Aufschluss über die Intention eines Nutzers geben kann. Der Fokus dieser Arbeit liegt auf der Auswertung der Daten. Hierbei sollen Bewegungsmuster gefunden werden, die Rückschlüsse auf die Intention des beobachteten Nutzers erlauben. Dadurch wird es möglich, differenziertere Vorhersagen bezüglich

der Intentionen des Benutzers zu treffen.

1.2. Ziel der Arbeit

Das Ziel dieser Arbeit ist die Erweiterung des Kontextes, der zur Auswahl eines mobilen Dienstes genutzt werden kann, um die Komponente *Bewegungsmuster*. Hierzu soll ein Verfahren entwickelt werden, das zunächst Bewegungsmuster in einem großen Datensatz von Trajektorien erkennt. Jedem Bewegungsmuster soll eine Intention zugeordnet werden, sodass Entwickler eines mobilen Diensts einem Bewegungsmuster eine Anwendung zuweisen können. Im Anschluss soll es möglich sein, die gefundenen Bewegungsmuster während der Nutzung des mobilen Dienst wiederzuerkennen, um eine zugeordnete Anwendung aufzurufen.

Um das entwickelte Verfahren bewerten zu können, soll es zunächst prototypisch implementiert und anschließend anhand von Bewegungsdaten, die während eines Fußballspiels aufgezeichnet wurden, evaluiert werden.

1.3. Struktur der Arbeit

In Kapitel 2 werden die Grundlagen dargelegt, die für die Entwicklung der Anwendung benötigt werden. Dabei wird insbesondere auf Kontext, mobile Dienste und die Trajektorie-Analyse eingegangen.

Im Folgenden wird eine Literaturrecherche dokumentiert. Verwandte Arbeiten werden im Hinblick auf die Relevanz und Abgrenzung zu dieser Arbeit analysiert.

In Kapitel 4 wird das Konzept für die mobilen Dienste vorgestellt. Hierbei wird insbesondere auf den Zusammenhang zwischen Intention und Bewegung eingegangen und es werden verschiedene Explorationsarten und ihre Bedeutung für ubiquitäre Computersysteme dargestellt. Des Weiteren wird in diesem Kapitel ein Vergleich zwischen den Ansätzen Bottom-up und Top-down gezogen und diskutiert, welcher dieser beiden für das vorgestellte Problem besser geeignet ist.

Kapitel 5 enthält das Konzept für die Trajektorie-Analyse. Hier werden zunächst Voraussetzungen geschaffen und ein Überblick über das Verfahren gegeben. Anschließend werden die einzelnen Phasen des Verfahrens mit Hilfe von Pseudocode erklärt - unter Verdeutlichung ihrer jeweiligen Wichtigkeit.

Daraufhin folgt in Kapitel 6 eine Evaluation des vorgestellten Verfahrens und eine Diskussion. Zur Evaluation werden hierbei Bewegungsdaten von Fußballspielern genutzt. Diese sind besonders geeignet, da sie sich durch schnelle Bewegungen und häufige Geschwindigkeits- und Richtungswechsel auszeichnen.

Die Arbeit schließt mit einer Zusammenfassung und einem Ausblick, welche Voraussetzungen in der Zukunft geschaffen werden müssen, um das vorgestellte Verfahren im Alltag etablieren zu können.

2. Grundlagen

In diesem Kapitel sollen zunächst die Grundlagen erläutert werden, die nötig sind, um die folgenden Kapitel nachvollziehen zu können. Hierzu wird zunächst darauf eingegangen, wie der Kontext definiert ist und wozu er genutzt werden kann. Anschließend wird erklärt, was unter einem mobilen Dienst zu verstehen ist und welche Vorteile er gegenüber einer Ansammlung von Apps bietet. Zum Abschluss dieses Kapitels wird die Trajektorie-Analyse genauer betrachtet, wobei die für sie notwendigen Definitionen gegeben werden.

2.1. Kontext

Um zunächst eine allgemeine Definition des Kontextes zu geben, bietet sich die Arbeit von Dey an, in welcher der Kontext folgendermaßen definiert wird:

> *„Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.“[4]*

Eine äquivalente deutsche Definition findet sich in einer Arbeit von Rothermel[5].

Anhand dieser Definitionen kann nun genau beurteilt werden, welche Informationen zu dem Kontext der momentanen Situation gehören und welche keinen Einfluss auf den Kontext haben. Ein Beispiel, das dieses verdeutlicht, ist folgendes:

Eine Person läuft zu Fuß durch die Stadt und benutzt ihr Mobiltelefon, um den richtigen Weg zu finden. Vorliegende Informationen, die zum Kontext gehören könnten, sind beispielsweise die *genaue Uhrzeit* sowie die *aktuelle Position* der Person. Jedoch ist nur die aktuelle Position für den Kontext einer Navigationsanwendung relevant, da unterschieden werden muss, an welchem Ort sich der Nutzer befindet. Die genaue Uhrzeit hingegen hat in der Regel keinen Einfluss auf den Kontext, da die Navigation unabhängig von der Zeit ständig dieselbe Route anzeigt. Es gibt jedoch Ausnahmen, in denen auch die Uhrzeit zum Kontext gehört (z.B. wenn der Nahverkehr genutzt werden soll), sodass der Kontext oftmals vom Ziel des Anwendungsfalls abhängt.

Des Weiteren lassen sich zwei verschiedene Arten von Kontext unterscheiden. Auf der einen Seite der sogenannte *Primärkontext*, der Ort, Identität und Zeit umfasst. Auf der anderen Seite der *Sekundärkontext*, der alle weiteren möglichen Kontextinformationen beinhaltet. Die Situation einer Entität lässt sich hierbei bereits durch die Informationen des Primärkontext hinreichend charakterisieren, wobei diese Informationen schließlich genutzt werden können, um auch den Sekundärkontext zu erweitern. Beispielsweise

können anhand der Identität (Primärkontext) einer Person zahlreiche weitere Informationen, wie zum Beispiel die Email-Adresse (Sekundärkontext), herausgefunden werden[6].

Nachdem nun eindeutig geklärt wurde, wie der Kontext definiert ist und in welche Arten er sich aufteilt, soll nun erläutert werden, inwiefern Systeme durch die Nutzung ihres Kontexts profitieren können. Hierzu folgt zunächst abermals eine Definition für kontextbezogene Systeme durch Dey:

> *„A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.“[4]*

Eine ähnliche Definition auf Deutsch liefert erneut Rothermel[5].

Studien belegen[7][8][9], dass ein Kontext nützlich für Anwendungen ist. Der Grund hierfür ist die Möglichkeit, die Anwendungen an verschiedene Situationen anpassen zu können, um dem Nutzer dadurch besser helfen zu können. Das wichtige Stichwort hierbei ist *Anpassung*. Dabei ist nicht nur der eigene Kontext relevant, sondern auch der Kontext von umgebenden Entitäten, mit denen die Anwendung interagieren könnte[7].

Ein Beispiel für eine kontextbezogene Anwendung ist der *Cyberguide*[8] aus 1996. Das Ziel dieses Systems ist es, einen Touristen bei der Besichtigung eines Museums oder einer Stadt so zu unterstützen und mit Informationen zu versorgen, wie es beispielsweise ein Stadtführer würde. Um dies zu ermöglichen, benötigt die Anwendung Informationen über den Aufenthaltsort sowie die Orientierung des Touristen. Eine aktuellere Studie bestätigt, dass sich für mobile Reisebegleiter kontextbezogene Anwendungen durchgesetzt haben, da sie von Touristen als nützlich wahrgenommen werden[9].

Abschließend lässt sich feststellen, dass der Kontext wichtig für mobile Anwendungen ist, da sie sich mithilfe der zusätzlichen Informationen individuell an den Nutzer anpassen und ihn ergo besser unterstützen können.

2.2. Mobile Dienste

Unter einem mobilen Dienst wird ein Programm auf einem mobilen Gerät verstanden, das alle vom Nutzer gewünschten Anwendungen in sich vereint. Dadurch ist es nicht mehr nötig, alle Anwendungen einzeln zu installieren. Stattdessen wird lediglich der mobile Dienst benötigt, der die jeweiligen Apps bereitstellt. Die Nutzung mobiler Dienste ist bisher kaum verbreitet, jedoch beginnen bereits einige Unternehmen damit, mehrere Anwendungen in einer einzelnen zu verbinden. Ein Beispiel hierfür ist die Vereinigung von Landkarten, Navigation und Fahrplänen öffentlicher Verkehrsmittel, wie sie bei Google Maps[10] (siehe Abbildung 2.1) verwendet wird. In Zukunft könnten mobile Dienste herkömmliche Apps ersetzen, da durch diese die Installation und Verwaltung zahlreicher verschiedener Anwendungen entfiele.

Hierin liegt auch bereits der große Vorteil, der sich durch die Verwendung mobiler Dienste ergibt. Während die Installation einzelner Anwendungen oft mühsam ist, muss

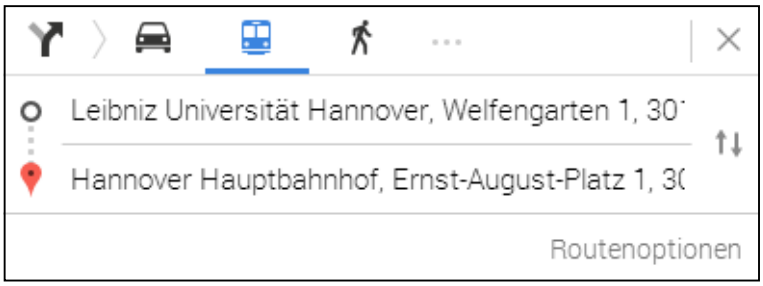

Abbildung 2.1.: Google Maps Navigation-Auswahlmaske

der Nutzer für den mobilen Dienst lediglich eine Installation vornehmen. Des Weiteren steigert ein mobiler Dienst auch die Übersichtlichkeit auf einem mobilen Gerät, da keine Vielzahl von Apps mehr vorhanden ist. Dadurch erhöht sich der Komfort und die Einfachheit bei der Benutzung, weil keine Zeit mehr damit verbracht werden muss, nach der gewünschten Anwendung zu suchen, sondern diese sofort im mobilen Dienst gefunden werden kann.

Um die Benutzbarkeit eines mobilen Dienstes weiter zu erhöhen und die Vorteile zu verstärken, wird der Kontext benötigt, mit dessen Hilfe besser auf die Anforderungen des Nutzers eingegangen werden kann. Durch die zusätzlichen Informationen soll es möglich werden, vorauszusagen, welche App der Nutzer als Nächstes benötigt. Sobald dieser anschließend den mobilen Dienst aufruft, soll die entsprechende Anwendung direkt angeboten werden, sodass die Suche nach dieser entfällt und dem Nutzer viel Zeit erspart wird.

[11] zeigt, dass sich Nutzer aus verschiedenen Gründen einer mobilen Anwendung bedienen. Ein Grund ist hierbei beispielsweise Zeitmangel, ein anderer ist der Aufenthaltsort des Nutzers. Dadurch, dass ein mobiler Dienst durch zusätzliche Kontextinformationen diese Gründe im Voraus erkennen kann, wird es möglich, den Nutzer direkt in jeder Situation gezielt zu unterstützen.

2.3. Trajektorie-Analyse

Für das Verständnis der Trajektorie-Analyse, muss zunächst geklärt werden, was unter einer Trajektorie zu verstehen ist: Eine Trajektorie ist eine Sequenz von zeitlich annotierten Punkten im Raum. Im Zusammenhang dieser Arbeit befinden sich die Punkte im zweidimensionalen Raum und beschreiben die aufeinander folgenden Aufenthaltsorte von Menschen. Eine Trajektorie stellt also die Spur der Bewegung eines Menschen dar. Dabei sind die aufeinander folgenden Punkte jeweils so geordnet, dass über die Reihenfolge direkt die Bewegung rekonstruiert werden kann. Die Aufzeichnung dieser Daten kann beispielsweise durch einen GPS-Empfänger oder durch die Beobachtung mit Kameras erfolgen. Eine formale Definition für eine Trajektorie gibt Giannotti[12]:

Definition 1 (Trajektorie) *Eine Trajektorie ist eine Sequenz von Tripeln*

$$T = \langle (x_0, y_0, t_0), \ldots, (x_k, y_k, t_k) \rangle,$$

wobei $t_i (i = 0, \ldots, k)$ *ein Zeitstempel ist,* $\forall \quad 0 \leq i \leq k \qquad t_i < t_{i+1}$ *und* (x_i, y_i) *sind die Punkte im* R^2*. Die Länge einer Trajektorie ist* k*.*

Als Nächstes wird eine Definition für einen Richtungscode gegeben, da dieser für das später vorgestellte Verfahren von zentraler Bedeutung sein wird. Jeder Eintrag des Richtungscodes enthält die absolute Richtung des Vektors zwischen zwei aufeinander folgenden Punkten in der Trajektorie. Somit lässt sich eine Trajektorie komplett in einen Richtungscode übersetzen, der anschließend einen Eintrag weniger als die Ausgangs-Trajektorie enthält.

Definition 2 (Richtungscode) *Gegeben sei eine Trajektorie* $T = \langle (x_0, y_0, t_0), \ldots, (x_k, y_k, t_k) \rangle$*. Ein Richtungscode ist eine Sequenz*

$$R(T) = (r_0, r_1, \ldots, r_{k-1}),$$

wobei k *die Länge von* T *ist und* $r_i (i = 0 \ldots k-1)$ *der Winkel zwischen den Vektoren* $v_1 = \begin{pmatrix} x_{i+1} - x_i \\ y_{i+1} - y_i \end{pmatrix}$ *und* $v_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ *ist.*

Ähnlich zu der Definition des Richtungscodes wird eine Definition eines Distanzcodes benötigt, der die euklidischen Distanzen zwischen zwei aufeinander folgenden Punkten der Trajektorie enthält.

Definition 3 (Distanzcode) *Gegeben sei eine Trajektorie* $T = \langle (x_0, y_0, t_0), \ldots, (x_k, y_k, t_k) \rangle$ *. Ein Distanzcode ist eine Sequenz*

$$D(T) = (d_0, d_1, \ldots, d_{k-1}),$$

wobei k *die Länge von* T *ist und* $d_i (i = 0 \ldots k-1)$ *die euklidische Distanz zwischen den Punkten* $P_1 = (x_i, y_i)$ *und* $P_2 = (x_{i+1}, y_{i+1})$ *ist.*

Nun folgt eine Definition für Bewegungsmuster, die mit Hilfe des erarbeiteten Verfahrens gefunden werden sollen. Ein Muster ist - ähnlich zu einer Trajektorie - eine Folge von Punkten im zweidimensionalen Raum. Anstatt die Zeitpunkte den Punkten wie bei einer Trajektorie direkt zuzuordnen, enthält ein Muster eine zusätzliche zeitliche Annotation, die die vergangene Zeit zwischen zwei aufeinander folgenden Punkten des Musters beschreibt. Zudem ist ein Muster stets orts- und richtungsunabhängig. Dies bedeutet, dass der Startpunkt und die Startrichtung nicht definiert sind.

Definition 4 (Trajektorie-Muster) *[12] Ein Trajektorie-Muster ist ein Paar* (S, A) *, wobei* $S = \langle (x_0, y_0), \ldots, (x_k, y_k) \rangle$ *eine Sequenz von Punkten im zweidimensionalen Raum ist, und* $A = \langle a_1, \ldots, a_k \rangle$ *die zeitliche Annotation der Sequenz ist.*

Abschließend folgt noch eine Definition für die Ähnlichkeit von Trajektorien, mit deren Hilfe schließlich Muster gefunden werden können. Zwei Trajektorien werden hierbei als ähnlich bezeichnet, wenn sie Subsequenzen beinhalten, die bis auf kleine Abweichungen in ihren Richtungs- und Distanzcodes übereinstimmen (siehe Definition 5). Dies regeln die Bedingungen 1 und 2. Dabei dürfen die Subsequenzen nicht beliebig gebildet werden, sondern nur unter Beachtung einer maximalen Zeittoleranz zwischen den aufeinander folgenden Punkten jeder Sequenz (Axiome 3 4). Die Einschränkung 5, dass die Zeitpunkte der Subsequenzen jeweils übereinstimmen, gilt, wenn Daten mit äquidistanten Zeitschritten vorliegen. Sollte dies nicht gegeben sein, kann die Bedingung so abgewandelt werden, dass die Zeitpunkte nur ungefähr übereinstimmen.

Zusätzlich werden in dieser Arbeit zwei weitere Einschränkungen gemacht. Die erste ist, dass die Ähnlichkeit der beiden Trajektorien jeweils am Anfang beginnen muss (Axiom 6). Dies liegt darin begründet, dass am Anfang jeder Trajektorie jeweils ein Intentionswechsel stattgefunden haben muss und somit nur Muster, die am Anfang beginnen, als relevant betrachtet werden. Eine tiefgreifendere Begründung folgt im Kapitel 5 (Trajektorie-Analyse). Die zweite Einschränkung ist, dass mindestens jeweils eine Hälfte der gesamten Trajektorien ähnlich zueinander sein müssen (Axiom 7). Diese Bedingung ist notwendig, um zu verhindern, dass Trajektorien, die sich nur über wenige Punkte ähneln, später als Muster erkannt werden.

Definition 5 (Ähnlichkeit von Trajektorien) *Gegeben seien zwei Trajektorien* $T = \langle (x_{T_0}, y_{T_0}, t_{T_0}), \ldots, (x_{T_n}, y_{T_n}, t_{T_n}) \rangle$ *und* $U = \langle (x_{U_0}, y_{U_0}, t_{U_0}), \ldots, (x_{U_n}, y_{U_n}, t_{U_n}) \rangle$*, eine maximale Richtungsabweichung* δ*, eine maximale Distanzabweichung* ϵ*, sowie eine Zeittoleranz* τ*.*
T *ist ähnlich zu* U *falls es zwei Subsequenzen* $T' = \langle (x'_{T_0}, y'_{T_0}, t'_{T_0}), \ldots, (x'_{T_k}, y'_{T_k}, t'_{T_k}) \rangle$ *von* T *und* $U' = \langle (x'_{U_0}, y'_{U_0}, t'_{U_0}), \ldots, (x'_{U_k}, y'_{U_k}, t'_{U_k}) \rangle$ *von* U *mit ihren zugehörigen Richtungs- und Distanzcodes* $R(T') = (r_0, r_1, \ldots, r_{k-1})$*,* $R(U') = (s_0, s_1, \ldots, s_{k-1})$*,* $D(T') = (d_0, d_1, \ldots, d_{k-1})$*,* $D(U') = (f_0, f_1, \ldots, f_{k-1})$ *gibt, sodass*

1. $\forall \quad 0 \leq i \leq k-1 \qquad |r_i - s_i| \leq \delta$
2. $\forall \quad 0 \leq i \leq k-1 \qquad |d_i - f_i| \leq \epsilon$
3. $\forall \quad 0 \leq i \leq k-1 \qquad t'_{T_{i+1}} - t'_{T_i} \leq \tau$
4. $\forall \quad 0 \leq i \leq k-1 \qquad t'_{U_{i+1}} - t'_{U_i} \leq \tau$
5. $\forall \quad 0 \leq i \leq k-1 \qquad t'_{T_i} = t'_{U_i}$
6. $t'_{T_0} - t_{T_0} \leq \tau$ *und* $t'_{U_0} - t_{U_0} \leq \tau$
7. $0.5 * n \leq k$ *und* $0.5 * m \leq k$

Das Problem der Erkennung von Bewegungsmustern in Trajektorien liegt nun darin, Trajektorien zu finden, die zu möglichst vielen anderen Trajektorien ähnlich sind und somit als Muster bezeichnet werden können. Im Folgenden werden Bewegungen, die mindestens zweimal gefunden wurden, als Muster bezeichnet.

3. Verwandte Arbeiten

Die im Folgenden vorgestellten verwandten Arbeiten beschäftigen sich mit dem Sammeln von Kontextinformationen eines mobilen Nutzers.

Zunächst werden zwei Arbeiten vorgestellt, in denen die Aktivität eines mobilen Nutzers durch sein Smartphone erkannt wird. Berchtold et al.[13] haben eine Anwendung namens *ActiServ* entwickelt, die automatisch erkennt, welche Aktivität der Nutzer gerade durchführt, und dementsprechend Einstellungen wie beispielsweise die Lautstärke ändert. Die Informationen hierfür werden über die Sensoren des Smartphones gesammelt. In einem Video[14] stellen die Autoren schließlich vor, wie die Anwendung funktioniert und welche Vorteils sie mit sich bringt. Sobald das Smartphone auf einen Tisch gelegt wird, wird diese Aktivität erkannt und das Gerät auf stumm geschaltet, da sich der Nutzer in einer Besprechung befindet. Wird es wieder aufgenommen, schaltet sich die Lautstärke automatisch wieder ein. Bei dieser Anwendung werden also Kontextinformationen über die Sensoren des Smartphones gesammelt.

Eine weitere Arbeit in diesem Bereich gibt es von Mayrhofer et al.[15]. Hierin stellen die Autoren eine Architektur vor, die den Kontext des Nutzers erkennen und zusätzlich voraussagen kann. Dazu werden verschiedene interne Sensoren wie ein Mikrofon, Bluetooth, Wireless LAN und externe Sensoren wie andere Smartphones, die über Bluetooth erreichbar sind, genutzt. Das Ziel der Autoren ist hierbei erneut, den Kontext eines Nutzers zu erkennen und somit Anwendungen zu entwickeln, die den Nutzer selbstständig unterstützen.

Gordon et al.[16] fassen diverse Arbeiten zum Thema *Erkennung der Aktivität eines Nutzers* zusammen. Zusätzlich befassen sich die Autoren mit dem Problem des Energiemangels, der oft der limitierende Faktor für die Nutzung der Sensoren eines Smartphones ist. Von ihnen wird ein Ansatz vorgestellt und implementiert, der trotz der großen Energieeinsparung qualitativ ähnliche hochwertige Vorhersagen trifft.

Alle bisher genannten Arbeiten grenzen sich von dieser ab, da in ihnen Kontextinformationen direkt über die Sensoren des Smartphones gesammelt werden. In dieser Arbeit jedoch sollen Informationen abhängig von Bewegungen des beobachteten Nutzers gesammelt werden. Somit lassen sich die Kontextinformationen, die durch die vorgestellten Anwendungen gesammelt werden, mit den im Rahmen dieser Arbeit gewonnenen kombinieren, um eine genauere Vorhersage für mobile Nutzer treffen zu können.

Eine andere Arbeit von Ashbrook und Starner[17] beschäftigt sich mit der Vorhersage der Bewegung eines Nutzers. Hierzu werden GPS-Daten verschiedener Nutzer gesammelt. Anschließend werden wichtige Plätze ermittelt, indem untersucht wird, wo sich Nutzer besonders lang aufhalten. Daraufhin werden die Bewegungen der beobachteten Menschen in eine Abfolge der wichtigen Plätze übersetzt, sodass vorherge-

sagt werden kann, zu welchem Platz sich eine Person als nächstes bewegen wird, wenn sie zuvor andere Plätze in einer bestimmten Reihenfolge besucht hat. Die Kontextinformationen werden in dieser Arbeit also über GPS-Daten und eine Übersetzung in wichtige Plätze generiert.

Die Arbeit von Laasonen[18] verfolgt dasselbe Ziel. Auch in dieser sollen Bewegungen vorhergesagt werden können, indem eine Datenbank von wichtigen Plätzen aufgebaut wird, die Nutzer regelmäßig besuchen. Anschließend werden Clustering Algorithmen genutzt, um die Routen der beobachteten Menschen voraussagen zu können. Die Informationen werden beispielsweise über GSM-Netze empfangen und verarbeitet.

Taniar und Goh[19] versuchen auf eine ähnliche Weise Bewegungsmuster von mobilen Nutzern zu sammeln. Ihr Ansatz unterscheidet sich von den beiden zuvor genannten dahingehend, dass zwei feste Standorte, zwischen denen Nutzer wechseln, betrachtet werden. Anschließend wird untersucht, wie sich die Nutzer vom einen zum anderen Standort bewegen, um daraus Muster ableiten zu können. Schließlich kann vorhergesagt werden, welche Zwischenstationen die Person auf ihrem Weg zwischen den Standorten besuchen wird.

Die drei zuletzt genannten Arbeiten grenzen sich von dieser Arbeit ab, da keine feinen Bewegungsmuster gesammelt werden, sondern eine grobe Aneinanderreihung von Plätzen, die vom Nutzer nacheinander besucht wurden. Die bearbeiteten Trajektorien weisen eine größere Granularität als in dieser Arbeit auf, sodass die Vorhersagen somit lediglich darüber getroffen werden können, welcher Platz oder welche Zwischenstation als nächstes besucht wird.

Ein ähnlicher Ansatz wird von Yavas et al.[20] verfolgt. Die Autoren dieser Arbeit stellen einen Algorithmus vor, der den nächsten Schritt eines Nutzers voraussagen kann. Dazu werden in einer ersten Phase Muster, die aus einer Abfolge von Zellen bestehen, gesammelt. Dazu wird das Gebiet, in dem sich der Nutzer bewegt, in Rasterzellen unterteilt. In einer zweiten Phase werden Regeln aufgestellt, die besagen, welche Zelle der Nutzer als Nächstes besuchen wird, wenn er zuvor eine bestimmte Folge von Zellen besucht hat. Im letzten Schritt des Verfahrens kann schließlich mit Hilfe der Regeln vorhergesagt werden, wohin der Nutzer sich nun bewegen wird. Diese Arbeit grenzt sich von der vorgestellten insofern ab, als sich die Art des Sammelns der Bewegungsmuster unterscheidet, da die gefundenen Muster auf einer Trajektorie anstatt auf einer Folge von Rasterzellen basieren. Des Weiteren verfolgt die Publikation von Yavas et al. nicht das Ziel, den gefundenen Bewegungsmustern Intentionen zuzuordnen. Als Intention wird hierbei die Absicht, etwas zu tun, bezeichnet. Die referenzierte Arbeit beschränkt sich darauf, vorherzusagen, wohin sich der Nutzer im Anschluss bewegt.

In den bisher genannten Publikationen werden geographische Informationen genutzt, um Bewegungsmuster zu finden. Im Gegensatz dazu stellen Ying et al.[21] in ihrer Arbeit einen Ansatz vor, der sowohl geographische als auch semantische Trajektorien untersucht. Als semantische Trajektorie wird eine Abfolge von Plätzen, die mit semantischen Kennzeichnungen versehen sind, bezeichnet (z.B. Kirche, Marktplatz, Rathaus). Auch hier wird im Gegensatz zu dieser Arbeit eine Abfolge von Plätzen betrachtet, anstatt feine Bewegungsmuster zu sammeln. Die semantischen Kennzeichnungen kön-

nen jedoch als zusätzliche Informationen für feine Bewegungsmuster genutzt werden, um den Kontext zu verdeutlichen.

Die letzte referenzierte Arbeit von Tseng und Lin[22] stellt ein Verfahren vor, mit dessen Hilfe Standorte in der Welt mit Webseiten verknüpfen werden können, die vom mobilen Nutzer besucht wurden, während er sich an diesem Standort aufhielt. Hierzu wurde ein Algorithmus entwickelt, der in der Lage ist zu untersuchen, ob der Verknüpfung von Standorten mit Webseiten ein Muster zugrunde liegt. Es werden jedoch keine Bewegungsmuster gesucht- im Gegensatz zu der in dieser Arbeit durchgeführten Methodik. Der Ansatz ist der in dieser Arbeit verfolgten Intention ähnlich. Durch dieses Verfahren wird es möglich, dem Nutzer stets die richtige Webseite zu öffnen, wenn er sich an einem bestimmten Standort aufhält.

4. Mobile Dienste

Das Ziel dieser Arbeit ist die Erkennung von Bewegungsmustern, um mobile Dienste unterstützen zu können. Aufgrund dessen werden in diesem Kapitel Konzepte vorgestellt, die benötigt werden, um aus den Ergebnissen der Trajektorie-Analyse einen Nutzen für mobile Dienste ziehen zu können. Zunächst wird betrachtet, inwiefern Intention und Bewegung in Zusammenhang stehen und welche Erkenntnisse daraus für die mobilen Dienste gewonnen werden können. Anschließend werden verschiedene Explorationsarten aufgezeigt und den entsprechenden Bewegungsmustern zugeordnet. Zum Schluss folgt ein Vergleich zwischen den Vorgehensweisen Bottom-up und Top-down, die als Ansätze für das Sammeln der Bewegungsdaten in Frage kommen. Parallel werden geeignete Beispiele zur Verdeutlichung der vorgestellten Konzepte angeführt.

4.1. Zusammenhang zwischen Intention und Bewegung

Um voraussagen zu können, welche mobile Anwendung der Nutzer nach oder während einer bestimmten Bewegung benötigt, muss das gefundene Bewegungsmuster mit einer Intention verknüpfen werden. Wird ein Bewegungsmuster gefunden ohne diesem jedoch eine Intention zuordnen zu können, ist es für einen mobilen Dienst wertlos, da keine Informationen daraus abgeleitet werden können. Das Ziel muss also sein, von der Bewegung des Nutzers auf sein momentanes Anliegen bzw. Ziel schließen zu können.

Ein Beispiel hierfür ist eine schnelle, geradlinige Bewegung des beobachteten Nutzers. Die Intention dieser Bewegung ist wahrscheinlich das schnelle Erreichen eines anderen Orts, da der Nutzer sich geradlinig und schnell bewegt. Unter Zuhilfenahme weiterer (zum Beispiel semantischer) Kontextinformationen, kann diese Intention konkretisiert werden. Findet die Bewegung beispielsweise in der Nähe und in Richtung eines Bahnhofs statt, lässt sich mit hoher Wahrscheinlichkeit feststellen, dass die Intention des Nutzers das Erreichen eines Zuges ist. Ein mobiler Dienst sollte in diesem Fall den aktuellen Fahrplan aller bald abfahrenden Züge in einer kompakten Form bereitstellen, um dem Nutzer relevante Informationen auf dem neuesten Stand bereitzustellen.

Ein anderes Beispiel aus dem Alltag ist die Bewegung eines Nutzers in einem Kreis mit kleinem Radius. Diese Bewegung lässt vermuten, dass der Nutzer sich an seinem momentanen Aufenthaltsort nicht auskennt oder etwas innerhalb der vom Kreis eingeschlossenen Fläche sucht. Um auch diesem Nutzer helfen zu können, sollte ein mobiler Dienst in diesem Fall eine Karte des Gebiets und Informationen über die in der

Fläche liegenden Orte oder Gebäude bereitstellen.

Ein drittes Beispiel, das im Gegensatz zu den bisher genannten verdeutlichen soll, dass nicht jeder Bewegung eindeutig eine Intention zugeordnet werden kann, ist folgendes: Der beobachtete Nutzer geht langsam und mäanderförmig. Für diese Bewegung kommen nun verschiedene Intentionen in Frage: Ein möglicher Grund für diese Bewegung könnte sein, dass der Nutzer auf dem Weg zum Bahnhof ist und sich hierbei Zeit lässt, da sein Zug erst später fährt. Ein anderer möglicher Grund ist, dass der Nutzer durch die Stadt schlendert und sich umliegende Geschäfte anschaut. Beide Fälle erfordern unterschiedliche Reaktionen des mobilen Diensts. Während im ersten Fall ein detaillierter Fahrplan für den Nutzer hilfreich wäre, würde er im zweiten Fall Informationen über Angebote oder Aktionen der umliegenden Geschäfte bevorzugen.

Somit lässt sich feststellen, dass ein Zusammenhang zwischen Intention und Bewegung erkennbar ist. Dieser Zusammenhang ist jedoch nicht immer eindeutig, sodass sich möglicherweise einer Bewegung nicht nur eine einzige Intention zuordnen lässt. Wie auch im ersten Beispiel ersichtlich, werden oftmals weitere Kontextinformationen benötigt, um mit hoher Wahrscheinlichkeit die für den Nutzer richtige Anwendung auszuwählen. Sollte sich einem Bewegungsmuster jedoch gar keine Intention zuordnen lassen, ist dieses Muster wertlos und ohne Bedeutung für einen mobilen Dienst.

4.2. Explorationsarten

Mit der voranschreitenden Entwicklung der Computer im täglichen Leben, zeichnet sich ein Trend zum *Ubiquitous Computing* ab[23]. Dies bedeutet, dass Computer allgegenwärtig in unserer Umwelt verankert sind und dem Menschen bei seinen Tätigkeiten helfen, ohne dass dieser es bemerkt. Ein Schritt in diese Richtung ist[3]. Hier beschreibt der Autor, wie kontextbezogene Dienste in die Umwelt integriert werden können, indem diese an bestimmten Orten installiert werden und nur über eine bestimmte Distanz zur Verfügung stehen. Ein in diesem Zusammenhang genanntes Beispiel ist das Anlegen eines Dienstes für ein Parkhaus, der nur in einem bestimmten Umkreis zur Verfügung steht.

Um in die Umwelt integrierte, kontextbezogene Dienste für mobile Nutzer interessant zu machen, dürfen diese nicht ständig zur Verfügung stehen, sondern nur, wenn sie benötigt werden. Beispielsweise sollte vermieden werden, einem Nutzer, der nur an einem Parkhaus vorbeigeht, Informationen über die Preise für das Parken anzuzeigen. Um dies zu erreichen, sollen den verschiedenen Bewegungsmustern verschiedene Explorationsarten zugeordnet werden.

Explorationsarten sind Flächen, die sich um den Nutzer herum befinden. Jeder kontextbezogene Dienst, dessen Gebiet sich mit der Explorationsart des Nutzers überlappt, wird für den Nutzer als interessant eingestuft. Mit zusätzlichen Kontextinformationen kann die Auswahl an kontextbezogenen Diensten weiter eingeschränkt werden, sodass der mobile Dienst nur die für den Nutzer relevanten Informationen und Anwendungen anzeigt.

Als Formen für die Explorationsarten eignen sich zunächst einfache geometrische Formen, die im Folgenden anhand von Beispielen verdeutlicht werden. Für die Karten im Hintergrund wurden Screenshots von OpenStreetMap[24] genutzt.

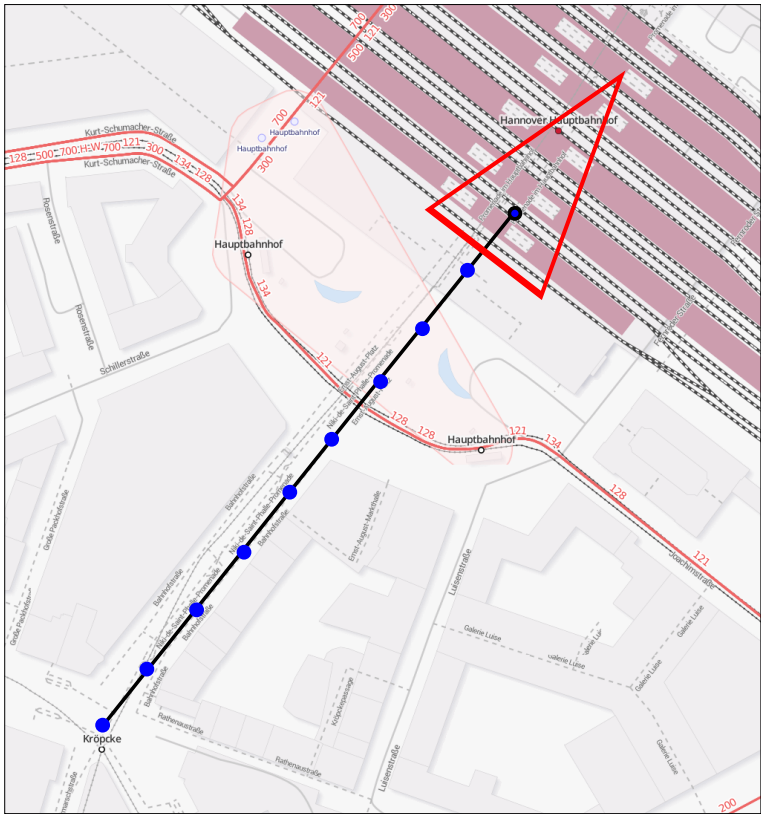

Abbildung 4.1.: Explorationsart 1: spitzes Dreieck

Das erste Beispiel ist ein spitzes Dreieck. Es eignet sich bei schnellen, geradlinigen Bewegungen. Wie in Abbildung 4.1 zu sehen, bewegt sich die beobachtete Person schnell auf den Bahnhof zu. Die Geschwindigkeit der Bewegung kann an den Abständen der aufgezeichneten Punkte erkannt werden. Je größer die Abstände sind, desto schneller bewegt sich das beobachtete Objekt. Dies gilt jedoch nur, wenn die Punkte in regelmäßigen Zeitabständen aufgenommen wurden. Das spitze Dreieck ist für diese Bewegung als Explorationsart angebracht, da das Ziel der Person offensichtlich in Richtung der Bewegung liegt und wahrscheinlich noch weit entfernt ist. Je langsamer die Bewegung wird, desto stumpfer sollte der entsprechende Winkel des Dreiecks gewählt und desto weiter sollte das Dreieck zu den Seiten ausgedehnt werden, da das Ziel des Objekts wahrscheinlich nicht mehr weit entfernt ist.

Das zweite Beispiel ist eine Ellipse, die sich senkrecht zu der Bewegungsrichtung der beobachteten Person ausdehnt. Sie eignet sich für langsame, mäanderförmige Bewegungen. Abbildung 4.2 zeigt die beobachtete Person sich auf den Bahnhof zu bewegend. Diesmal jedoch nicht geradlinig, sondern mäanderförmig und deutlich langsamer, wie an den verminderten Abständen zwischen den einzelnen Messpunkten zu

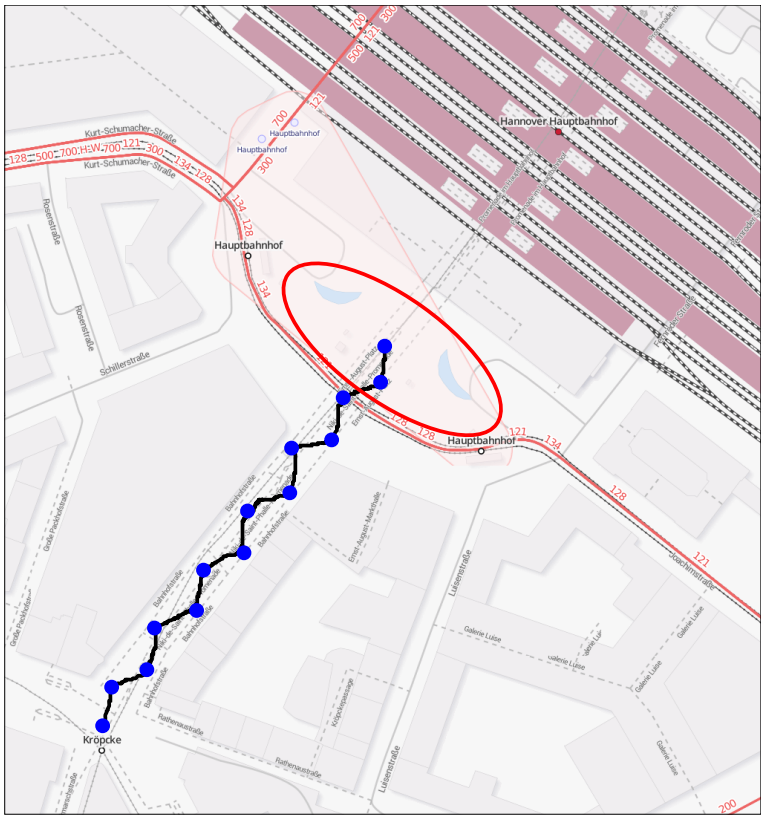

Abbildung 4.2.: Explorationsart 2: Ellipse

erkennen ist. Diese Bewegung deutet darauf hin, dass die Person nicht in Eile ist und kein festes Ziel in Richtung der Bewegung anstrebt. Vielmehr könnte sie bedeuten, dass die Person sich in der Gegend umschaut. Dabei sind besonders die kontextbezogenen Dienste interessant, die senkrecht zur Bewegungsrichtung liegen. Dies liegt darin begründet, dass sich die Person vermutlich zur Seite umschaut und aufgrund der langsamen Bewegung eher nicht an Diensten vor oder hinter sich interessiert ist.

Das dritte Beispiel ist ein Kreis, der so verschoben ist, dass er sich zur inneren Seite der Kreisbewegung ausdehnt. Er eignet sich für kreisförmige Bewegungen. Wie in Abbildung 4.3 zu sehen, bewegt sich die beobachtete Person in einem Kreis, sodass sie fast wieder an den Ausgangspunkt der Bewegung zurückgekehrt ist. Dies ist keine typische Bewegung eines Menschen, sondern deutet vielmehr darauf hin, dass sich die beobachtete Person nicht auskennt und möglicherweise etwas in der von der Bewegung eingeschlossenen Fläche sucht. Deshalb ist ein nach innen verschobener Kreis die richtige Explorationsart für diesen Fall. Mit ihm können kontextbezogene Dienste bereitgestellt werden, nach denen die Person sucht oder die der Person helfen können, sich besser in der fremden Umgebung zurechtzufinden.

Da diese Arbeit anhand von Fußballdaten evaluiert wird, lassen sich die gefundenen Muster nicht direkt mit Explorationsarten versehen. Daher wurden Bewegungsmuster gewählt, die sich potenziell in der Stadt finden lassen. Die Stadt ist eine Umgebung,

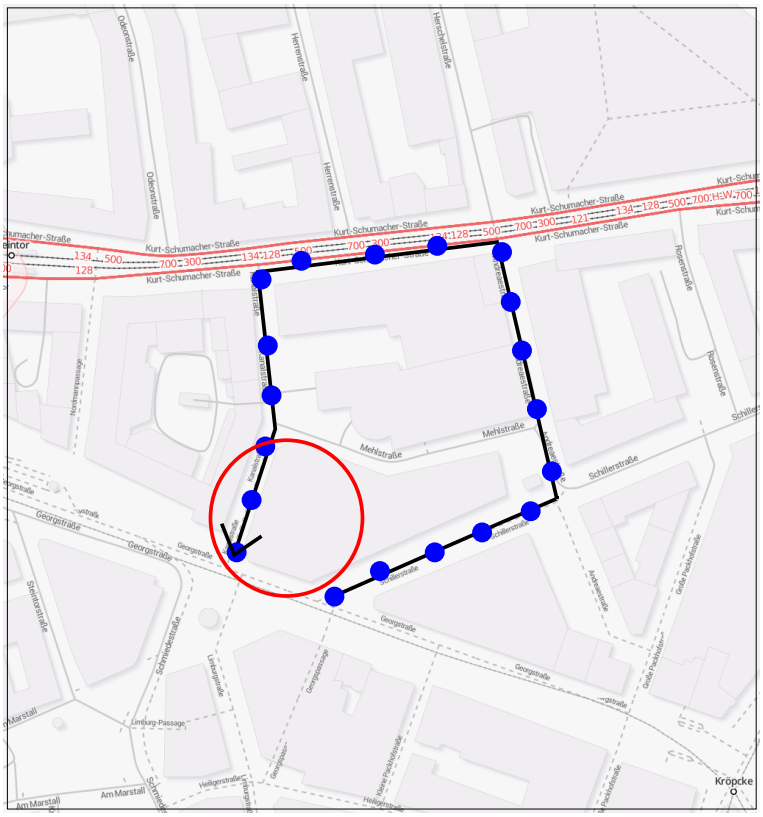

Abbildung 4.3.: Explorationsart 3: Kreis

in der viele kontextbezogene Dienste sowie viele verschiedene Bewegungsmuster vorkommen. Deswegen eignet sie sich deutlich besser, um die vorgestellten Explorationsarten erklären und mit Beispielen versehen zu können.

Die vorgestellten Explorationsarten sind als Beispiele zu verstehen und decken nicht alle Bewegungsarten ab. Sie eignen sich, um Explorationsarten und deren Umsetzung zu evaluieren, da sie typische Bewegungen berücksichtigen. In Zukunft muss getestet werden, ob die vorgestellten Explorationsarten ausreichend sind oder ob es weitere Bewegungsarten gibt, die andere Formen benötigen. Des Weiteren ist die Wahl der Parameter der geometrischen Form entscheidend. Anhand von Studien können zukünftig Zusammenhänge zwischen Bewegungsarten und Parametern gefunden werden. Außerdem müssen die Explorationsarten in anderen Umfeldern getestet werden, um die Herangehensweise über kontextbezogene Dienste und Flächen für verschiedene Explorationsarten hinsichtlich ihres Nutzens zu bewerten.

4.3. Zwei Ansätze zur Bestimmung des Kontextes

Bei der Wahl des Ansatzes stehen zwei Alternativen zur Auswahl: Bottom-up (von unten nach oben) oder Top-down (von oben nach unten). Der Top-down-Ansatz startet

mit der Betrachtung, welche Bewegungsmuster potenziell in der untersuchten Umgebung gefunden werden und mit welcher mobilen Anwendung sie assoziiert sind. Anschließend wird evaluiert, ob die Bewegungsmuster tatsächlich in der Umgebung gefunden werden können und auf sie stets ein Aufruf der spezifizierten mobilen Anwendung folgt. Bei dieser Vorgehensweise wird zunächst ein Gesamtbild erstellt, das im Anschluss in seine Einzelteile zerlegt und überprüft wird.

Im Gegensatz dazu steht der Bottom-up-Ansatz. Hier werden zunächst anhand gesammelter Daten Bewegungsmuster gefunden. Anschließend wird getestet, ob und wie die gefundenen Bewegungsmuster mit mobilen Anwendungen verknüpft werden können. Dies kann zum Beispiel, wie in dieser Arbeit vorgestellt, über Intentionen erreicht werden. Eine andere Möglichkeit wäre die Verteilung von Fragebögen, um herauszufinden, welche mobilen Anwendungen die Anwender während der Beobachtung nutzten. Bei dieser Vorgehensweise werden also zunächst einzelne Informationen zusammengetragen, die anschließend zu einem Gesamtbild geformt werden.

In einer aktuellen Studie wurde untersucht, wie sich die beiden Ansätze unterscheiden, wenn mit ihnen das Verhalten von Nutzern in sozialen Medien modelliert wird[25]. Dabei kommen die Autoren zu der Erkenntnis, dass beide Verfahren keine großen Unterschiede im Ergebnis aufweisen. Diese Erkenntnis lässt sich zum Teil auch auf diese Arbeit übertragen. Jedoch unterscheiden sich die Verlässlichkeit und der Aufwand beider Ansätze, wie im folgenden Absatz erläutert wird.

Der Vorteil des Bottom-up-Ansatz liegt darin, dass von vornherein Fakten auswertet werden, anstatt - wie beim Top-down-Ansatz - Hypothesen aufzustellen. Die Daten sind sehr einfach zu sammeln (z.B. indem Nutzer mit GPS-Empfängern ausstattet werden). Der Nachteil ist jedoch, dass nur bereits gezeigte Intentionen bekannt sind und deswegen die Datenbasis breit genug sein muss.

Auch Unternehmen wie *Amazon.com* nutzen den Bottom-Up-Ansatz, um das Kundenverhalten auf ihrer Webseite zu beobachten. Es werden zunächst Daten gesammelt, die anschließend genutzt werden, um dem Kunden Empfehlungen anzuzeigen[26].

Zusammengefasst gilt insbesondere für diese Arbeit, dass der Bottom-up-Ansatz zu bevorzugen ist. Die große Menge an Daten, die ohne großen Aufwand gesammelt werden kann, sorgt für eine Basis, auf der der mobile Service aufgebaut werden kann. Dafür werden zunächst mit dem in dieser Arbeit entwickelten Verfahren die Bewegungsmuster gesammelt und anschließend mit Intentionen verknüpft. Dies kann geschehen, indem Kunden interviewt werden. Anschließend wird der mobile Dienst basierend auf diesen Daten aufgebaut.

5. Trajektorie-Analyse

In diesem Kapitel werden nun die einzelnen Schritte des Verfahrens vorgestellt, mit dessen Hilfe Bewegungsmuster gefunden werden sollen. Zunächst werden die Voraussetzungen betrachtet, die gegeben sein müssen, um einen ordnungsgemäßen Ablauf des Verfahrens zu gewährleisten (Kapitel 5.1). Anschließend wird die Segmentierung behandelt, deren Notwendigkeit erläutert, und wie sie zu erreichen ist (Kapitel 5.2). Daraufhin wird der wichtigste Teil des Verfahrens vorgestellt, in dem zueinander ähnliche Segmente gesammelt und gespeichert werden. Anschließend wird die Aufbereitung der gefundenen Muster erläutert.

5.1. Voraussetzungen und Überblick

Um Trajektorie-Analyse betreiben zu können, werden zunächst verschiedene Anforderungen an die Daten gestellt. Die erste wichtige Bedingung ist, dass ausreichend Daten von verschiedenen Personen gesammelt wurden. Liegen Daten von einer Person vor oder sind sie nicht umfangreich genug, können durch das Verfahren nur unzureichend viele aussagekräftige Muster gefunden werden. Des Weiteren muss sichergestellt werden, dass die Daten genügend hoch frequentiert sind. Sind die Zeitabstände zwischen den einzelnen aufgenommenen Punkten zu groß, besteht die Gefahr, dass gefundene Muster in der Realität gar nicht vorhanden waren. Dies begründet sich darin, dass die Personen in den Zeitabständen, in denen nicht gemessen wurde, völlig unterschiedliche Bewegungen ausgeführt haben könnten.

Die nächste Voraussetzung ist, dass die eingegebenen Daten keine großen Abweichungen oder Messfehler aufweisen. Dies lässt sich jedoch bei der Menge an benötigten Daten kaum verhindern, sodass eine Filterung notwendig ist. Hierfür kommen Tiefpassfilter in Frage, die den hochfrequenten Anteil aus den Messdaten herausfiltern, wodurch die Trajektorie geglättet wird. Eine mögliche Implementierung ist ein Butterworth-Filter. Die Filterung ist wichtig, um zu verhindern, dass an Stellen segmentiert wird, an denen lediglich ein Messfehler in den Daten das Kriterium für die Segmentierung auslöst.

Des Weiteren ist eine zentrale Voraussetzung für das Verfahren, dass die Trajektorien in eine orts- und richtungsunabhängige Form gebracht werden, denn Muster sind unabhängig vom Startpunkt und ihrer Startrichtung definiert. Wie dies erreicht werden kann, wird in Unterkapitel 5.3 erläutert.

Ein anderer, wichtiger Faktor, der vor der Trajektorie-Analyse beachtet werden muss, ist die Umgebung, in der die Daten aufgenommen wurden. Durch sie ergeben sich unterschiedliche Einstellungen für das Verfahren, die jeweils angepasst werden müssen.

Beispielsweise benötigt das Verfahren für Fußball-Daten andere Einstellungen als für Stadt-Daten. Dieser Punkt wird unten eingehender diskutiert.

Eine weiterer Vorverarbeitungsschritt, der das Verfahren erleichtert, ist, dass die Daten in äquidistante Zeitabstände gebracht werden. Dies bedeutet, dass die Zeitabstände, in denen die einzelnen Punkte aufgenommen wurden, stets den gleichen Wert annehmen. Realisiert werden kann dies beispielsweise mit Hilfe von Interpolation. Dies ist hilfreich, da zwischen aufeinander folgenden Punkten nicht stets der Zeitabstand berechnet werden muss, falls er relevant ist. Dieser Vorverarbeitungsschritt ist optional, sodass das Verfahren auch auf Daten abgewandelt werden kann, die diese Eigenschaft nicht aufweisen.

Es folgt ein kurzer Überblick über das gesamte Verfahren, bevor in den weiteren Unterkapiteln detaillierter auf die einzelnen Schritte eingegangen wird.

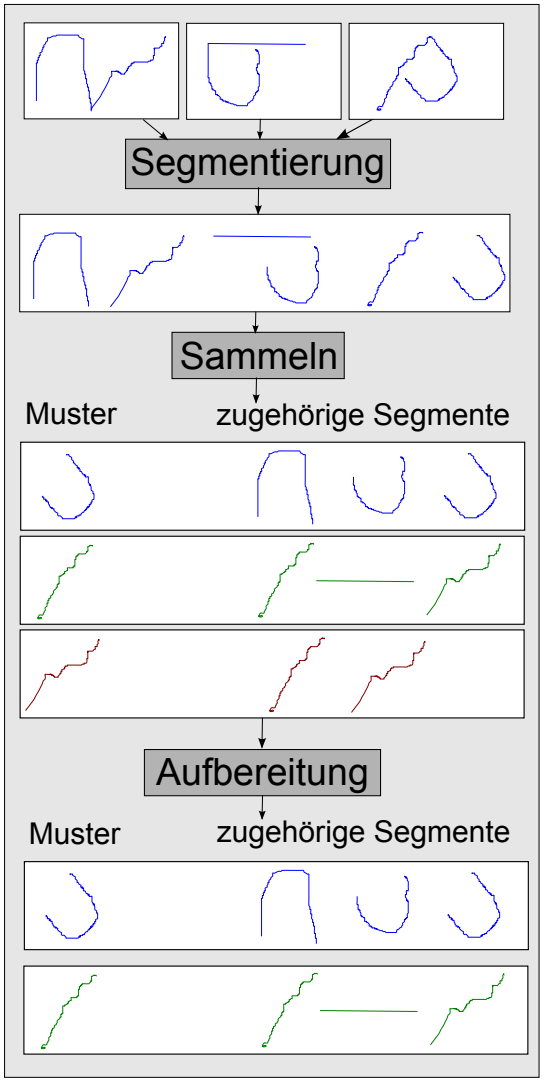

Abbildung 5.1.: Übersicht über das Verfahren

Als Eingabe für das Verfahren werden die Trajektorien der beobachteten Personen benötigt. Hier ist zunächst zu beachten, dass zu Anfang jeder Person nur genau eine Trajektorie zugeordnet werden kann, sodass es unmöglich ist, Muster in diesen zu finden. Also müssen die Trajektorien zuerst aufgetrennt werden, um eine Vielzahl von Segmenten zu erzeugen, in denen schließlich Muster gefunden werden können. Im nächsten Schritt können diese Segmente nun miteinander vergleichen werden und somit ähnliche Segmente gefunden werden, die gemeinsam ein Muster bilden. Zum Schluss müssen diese schließlich noch aufbereitet werden, um sicherzustellen, dass ein einzelnes Segment nicht mehreren Mustern angehört. Der Ablauf des Verfahrens und die einzelnen Ergebnisse sind anhand eines Beispiels in Abbildung 5.1 dargestellt. Hier tritt der Fall auf, dass zwei Muster gefunden werden, die gleiche Trajektorien enthalten. Aufgrund dessen wird in der Aufbereitung das Muster mit der geringeren Anzahl an zugehörigen Trajektorien gelöscht.

5.2. Segmentierung

Der erste Schritt des Verfahrens ist die Segmentierung. Hier werden die Trajektorien nach zwei Kriterien, die in den Unterkapiteln näher erläutert werden, zerschnitten. Dies ist notwendig, da das Verfahren einen Anhaltspunkt benötigt, wo die einzelnen Muster beginnen. Wenn Ähnlichkeiten in unsegmentierten Daten gesucht werden müssten, könnte jeder mögliche Teil der einen Trajektorie zu jedem möglichen Teil der anderen Trajektorie ähnlich sein. Durch die Segmentierung und die zusätzliche Bedingung, dass Muster jeweils am Anfang eines Segments beginnen, wird das Problem darauf reduziert, nur noch jedes Segment mit jedem möglichen anderen zu vergleichen. Die Einschränkung ist valide, wenn die Trajektorien dort segmentiert werden, wo ein Intentionswechsel stattgefunden hat. Dieser Intentionswechsel wird dann angenommen, wenn die beobachtete Person ihre Bewegung stark ändert. Das neue Segment beginnt somit direkt mit einer neuen Intention, die später für die mobilen Dienste wichtig ist.

Die im Folgenden vorgestellten Verfahren zur Segmentierung orientieren sich an der Arbeit Buchins und anderer[27]. Die einzelnen Kriterien zur Segmentierung werden hier kombiniert.

5.2.1. Geschwindigkeit

Das erste Kriterium zur Segmentierung ist die Geschwindigkeit. Die Daten sind entweder direkt im Datensatz enthalten oder aber für jede Kante zwischen zwei Punkten berechnet worden. Anschließend muss definiert werden, dass die Geschwindigkeit an einem Punkt entweder der Geschwindigkeit der eingehenden oder der ausgehenden Kante entspricht. Die Segmentierung erfolgt so, dass jedes Segment nur einen maximalen Geschwindigkeitsunterschied aufweisen darf, der als Parameter übergeben wird. Ein Beispiel hierfür ist in Abbildung 5.2 zu sehen. Die Trajektorie wird an dem roten Strich geschnitten, da der Geschwindigkeitsunterschied an diesem Punkt 6 m/s beträgt und somit die erlaubten 5 m/s überschreitet.

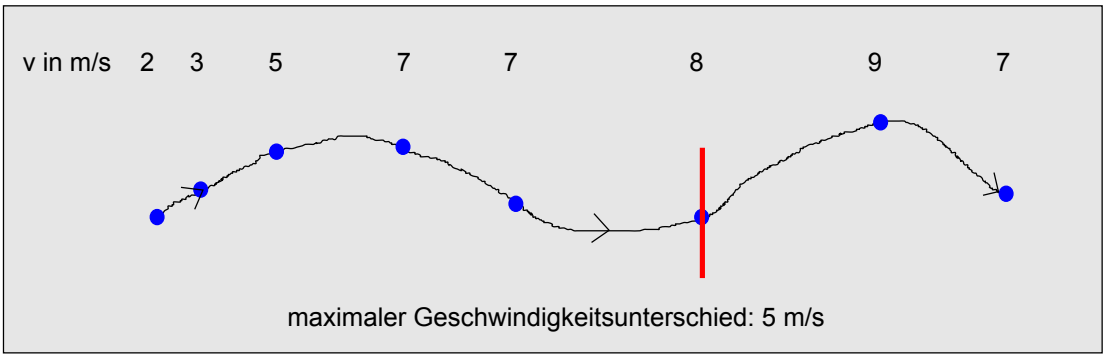

Abbildung 5.2.: Segmentierung nach Geschwindigkeit

Der Algorithmus 5.1 zeigt die Segmentierung nach Geschwindigkeit. Als Eingabe bekommt er die zu segmentierende Trajektorie sowie den Geschwindigkeitsunterschied, der innerhalb eines Segments maximal erlaubt ist. Die Variablen v_{min} und v_{max} enthalten die minimale bzw. maximale Geschwindigkeit, die bisher im Segment erreicht wurde. Anschließend durchläuft der Algorithmus jeden Punkt der Trajektorie (Zeile 5) und zerschneidet diese, falls die hinzugekommene Geschwindigkeit die Spanne zwischen der minimalen und maximalen Geschwindigkeit über den maximalen Geschwindigkeitsunterschied steigen lässt (Zeile 9). Am Ende gibt der Algorithmus schließlich die Liste der erstellten Segmente zurück.

Algorithmus 5.1 : Segmentierung nach Geschwindigkeit (angelehnt an[27])

Input : Trajektorie T
maximaler Geschwindigkeitsunterschied δ
Output : Liste von Segmenten S

1 P_0 = erster Punkt von T;
2 v_{min} = Geschwindigkeit(P_0);
3 v_{max} = Geschwindigkeit(P_0);
4 $a = 0$;
5 **foreach** $P_i \in T$ **do**
6 $\quad v$ = Geschwindigkeit(P_i);
7 $\quad v_{max} = \max(v, v_{max})$;
8 $\quad v_{min} = \min(v, v_{min})$;
9 $\quad$ **if** $v_{max} - v_{min} > \delta$ **then**
10 $\quad\quad S = S \cup T[a, i]$; /* neues Segment hinzufügen */
11 $\quad\quad a = i$;
12 $\quad\quad v_{min}$ = Geschwindigkeit(P_i) ; /* Variablen zurücksetzen */
13 $\quad\quad v_{max}$ = Geschwindigkeit(P_i);
14 $\quad$ **end**
15 **end**
16 **return** S

Dieses Kriterium eignet sich gut, um Intentionswechsel zu erkennen, da mit unter-

schiedlichen Geschwindigkeiten oft auch unterschiedliche Intentionen verbunden sind. Dadurch, dass die Geschwindigkeit über das gesamte Segment betrachtet wird, lassen sich sowohl schleichende als auch abrupte Geschwindigkeitswechsel erfassen.

5.2.2. Bewegungsrichtung

Das zweite Kriterium, das im Rahmen dieser Arbeit zur Segmentierung angewandt wird, ist die Richtung. Auch hierfür können die Daten direkt aus dem Datensatz entnommen werden, falls sie enthalten sind, oder aber - wie in dieser Arbeit - berechnet werden. Die Segmentierung funktioniert so, dass zwischen zwei aufeinander folgenden Vektoren der Trajektorie ein maximaler Winkel nicht überschritten werden darf. Wichtig ist, dass die Kanten der Trajektorie als ortsunabhängige Vektoren dargestellt werden, zwischen denen letztendlich der Winkel berechnet wird. Der maximale Richtungsunterschied kann als Parameter übergeben werden. Zusätzlich gibt es einen weiteren Parameter, der festlegt, über wie viele Punkte der Richtungsunterschied bestehen muss, damit die Trajektorie segmentiert wird. Dadurch wird sichergestellt, dass die beobachtete Person den Richtungswechsel nicht nur für einen kurzen Moment vollzogen hat.

Abbildung 5.3 stellt ein Beispiel für diese Segmentierung dar. Hier wurde parametrisiert, dass der Richtungsunterschied über zwei Punkte bestehen muss. Deswegen werden wie abgebildet zwei Winkel α und β zwischen den entsprechenden Vektoren berechnet. Da beide Winkel den maximalen Richtungsunterschied δ überschreiten, wird die Trajektorie schließlich am mittleren Punkt zwischen den betrachteten Kanten geschnitten.

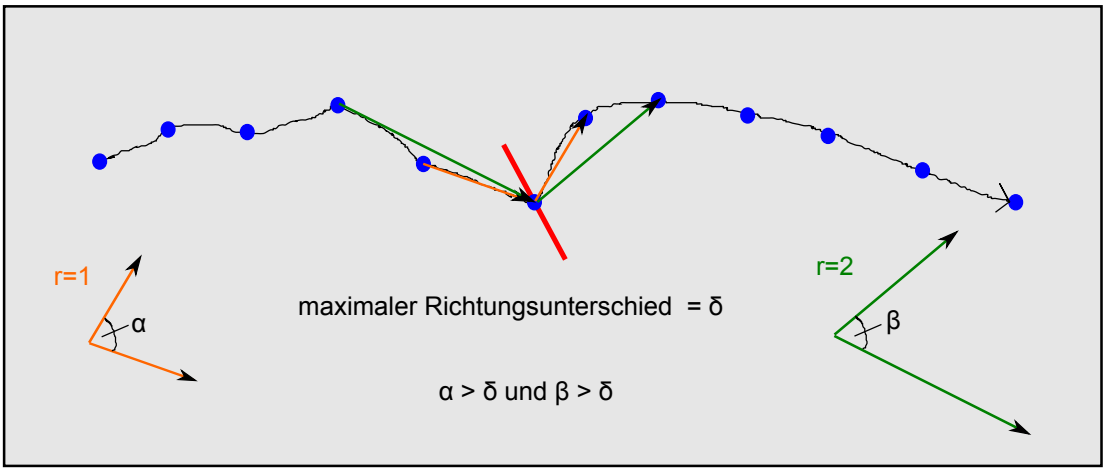

Abbildung 5.3.: Segmentierung nach Richtung

Algorithmus 5.2 zeigt den Pseudocode zur Segmentierung nach Richtung. Wie oben erwähnt, benötigt der Algorithmus als Eingabe die zu segmentierende Trajektorie, einen maximalen Richtungsunterschied sowie die Reichweite der Segmentierung. Wie auch bei der Segmentierung nach der Geschwindigkeit durchläuft der Algorithmus die

gesamte Trajektorie und überprüft an jedem Punkt die Notwendigkeit einer Segmentierung. Dazu wird in Zeile 4 der Winkel zwischen dem Vektor der eingehenden und dem Vektor der ausgehenden Kante des Punktes berechnet. Wenn dieser Winkel den maximal erlaubten Wert übersteigt, werden in der inneren Schleife (ab Zeile 7) für jede mögliche Reichweite Winkel zwischen zwei neuen Vektoren berechnet. Der eine Vektor geht hierbei von dem Punkt, der in der aktuellen Reichweite **vor** dem Mittelpunkt liegt, zum Mittelpunkt. Der andere Vektor hingegen geht vom Mittelpunkt zu dem Punkt, der in der aktuellen Reichweite **hinter** dem Mittelpunkt liegt. Überschreiten alle berechneten Winkel den maximalen Richtungsunterschied, wird die Trajektorie schließlich segmentiert und das neu gewonnene Segment dem Ergebnis hinzugefügt (Zeile 15). Die Variable a wird benötigt, um zu markieren, wo das letzte Segment geendet hat und das neue Segment dementsprechend anfangen muss. Des Weiteren wird in der Variable b gespeichert, ob der maximale Richtungsunterschied für alle untersuchten Winkel Bestand hatte und dementsprechend die Trajektorie segmentiert werden muss.

Algorithmus 5.2 : Segmentierung nach Richtung (angelehnt an[27])

Input : Trajektorie T mit Länge n
maximaler Richtungsunterschied δ
Reichweite r
Output : Liste von Segmenten S

1 $a = 0$;
2 **for** $i = r$ **to** $n - r$ **do**
3 $\quad P_{i-1}, P_i, P_{i+1} \in T$;
4 $\quad w$ = Winkel zwischen $\begin{pmatrix} P_i.x - P_{i-1}.x \\ P_i.y - P_{i-1}.y \end{pmatrix}$ und $\begin{pmatrix} P_{i+1}.x - P_i.x \\ P_{i+1}.y - P_i.y \end{pmatrix}$;
5 $\quad$ **if** $w > \delta$ **then**
6 $\quad\quad b$ = true;
7 $\quad\quad$ **for** $j = 2$ **to** r **do**
8 $\quad\quad\quad P_{i-j}, P_{i+j} \in T$;
9 $\quad\quad\quad w'$ = Winkel zwischen $\begin{pmatrix} P_i.x - P_{i-j}.x \\ P_i.y - P_{i-j}.y \end{pmatrix}$ und $\begin{pmatrix} P_{i+j}.x - P_i.x \\ P_{i+j}.y - P_i.y \end{pmatrix}$;
10 $\quad\quad\quad$ **if** $w' \leq \delta$ **then**
11 $\quad\quad\quad\quad b$ = false;
12 $\quad\quad\quad$ **end**
13 $\quad\quad$ **end**
14 $\quad\quad$ **if** b = *true* **then**
15 $\quad\quad\quad S = S \cup T[a, i]$; `/* neues Segment hinzufügen */`
16 $\quad\quad\quad a = i$;
17 $\quad\quad$ **end**
18 $\quad$ **end**
19 **end**
20 **return** S

Auch dieses Kriterium eignet sich gut, um einen Intentionswechsel feststellen zu kön-

nen, da mit einer Änderung der Bewegungsrichtung oftmals ein Intentionswechsel einhergeht. Zwischen abrupten und langsamen Richtungswechseln kann durch den zusätzlichen Parameter unterschieden werden, sodass sich die Segmentierung an verschiedene Umgebungen anpassen lässt.

5.3. Sammeln von ähnlichen Segmenten

Um ähnliche Segmente sammeln zu können, müssen die Daten zunächst in eine orts- und richtungsunabhängige Form gebracht werden. Durch die Nutzung des Richtungs- und Distanzcodes, wie in Kapitel 2 definiert, sind die Daten bereits ortsunabhängig, da kein Startpunkt definiert wird und dieser somit für alle Segmente im Ursprung liegt. Richtungsunabhängigkeit wird erreicht, indem die Anfangsrichtung jedes Segments von allen Richtungen innerhalb des Richtungscodes subtrahiert wird. Somit starten alle Segmente in dieselbe Richtung und sind richtungsunabhängig. Ein Beispiel hierfür ist in Abbildung 5.4 zu sehen.

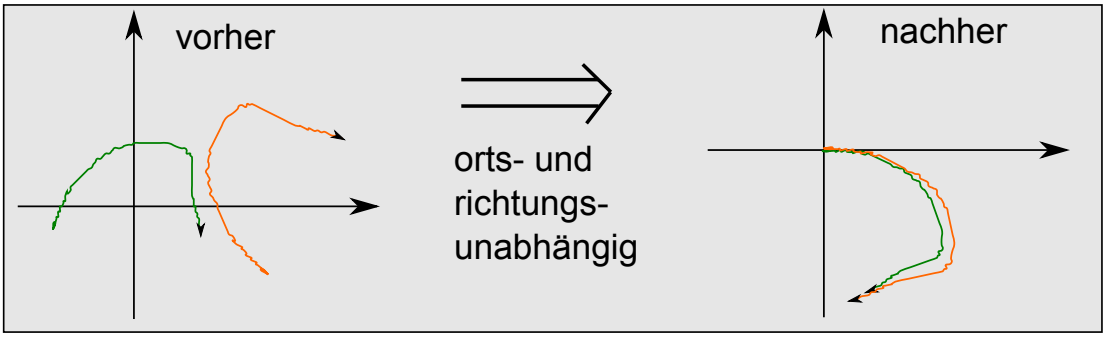

Abbildung 5.4.: Umwandlung in eine orts- und richtungsunabhängige Form

Damit Segmente ähnlich und nicht komplett identisch sein müssen, werden die in den Grundlagen definierten Parameter verwendet (siehe 2.3):

- *maximale Richtungsabweichung* δ. Sorgt dafür, dass jede Kante in einem Segment jeweils eine kleine Richtungsabweichung zu ihrem Partner im anderen Segment haben darf.
- *maximale Distanzabweichung* ϵ. Toleriert analog zur Richtungsabweichung Unterschiede in den Distanzcodes der untersuchten Segmente.
- *Zeittoleranz* τ. Ermöglicht, dass die Segmente über den kurzen, gegebenen Zeitraum unterschiedlich verlaufen dürfen.
- *minimale Unterstützung* π. Legt die minimale Anzahl an Segmenten fest, die ein Muster enthalten muss.
- *minimale Länge* λ. Definiert die minimale Länge jedes Musters.

Der Algorithmus unterteilt sich in zwei Teile. Im ersten Teil wird überprüft, ob zwei Segmente ähnlich zueinander sind. Hierzu eignet sich beispielsweise der Needleman-

Wunsch-Algorithmus[28], der ursprünglich dazu gedacht war, Aminosäuresequenzen zu vergleichen und ihre Ähnlichkeit zu berechnen. Er wird auch genutzt, um Ähnlichkeiten in anderen Sequenzen aufzuspüren. Im zweiten Teil des Algorithmus werden schließlich alle Segmente miteinander verglichen und jeweils ähnliche Segmente in einem Muster zusammengefasst, das jedoch noch nicht das Ergebnis des Verfahrens ist.

Algorithmus 5.3 zeigt den Pseudocode für den Teil des Verfahrens zum Sammeln ähnlicher Segmente. Das Ziel dieses Algorithmus ist, herauszufinden, für wie viele Punkte die beiden übergebenen Trajektorien unter Beachtung der Parameter übereinstimmen. Dafür werden zunächst in Zeile 1 bis 4 die Richtungs- und Distanzcodes der beiden übergebenen Trajektorien berechnet. Anschließend werden alle Codes durchlaufen und die entsprechenden Einträge jeweils miteinander verglichen. Stimmen zwei Einträge auch unter Beachtung der Toleranzen nicht mehr überein, wechselt der Algorithmus in eine innere Schleife. Hier wird überprüft, ob die Abweichung auch über die gegebene Zeittoleranz bestehen bleibt (siehe Zeile 10 bis 13). Ist dies der Fall, so wird schließlich in Zeile 17 die Länge der Übereinstimmung zurückgegeben. Sie beträgt die Anzahl der Einträge, die bisher durchlaufen wurden, da die Trajektorien bis hier übereingestimmt haben (wenn genau $i - 1$ Einträge übereinstimmen, stimmen i Punkte der Trajektorien überein, da ein Eintrag im Code zwischen zwei Punkten liegt). Bleibt die Abweichung nicht über die Zeittoleranz bestehen, setzt der Algorithmus mit den nächsten beiden Einträgen der Richtungs- und Distanzcodes fort. Die Variable b dient hierbei zur Speicherung, ob die Abweichung über den übergebenen Zeitraum Bestand hatte oder nicht.

Abbildung 5.5 zeigt ein Beispiel zu Algorithmus 5.3. Es ist zu sehen, dass die Distanzen und Richtungen der ersten sechs Kanten der Segmente sehr ähnlich sind. Danach weichen die Einträge in den Richtungscodes zu stark voneinander ab, sodass die Segmente nicht mehr ähnlich verlaufen. Aufgrund dessen endet die Übereinstimmung der Trajektorien an der roten gestrichelten Linie.

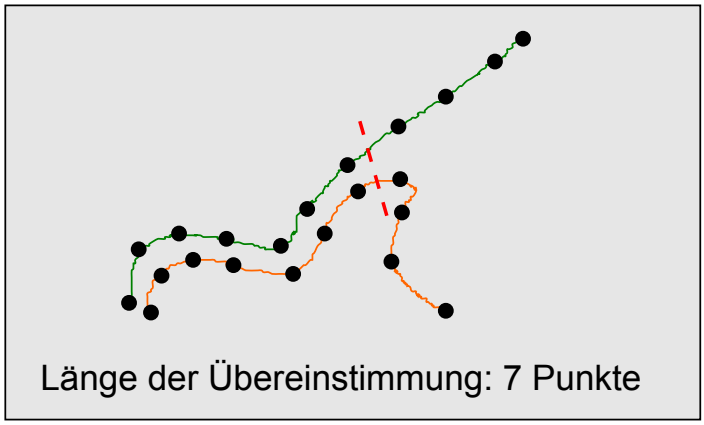

Abbildung 5.5.: Länge der Übereinstimmung zweier Trajektorien

Für die weitere Darstellung des Verfahrens wird hier eine Definition des Ergebnisformats geliefert. Ein Ergebnis besteht aus drei Teilen: dem Bewegungsmuster an sich,

Algorithmus 5.3 : Länge der Übereinstimmung zweier Trajektorien (angelehnt an [28])

```
   Input : zwei Trajektorien T und U
   maximale Richtungsabweichung δ
   maximale Distanzabweichung ϵ
   Zeittoleranz τ
   Output : Länge der Übereinstimmung k
1  R_T = Richtungscode(T);
2  R_U = Richtungscode(U);
3  D_T = Distanzcode(T);
4  D_U = Distanzcode(U);
5  n = min(Länge(R_T), Länge(R_U));
6  for i = 0 to n do
7      r_i ∈ R_T, s_i ∈ R_U, d_i ∈ D_T, f_i ∈ D_U;
8      if |r_i − s_i| > δ   oder   |d_i − f_i| > ϵ then
9          b = true;
10         for j = i to i + τ do
11             r_j ∈ R_T, s_j ∈ R_U, d_j ∈ D_T, f_j ∈ D_U;
12             if |r_j − s_j| ≤ δ   und   |d_j − f_j| ≤ ϵ then
13                 b = false;
14             end
15         end
16         if b = true then
17             return k = i
18         end
19     end
20 end
21 return k = n
```

einer Musterlänge k und allen unterstützenden Segmenten. In dem Bewegungsmuster ist das gesamte Segment gespeichert, das das Muster darstellt. Wird es angefordert, werden nur die ersten k Punkte zurückgegeben, da in k die Zahl gespeichert ist, die angibt, über wie viele Punkte sich alle unterstützenden Segmente ähneln. Zuletzt sind in den unterstützenden Segmenten diejenigen Segmente gespeichert, die dem Bewegungsmuster ähneln.

Der Pseudocode für den zweiten Teil des Verfahrens zum Sammeln ähnlicher Segmente ist in Algorithmus 5.4 zu sehen. Das Ziel in diesem Teil des Algorithmus ist nun, alle Muster zu finden, die mindestens von π Trajektorien unterstützt werden und eine Musterlänge von mindestens λ aufweisen. Hierzu werden zunächst in einer Schleife (siehe Zeile 1) alle übergebenen Segmente durchlaufen. Es wird dabei für jedes Segment ein neues Ergebnis angelegt, das wie folgt aufgebaut ist: Als Bewegungsmuster enthält es das aktuelle Segment, als unterstützendes Segment wird das aktuelle Segment hinzugefügt und als Musterlänge wird die Länge des aktuellen Segments gesetzt.

Hierauf wird in einer inneren Schleife (siehe Zeile 6) für jedes andere Segment überprüft, ob und über wie viele Punkte es ähnlich zu dem ersten Segment ist. Erreicht die Länge der Übereinstimmung die geforderte Länge λ, wird das Segment der inneren Schleife dem Ergebnis als unterstützendes Segment hinzugefügt und die Musterlänge, falls nötig, angepasst. Hierbei muss darauf geachtet werden, dass die Musterlänge immer der kleinsten Länge der Übereinstimmung entspricht (siehe Zeile 11).

Es ist ausreichend, die Richtungs- und Distanzcodes der Segmente zur Sammlung von ähnlichen Segmenten zu betrachten, da diese eindeutig darüber definiert sind. Dies ist daran zu erkennen, dass sich ein Segment mit Hilfe seines Richtungs- und Distanzcodes vollständig rekonstruieren lässt. Lediglich die ursprüngliche Richtung und der ursprüngliche Startpunkt können verloren gehen. Dies ist jedoch nicht von Relevanz für das Verfahren, da die Muster orts- und richtungsunabhängig gesucht werden.

Algorithmus 5.4 : Sammeln der Muster

Input : Liste von Segmenten L
maximale Richtungsabweichung δ
maximale Distanzabweichung ϵ
Zeittoleranz τ
minimale Unterstützung π
minimale Länge λ
Output : Liste von Ergebnissen E

```
1  for i = 0 to |L| do
2      S_i ∈ L;
3      M = neues Ergebnis mit S_i als Bewegungsmuster;
4      M = M ∪ S_i;
5      setze die Musterlänge von M auf |S_i|;
6      for j = 0 to |L|   , wobei i ≠ j do
7          S_j ∈ L;
8          k = Länge der Übereinstimmung von S_i und S_j mit δ, ϵ, τ;
9          if k ≥ λ und k ≥ |S_i * 0.5| und k ≥ |S_j * 0.5| then
10             M = M ∪ S_j;
11             setze die Musterlänge von M auf min(vorherige Musterlänge, k);
12         end
13     end
14     if |M| ≥ π then
15         E = E ∪ M;
16     end
17 end
18 return E
```

5.4. Aufbereitung der Muster

Das Ziel im letzten Schritt des Verfahrens ist, dass jedes Segment nur in einem einzigen Ergebnis vorkommt und dass jedes unterstützende Segment mindestens zu 50% durch das Muster abgedeckt ist. Zu diesem Zweck werden zwei Algorithmen entwickelt, die beide Bedingungen überprüfen und deren Einhaltung gewährleisten. In einem letzten Schritt am Ende wird getestet, ob alle Muster noch von der Mindestanzahl an Segmenten unterstützt werden. Ist dies nicht der Fall, werden entsprechende Ergebnisse gelöscht.

In Algorithmus 5.5 ist der Pseudocode zu sehen, der dazu führt, dass alle unterstützenden Segmente jeweils zu 50% durch das Bewegungsmuster abgedeckt sind, da lange Segmente, die nur ein kurzes Muster enthalten, unbrauchbar für mobile Dienste sind. Dadurch, dass das Segment im Vergleich zum Muster viel zu lang ist und somit nur ein kleiner Teil des Segments betrachtet wird, kann diesem Segment keine korrekte Intention zugeordnet werden. Hierfür werden alle unterstützenden Segmente durchlaufen, überprüft und - falls nötig - gelöscht.

Algorithmus 5.5 : Löschen von zu langen unterstützenden Segmenten

```
   Input : Liste von Ergebnissen L mit Länge n
   Output : Liste von Ergebnissen ohne zu lange unterstützende Segmente L
1  for i = 0 to n do
2      M_i ∈ L;
3      k = Musterlänge(M_i);
4      forall the unterstützenden Segmente S in M_i do
5          if k < |S| * 0.5 then
6              Lösche S aus den unterstützenden Segmenten von M_i;
7          end
8      end
9  end
10 return L
```

Der Algorithmus 5.6 stellt sicher, dass kein Segment in mehr als einem Ergebnis als unterstützendes Segment vorkommt. Dies ist nötig, da jedem Segment nur eine Intention zugeordnet werden kann und darf. Durch den Algorithmus werden Ergebnisse, die eine größere Anzahl an unterstützenden Segmenten enthalten, bevorzugt behandelt, sodass Ergebnisse mit weniger Unterstützung wegfallen, denen ohnehin schwieriger eine Intention zugeordnet werden kann, da sie seltener vorhanden sind. Enthält ein Ergebnis ein unterstützendes Segment, das gleichzeitig Bewegungsmuster eines Ergebnisses mit niedrigerer Unterstützung ist, wird das Ergebnis mit der niedrigen Unterstützung ganz gelöscht. Dieses geschieht, um einzelne Muster zu maximieren, denen anschließend besser eine Intention zugeordnet werden kann.

Die Funktionsweise des Algorithmus ist dabei wie folgt: Vor jeder Iteration wird die Liste von übergebenen Ergebnissen neu geordnet, sodass diejenigen mit höherer Unterstützung zuerst bearbeitet werden (siehe Zeile 2). Jedes Muster, das bereits bearbei-

tet wurde, wird markiert und enthält anschließend nur noch Segmente, die in keinem anderen vorhanden sind. In jedem Durchlauf der Schleife in Zeile 1 wird ein unmarkiertes Ergebnis bearbeitet. Dazu wird es zunächst markiert. Anschließend wird in allen unmarkierten Ergebnissen überprüft, ob diese ein Segment enthalten, das auch das momentan zu bearbeitende Ergebnis enthält (siehe Zeile 6 bis 10). Ist dies der Fall, wird zwischen zwei Möglichkeiten unterschieden: Entspricht das kritische Segment dem Bewegungsmuster des Ergebnisses, muss dieses Ergebnis komplett gelöscht werden, da das Bewegungsmuster auch immer Teil der unterstützenden Segmente sein muss. Entspricht es ihm jedoch nicht, reicht es, das kritische Segment aus den unterstützenden Segmenten des Ergebnisses zu entfernen.

Algorithmus 5.6 : Löschen doppelt vorkommender unterstützender Segmente

```
   Input : Liste von Ergebnissen L mit Länge n
   Output : Liste von Ergebnissen ohne doppelt vorkommende unterstützende
            Segmente L
 1 while es gibt unmarkierte Ergebnisse do
 2     Sortiere die Ergebnisse absteigend nach der Anzahl der unterstützenden
       Segmente;
 3     M_i = nächstes unmarkiertes Ergebnis;
 4     markiere M_i;
 5     A = alle unterstützenden Segmenten von M_i;
 6     forall the unmarkierte Ergebnisse M_j do
 7         B = alle unterstützenden Segmenten von M_j;
 8         forall the A_i ∈ A do
 9             forall the B_i ∈ B do
10                 if A_i = B_i then
11                     if B_i gleichzeitig Bewegungsmuster von M_j then
12                         lösche das Ergebnis M_j und fahre in Zeile 6 fort;
13                     end
14                     else
15                         lösche B_i aus den unterstützenden Segmenten von M_j;
16                     end
17                 end
18             end
19         end
20     end
21 end
```

Nachdem die Muster mit Hilfe der beiden Algorithmen aufbereitet wurden, muss im letzten Schritt sichergestellt werden, dass jedes noch die ausreichende Anzahl an unterstützenden Segmenten enthält. Ist dies nicht der Fall, muss das Ergebnis nachträglich gelöscht werden. Anschließend ist die Ausgabe vollständig und in der Form, dass jedem Muster Intentionen zugeordnet werden können. Das Verfahren zur Trajektorie-Analyse ist somit abgeschlossen.

6. Evaluation und Diskussion

Die Evaluation wurde mit einer prototypischen Implementierung des Verfahrens aus Kapitel 5 durchgeführt. Der hierbei untersuchte Datensatz stammt von einem Fußballspiel. Die drei zuvor vorgestellten Schritte des Verfahrens zur Trajektorie-Analyse werden während dieses Kapitels mit 5.2 *Segmentierung*, 5.3 *Sammeln* und 5.4 *Aufbereitung* abgekürzt.

In diesem Kapitel wird zunächst dargelegt, zu welchem Zweck die Evaluation durchgeführt wird und worauf sie sich konzentriert. Anschließend wird der Aufbau der Evaluation erläutert und erklärt, welche Vorbereitungen getroffen wurden. Daraufhin folgt die Präsentation des Ergebnisses sowie eine Diskussion, die verschiedene Themen der Arbeit aufgreift.

Während des gesamten Kapitel ist zu beachten, dass die Werte für die Geschwindigkeiten direkt aus dem gegebenen Datensatz übernommen wurden. Dadurch könnten sie höher als erwartet sein.

6.1. Ziel der Evaluation

Das Ziel der Evaluation unterteilt sich in drei Aspekte, die im Folgenden erläutert werden:

1. Werden genügend Punkte der Ausgangsdaten einem Muster zugeordnet? Diese Frage ist wichtig, da eine hohe Abdeckung der Ausgangsdaten durch Muster gegeben sein muss. Wenn nicht genug Daten durch ein Muster abgedeckt werden, bedeutet dies für einen mobilen Dienst, dass nur selten zusätzliche Kontextinformationen über die Bewegung des Nutzers gesammelt werden können.
2. Werden genügend Segmente durch ein Muster abgedeckt? Bei dieser Frage liegt der Fokus darauf, besonders aussagekräftige Segmente zu finden und nur wenige Segmente keinem Muster zuordnen zu können. Diese Frage ist wichtig, da sich die Ziele des Verfahrens innerhalb verschiedener Umfelder unterscheiden können.
3. Kann gefundenen Mustern eine Intention zugeordnet werden? Dies ist eine qualitative Untersuchung des Verfahrens, da nur solche Muster wertvoll sind, denen tatsächlich eine Intention zugeordnet werden kann. Es ist zu beachten, dass die hier gefundenen Intentionen im Vergleich mit anderen Umfeldern weniger zahlreich sind.

6.2. Vorbereitung und Durchführung

Um die Evaluation durchführen zu können, wurde das vorgestellte Verfahren unter Java implementiert sowie eine zusätzliche graphische Oberfläche entwickelt, um die gefundenen Muster betrachten zu können. Um die Laufzeit des Verfahrens im Rahmen zu halten, wurden die Daten von vier verschiedenen Fußballspielern genutzt, wobei jeweils zwei Spieler der gleichen Spielposition gewählt wurden. Die Frequenz der Daten betrug 2 Hz.

Zunächst wurde innerhalb eines automatisierten Testlaufs die Segmentierung mit unterschiedlichen Eingabeparametern getestet und die gefundenen Ergebnisse zusammengetragen. Hierbei war das Ziel herauszufinden, wie viele Segmente unterschiedliche Eingabeparameter zur Folge haben. Aufgrund dieser Erkenntnisse konnten bereits einige Parameter für den weiteren Verlauf der Evaluation ausgeschlossen werden.

Im nächsten Schritt wurde das Verfahren als Ganzes evaluiert. Dazu wurde erneut ein automatisierter Testlauf geschrieben, in dem zunächst alle Parameter variiert wurden. Als Ergebnis wurden erneut die genutzten Parameter und die sich daraus ergebene Anzahl der Segmente, Anzahl der Muster, Abdeckung der Punkte, Abdeckung der Segmente und die durchschnittliche Distanz zwischen den Mustern erfasst. Für die durchschnittliche Distanz zwischen den Mustern wurde jeweils die Distanz zwischen zwei Mustern berechnet und anschließend der Durchschnitt über diesen Distanzen gebildet. Der Abstand zwischen zwei Mustern wurde ermittelt, indem jeweils die euklidische Distanz zwischen den Punkten des Musters berechnet wurde, die sich am nächsten lagen, und anschließend über alle Distanzen gemittelt wurde. Die Abdeckung der Punkte wurde ermittelt, indem der Anteil der Punkte, die einem Muster angehörten, an den gesamten Punkten der Ausgangsdaten berechnet wurde. Analog wurde für die Abdeckung der Segmente verfahren. Die Anzahl der Segmente wurde nach der Segmentierung bestimmt, während die Anzahl der Muster nach der Aufbereitung berechnet wurde. Durch diesen Schritt wurde sichergestellt, dass es keine anderen Parametersätze gibt, die eine höhere Abdeckung erreichen, als jene, die im Laufe des Kapitels vorgestellt werden.

Basierend auf den neuen Erkenntnissen wurde im folgenden Schritt das Verfahren auf drei unterschiedliche Vorgehensweisen untersucht. Zuerst war das Ziel, die Parameter der Segmentierung so einzustellen, dass die Abdeckung der Punkte maximiert wurde (siehe 6.3.1). Anschließend wurde dieser Schritt analog für die Abdeckung der Segmente wiederholt (siehe 6.3.2). Zum Schluss wurde untersucht, mit welchen Parametern für das Sammeln der ähnlichen Segmente die besten Ergebnisse erzielt werden konnten (siehe 6.3.3). Hierbei wurde zwischendurch mit Hilfe der graphischen Oberfläche überprüft, ob die gefundenen Muster nach subjektivem Empfinden noch als solche bezeichnet werden konnten. Für die drei Vorgehensweisen wurde erneut ein automatisierter Testlauf erstellt.

Im letzten Schritt wurden mit Hilfe der graphischen Oberfläche die gefundenen Muster bewertet und - soweit möglich - ihnen eine Intention zugeordnet (siehe 6.3.4). Als Werte für die Parameter wurden hierbei jene ausgewählt, die in den zuvor durchgeführten

Schritten gefunden worden waren und sich in der quantitativen Analyse als am besten geeignet erwiesen hatten.

6.3. Ergebnis

In diesem Unterkapitel werden nun schließlich die Ergebnisse der Evaluation vorgestellt. Es unterteilt sich in vier Unterabschnitte, die jeweils unterschiedliche Aspekte der Evaluation darstellen. Die ersten drei Unterabschnitte befassen sich hierbei mit der *quantitativen* Analyse des Verfahrens, während im letzten Unterabschnitt eine *qualitative* Analyse gegeben wird. Im ersten Unterabschnitt wird das erste Ziel der Evaluation bearbeitet. Analog befasst sich der zweite Unterabschnitt mit dem zweiten Ziel. Im dritten Unterabschnitt werden die Parameter für das Sammeln der ähnlichen Segmente vorgestellt und untersucht. Der letzte Unterabschnitt zielt schließlich auf das dritte Ziel der Evaluation ab.

6.3.1. Maximierung der Punkt-Abdeckung

In diesem Unterabschnitt werden die Ergebnisse der Evaluation dargestellt, die eine Maximierung der Punkt-Abdeckung bezweckte. Hierfür wurden die Parameter für die Segmentierung einzeln nacheinander variiert, während die anderen Parameter des Verfahrens festgehalten wurden. Die folgenden Parameter gelten also für den gesamten Unterabschnitt:

- Maximaler Richtungsunterschied für Sammeln: 0,4 π
- Maximale Distanzabweichung für Sammeln: 1 m
- Mindestlänge des Musters: 6 Punkte
- Zeittoleranz: 4 Punkte $\hat{=}$ 2 s
- Mindest-Unterstützung: 2 Segmente

Da nur ein Parameter für jeden Durchlauf variiert wurde, mussten die übrigen zwei Parameter der Segmentierung ebenso festgehalten werden. Hierfür wurden die optimalen Werte dieser Parameter gewählt, um zu zeigen, welche Punkt-Abdeckung tatsächlich erreicht werden kann. Vor der Durchführung der im Folgenden dargestellten Durchläufe wurden alle möglichen Kombinationen von Parametern ausprobiert, um sicherzustellen, dass der optimale Parametersatz gefunden wird.

In den Tabellen 6.1, 6.2 und 6.3 wurden die einzelnen Parameter der Segmentierung variiert. Dadurch wurden diejenigen Parameter gefunden, die für diesen Datensatz die höchste Punkt-Abdeckung erreichen:

- Maximaler Richtungsunterschied für Segmentierung: 0,29 π
- Maximaler Geschwindigkeitsunterschied für Segmentierung : 8,4 m/s
- Reichweite: 7 Punkte

Tabelle 6.1.: Variieren des Parameters *Maximaler Richtungsunterschied für Segmentierung* zur Maximierung der Punkt-Abdeckung

Maximaler Richtungsunterschied für Segmentierung in π	Anzahl Segmente	Anzahl Muster	Segment-Abdeckung	Punkt-Abdeckung	Durchschnittliche Distanz zwischen Mustern in mm
0,20	4008	35	0,43	0,55	6248
0,23	3412	40	0,48	0,57	7043
0,26	2918	46	0,53	0,57	7301
0,29	2557	54	0,57	**0,59**	7079
0,32	2192	53	0,61	0,55	7086
0,35	1920	61	0,64	0,54	7633

Tabelle 6.2.: Variieren des Parameters *Maximaler Geschwindigkeitsunterschied für Segmentierung* zur Maximierung der Punkt-Abdeckung

Maximaler Geschwindigkeitsunterschied für Segmentierung in m/s	Anzahl Segmente	Anzahl Muster	Segment-Abdeckung	Punkt-Abdeckung	Durchschnittliche Distanz zwischen Mustern in mm
8,0	2579	50	0,58	**0,59**	6400
8,2	2581	52	0,58	**0,59**	6810
8,4	2557	54	0,57	**0,59**	7079
8,6	2570	54	0,57	**0,59**	7194
8,8	2563	50	0,57	0,58	6959
9,0	2557	51	0,57	0,58	7493
9,2	2540	53	0,57	0,58	6935

Zu beachten ist, dass diese Parameter eventuell nicht direkt auf Datensätze anderer Umgebungen übertragen werden können. Hierzu folgt eine tiefere Analyse in der Diskussion.

Es ist zu erkennen, dass die maximale Abdeckung aller Punkte bei ca. 60% liegt. Dies lässt sich als ein gut akzeptabler Wert bezeichnen, da er sicherstellt, dass ausreichend Muster für mobile Dienste gefunden werden.

Zudem wurden viele Punkte durch die Aufbereitung aus Mustern entfernt. Um eine höhere Punkt-Abdeckung zu erreichen, wurde eine andere Aufbereitung durchgeführt. Während bei der in dieser Arbeit vorgestellten Aufbereitung 21477 Punkte in Mustern

Tabelle 6.3.: Variieren des Parameters *Reichweite* zur Maximierung der Punkt-Abdeckung

Reichweite, über die der Richtungsunterschied bestehen muss in Punkten	Anzahl Segmente	Anzahl Muster	Segment-Abdeckung	Punkt-Abdeckung	Durchschnittliche Distanz zwischen Mustern in mm
2	3579	42	0,41	0,54	6299
3	3257	45	0,46	0,55	6478
4	3015	48	0,49	0,56	6881
5	2826	51	0,53	0,58	6988
6	2698	52	0,55	**0,59**	6570
7	2557	54	0,57	**0,59**	7079
8	2436	52	0,59	**0,59**	7152
9	2333	54	0,60	**0,59**	7207
10	2243	54	0,60	0,58	6736

enthalten waren, erreichte die Aufbereitung hier 28194 Punkte. Nach der Segmentierung enthielten die Daten insgesamt 36366 Punkte, sodass die Abdeckung von etwa 59% auf etwa 77,5% gesteigert werden konnte. Bei der Aufbereitung im Versuch wurden zudem weiterhin Muster bzw. unterstützende Segmente entfernt, die nach Definition nicht als solche bezeichnet werden konnten. Dieser Versuch zeigt, dass eine weitaus größere Punkt-Abdeckung mit einer anderen Aufbereitung der Muster möglich wäre. Das Ziel dieser Arbeit ist jedoch, vorhandene Muster zu maximieren, um den größtmöglichen Nutzen für mobile Dienste aus ihnen zu ziehen. Ist ein Nutzer hingegen daran interessiert, möglichst viele Daten durch ein Muster abdecken zu können, ist dieses Verfahren durch eine Anpassung der Aufbereitung auch hierzu in der Lage.

Zu beachten ist, dass die gefundenen Werte für die Parameter nicht dem tatsächlichen Optimum entsprechen. Dieses liegt in der Nähe der gefundenen Parameter und kann durch zusätzliche Dezimalstellen der Werte immer weiter angenähert werden. Die maximale Abdeckung ändert sich dadurch marginal.

Des Weiteren ist zu erkennen, dass auch die durchschnittliche Distanz zwischen den Mustern für die gefundenen Parameter ansteigt. Dies ist von Wichtigkeit für die Detektion unterschiedlicher Muster, damit verschiedene Intentionen abgedeckt werden können. Wenn sich die Muster zu stark ähneln, müssen allen die gleiche Intention zugeordnet werden, sodass kein Gewinn für einen mobilen Dienst erzielt werden kann.

In Abbildung A.1 im Anhang sind alle gefundenen Mustern mit den in diesem Unterabschnitt erarbeiteten Parametern dargestellt. Zudem stellen die Abbildungen A.2 bis A.7 beispielhaft Muster und ihre zugehörigen Segmente im Ausgangsdatensatz dar.

6.3.2. Maximierung der Segment-Abdeckung

In diesem Unterabschnitt werden die Ergebnisse der Evaluation dargestellt, die eine Maximierung der Segment-Abdeckung bezweckte. Hierfür wurden die Parameter für die Segmentierung einzeln nacheinander variiert, während die anderen Parameter des Verfahrens festgehalten wurden. Die folgenden Parameter gelten also für den gesamten Unterabschnitt:

- Maximaler Richtungsunterschied für Sammeln: 0,4 π
- Maximale Distanzabweichung für Sammeln: 1 m
- Mindestlänge des Musters: 6 Punkte
- Zeittoleranz: 4 Punkte $\hat{=}$ 2 s
- Mindest-Unterstützung: 2 Segmente

Wie im Unterabschnitt zuvor wurden nun in den Tabellen 6.4, 6.5 und 6.6 die einzelnen Parameter der Segmentierung variiert. Dieses Mal wurde darauf abgezielt, die maximale Segment-Abdeckung zu erreichen. Da jeweils nur ein Parameter variiert wurde, mussten die anderen Parameter der Segmentierung mit festen Werten versehen werden. Hierfür wurden jene Werte gewählt, die sich als optimal herausstellten. Dadurch ist besser ersichtlich, wie hoch die maximale Segment-Abdeckung liegt und wie sehr sie durch die einzelnen Parameter beeinflusst wird, wenn die anderen Parameter optimal sind. Für diesen Teil der Evaluation wurde mit Hilfe eines automatisierten Tests überprüft, ob die optimalen Parameter erreicht wurden.

Die folgenden Werte wurden im Rahmen der Evaluation für die Maximierung der Segment-Abdeckung gefunden:

- Maximaler Richtungsunterschied für Segmentierung: 0,49 π
- Maximaler Geschwindigkeitsunterschied für Segmentierung : 8,5 m/s
- Reichweite, über die der Richtungsunterschied bestehen muss: 7 Punkte

Wie zu erkennen ist, ähneln die gefundenen Werte denen aus dem Unterabschnitt zuvor. Der maximale Richtungsunterschied bewirkt, dass entweder die maximale Punkt-Abdeckung oder die maximale Segment-Abdeckung erreicht wird.

Die maximal zu erreichende Segment-Abdeckung liegt - wie in den entsprechenden Tabellen zu erkennen ist - bei rund 71%. Dieser Wert bedeutet, dass mit diesem Parametersatz besonders aussagekräftige Muster gefunden werden, denen klare Intentionen zugeordnet werden können. Jedoch werden diese aussagekräftigen Muster bei gleichzeitiger Abnahme der Punkt-Abdeckung des Datensatzes gefunden. Diese sinkt nämlich auf rund 43% - mit der Folge, dass mobile Dienste seltener, aber dafür aussagekräftigere Kontextinformationen erhalten.

Die durchschnittliche Distanz zwischen den gefundenen Mustern liegt in diesem Unterabschnitt hoch genug, um von unterschiedlichen Mustern zu sprechen. Sie beträgt rund 7,5 m, was auf einem Fußballplatz eine große Distanz ist. Es ist also sichergestellt, dass mit den angegebenen Werten für die Parameter tatsächlich unterschiedli-

Tabelle 6.4.: Variieren des Parameters *Maximaler Richtungsunterschied für Segmentierung* zur Maximierung der Segment-Abdeckung

Maximaler Richtungs-unter-schied für Segmentie-rung in π	Anzahl Segmente	Anzahl Muster	Segment-Abdeckung	Punkt-Abdeckung	Durch-schnittliche Distanz zwischen Mustern in mm
0,40	1552	74	0,69	0,52	7546
0,43	1381	67	0,69	0,46	7316
0,46	1224	73	0,70	0,45	7507
0,49	1106	75	**0,71**	0,43	7669
0,52	988	75	0,70	0,40	8438
0,55	896	77	0,70	0,39	8735
0,58	787	68	0,67	0,33	7614

Tabelle 6.5.: Variieren des Parameters *Maximaler Geschwindigkeitsunterschied für Segmentierung* zur Maximierung der Segment-Abdeckung

Maximaler Geschwin-digkeitsun-terschied für Seg-mentierung in m/s	Anzahl Segmente	Anzahl Muster	Segment-Abdeckung	Punkt-Abdeckung	Durch-schnittliche Distanz zwischen Mustern in mm
8,2	1150	68	0,70	0,43	7033
8,3	1136	73	0,70	0,43	7255
8,4	1118	75	0,70	0,43	7394
8,5	1106	75	**0,71**	0,43	7669
8,6	1095	74	0,70	0,42	7495
8,7	1077	74	0,70	0,42	7870
8,8	1069	78	**0,71**	0,43	7957
8,9	1064	75	0,70	0,43	8046

che, aussagekräftige Muster gefunden werden.

In Abbildung B.1 im Anhang sind alle gefundenen Muster dieses Unterabschnitts dargestellt. Des Weiteren sind in den Abbildungen B.2 bis B.6 einige auswählte Beispiele für gefundene Muster und ihre zugehörigen unterstützenden Segmente zu finden.

Welcher Parametersatz für die Segmentierung zu bevorzugen ist, hängt von dem Einsatzgebiet des mobilen Diensts ab. Ist das Ziel, möglichst viele Kontextinformationen

Tabelle 6.6.: Variieren des Parameters *Reichweite* zur Maximierung der Segment-Abdeckung

Reichweite, über die der Richtungsunterschied bestehen muss in Punkten	Anzahl Segmente	Anzahl Muster	Segment-Abdeckung	Punkt-Abdeckung	Durchschnittliche Distanz zwischen Mustern in mm
2	1745	73	0,54	0,46	7559
3	1515	74	0,60	0,45	7632
4	1384	73	0,65	0,45	6985
5	1258	76	0,68	0,45	7358
6	1172	76	0,70	0,43	7598
7	1106	75	**0,71**	0,43	7669
8	1039	74	0,70	0,41	7177
9	984	74	0,69	0,40	7497

zu sammeln, die aber unter Umständen weniger aussagekräftig sind, sollten die Werte aus dem Unterabschnitt 6.3.1 genutzt werden. Zielt der Nutzer jedoch darauf ab, aussagekräftigere Informationen zu bekommen, die dafür seltener bereitgestellt werden, sollte der Parametersatz aus diesem Unterabschnitt gewählt werden. Wie zu erkennen ist, hängt die Wahl der Parameter also stark vom jeweiligen Umfeld ab, worauf in der Diskussion detaillierter eingegangen wird.

6.3.3. Parameter für das Sammeln ähnlicher Segmente

In diesem Unterabschnitt werden die Ergebnisse der Evaluation vorgestellt, die eine Maximierung der Punkt- und Segment-Abdeckung durch die Parameter für das Sammeln ähnlicher Segmente bezweckte. Hierfür wurden die Werte für die Parameter der Segmentierung festgehalten. Des Weiteren wurde auch die Mindest-Unterstützung festgehalten, da im Rahmen dieser Arbeit eine Bewegung als Muster gilt, wenn sie in mindestens zwei Segmenten enthalten ist. Als Parameter für die Segmentierung wurden jene Werte gewählt, die eine Maximierung der Punkt-Abdeckung bewirken. Der Grund hierfür ist, dass die Punkt-Abdeckung im Rahmen dieser Arbeit als wichtiger bewertet wird. Dieselben Ergebnisse können jedoch auch erzielt werden, wenn jene Werte gewählt werden, die zu einer Maximierung der Segment-Abdeckung führen. Die Werte für die festgehaltenen Parameter lauten:

- Maximaler Geschwindigkeitsunterschied für Segmentierung: 8,5 m/s
- Maximaler Richtungsunterschied für Segmentierung: 0,29 π
- Reichweite, über die der Richtungsunterschied bestehen muss: 7 Punkte

- Mindest-Unterstützung: 2 Segmente

In den Tabellen 6.7, 6.8, 6.9 und 6.10 sind die Ergebnisse für die unterschiedlichen Parameter für das Sammeln der Muster zu finden. Den Parametern, die in der entsprechenden Tabelle nicht variiert wurden, wurde der Wert zugewiesen, der sich als optimal herausstellte. Die optimalen Werte für die vier Parameter lauten dabei wie folgt:

- Maximaler Richtungsunterschied für Sammeln: 0,4 π
- Maximale Distanzabweichung für Sammeln: 1 m
- Mindestlänge des Musters: 6 Punkte
- Zeittoleranz: 4 Punkte $\hat{=}$ 2 s

Tabelle 6.7.: Variieren des Parameters *Maximaler Richtungsunterschied für Sammeln* zur Maximierung der Abdeckung

Maximaler Richtungs-unter-schied für Sammeln in π	Anzahl Segmente	Anzahl Muster	Segment-Abdeckung	Punkt-Abdeckung	Durch-schnittliche Distanz zwischen Mustern in mm
0,10	2572	110	0,51	0,47	4840
0,15	2572	95	0,54	0,51	5828
0,20	2572	84	0,55	0,54	5770
0,25	2572	72	0,56	0,56	6405
0,30	2572	64	0,57	0,57	6749
0,35	2572	50	0,57	0,57	6592
0,40	2572	50	**0,57**	**0,58**	6981

Es ist ersichtlich, dass je größer die Parameter *Maximaler Richtungsunterschied für Sammeln*, *Maximale Distanzabweichung für Sammeln*, *Zeittoleranz* gewählt werden und je kleiner der Parameter *Mindestlänge des Musters* festgelegt wird, desto mehr Muster gefunden werden und desto größer die Abdeckung des Datensatzes durch die Muster ist. Dies geschieht jedoch zu Lasten der Muster, die ihren zugehörigen unterstützenden Segmenten immer weniger ähneln. Dies wird im Laufe dieses Unterabschnitts noch an mehreren Beispielen verdeutlicht.

Für den maximalen Richtungsunterschied ist zu erkennen, dass sich die Muster bei der Wahl eines kleinen Wertes für diesen Parameter stärker ähneln als bei der Wahl eines großen Wertes. Dies lässt sich darauf zurückführen, dass für kleine Werte unterschiedliche Muster gefunden werden, die für größere Werte in einem Muster zusammengefasst werden, da eine größere Abweichung innerhalb der Muster erlaubt ist. Diese Beobachtung entspricht also den Erwartungen. Auch dass die Abdeckung der Daten durch einen höheren Wert steigt, entspricht den Erwartungen. Dies ist bedingt dadurch, dass mehr Muster gefunden werden, wenn sich die zugehörigen Segmente

Tabelle 6.8.: Variieren des Parameters *Maximale Distanzabweichung für Sammeln* zur Maximierung der Abdeckung

Maximale Distanzabweichung für Sammeln in mm	Anzahl Segmente	Anzahl Muster	Segment-Abdeckung	Punkt-Abdeckung	Durchschnittliche Distanz zwischen Mustern in mm
300	2572	132	0,50	0,46	4584
400	2572	112	0,52	0,49	4913
500	2572	93	0,54	0,52	5481
600	2572	86	0,55	0,54	5739
700	2572	70	0,56	0,55	6139
800	2572	63	0,56	0,57	6635
900	2572	52	0,56	0,57	6507
1000	2572	50	**0,57**	**0,58**	6981

Tabelle 6.9.: Variieren des Parameters *Mindestlänge des Musters* zur Maximierung der Abdeckung

Mindestlänge des Musters in Punkten	Anzahl Segmente	Anzahl Muster	Segment-Abdeckung	Punkt-Abdeckung	Durchschnittliche Distanz zwischen Mustern in mm
6	2572	50	**0,57**	**0,58**	6981
10	2572	48	0,42	0,53	7090
14	2572	42	0,30	0,47	7199
18	2572	39	0,22	0,42	7232
22	2572	35	0,15	0,36	7415

weniger ähneln müssen, um als Muster wahrgenommen zu werden.

Analog gilt diese Argumentation für die maximale Distanzabweichung. Je größer der eingegebene Wert, desto mehr Muster werden gefunden, die sich untereinander weniger ähneln, aber die Ausgangsdaten besser abdecken.

Für die maximale Zeittoleranz lassen sich dieselben Beobachtungen anstellen, wie für die beiden vorangegangenen Parameter. Je mehr Zeit zwischen zwei Punkten eines Musters vergehen darf, desto weniger müssen sich Segmente ähneln, um als Muster gefunden zu werden. Auch die entsprechende Distanz zwischen den Mustern ist zu erwarten, da sich Muster durch eine größere Zeittoleranz immer weniger ähneln.

Tabelle 6.10.: Variieren des Parameters *Zeittoleranz* zur Maximierung der Abdeckung

Zeit-toleranz in Punkten	Anzahl Segmente	Anzahl Muster	Segment-Abdeckung	Punkt-Abdeckung	Durch-schnittliche Distanz zwischen Mustern in mm
0	2572	112	0,47	0,43	4801
1	2572	85	0,53	0,51	5614
2	2572	70	0,56	0,55	5912
3	2572	56	0,56	0,57	6409
4	2572	50	**0,57**	**0,58**	6981

Für die Mindestlänge eine Musters lässt sich hingegen feststellen, dass ein niedriger Wert besser für die Abdeckung der Daten ist. Dies liegt darin begründet, dass deutlich mehr kürzere als längere Muster gefunden werden können, die somit durch ihre Anzahl auch die Abdeckung der Daten steigern. Dies geschieht jedoch in diesem Fall bei gleichzeitiger Abnahme der Ähnlichkeit der Muster. Je kürzer die Mindestlänge für Muster gewählt wird, desto mehr ähneln sich die gefundenen. Die durchschnittliche Distanz sinkt dabei jedoch nicht so signifikant ab, dass sie ein Problem darstellt, da die Muster immer noch einen Abstand von rund 7 m aufweisen. Begründen lässt sich diese Beobachtung damit, dass mit der Senkung der Mindestlänge eines Musters tendenziell kürzere Muster gefunden werden, die einander stärker ähneln, als wenige, lange Muster.

Nach subjektivem menschlichem Empfinden werden Muster als solche wahrgenommen, um ihnen Intentionen zuordnen zu können. Werden die Parameter beliebig erhöht bzw. wird die Mindestlänge beliebig gesenkt, entstehen Muster, deren Segmente ein Mensch nicht mehr als ähnlich empfinden würde. Um dies zu verhindern, muss also eine Grenze gesetzt werden, an der nach subjektivem Empfinden gerade noch akzeptable Muster gefunden werden. Um zu verdeutlichen, warum die Grenze auf ihren Wert festgelegt wurde, befinden sich im Anhang die Abbildungen C.1 bis C.5, die ein Negativbeispiel enthalten. Dieses stellt die Konsequenzen einer leichten Erhöhung der Parameter bzw. Verringerung der Mindestlänge anschaulich dar. Dazu wird in der Abbildung C.1 zunächst ein Überblick gegeben. Anschließend zeigen die Abbildungen C.2 bis C.5 einige ausgewählte Muster, die der Meinung des Autors nach nicht mehr als solche bezeichnet werden können, da sie zum Beispiel zu kurz (siehe Abbildung C.4) oder zu unterschiedlich sind (siehe beispielsweise Abbildung C.5).

6.3.4. Qualitative Analyse der gefundenen Muster

Für eine qualitative Analyse der gefundenen Muster wurden alle Muster betrachtet und - soweit möglich - einer Intention zugeordnet. Hierbei ist zu beachten, dass bei

einem Fußballspiel weniger verschiedene Intentionen als beispielsweise in der Stadt vorhanden sind. Somit lässt sich dieser Teil der Evaluation nicht eindeutig auf andere Umgebungen übertragen. Er gibt jedoch einen guten Überblick, ob gefundene Muster mit Intentionen verbunden werden können. Des Weiteren ist eine Zuordnung von Intentionen zu Bewegungsmustern subjektiv, sodass von unterschiedlichen Menschen verschiedene Ergebnisse ermittelt werden können. Die Auswertung hierzu befindet sich in Tabelle D.1.

Von 54 gefundenen Mustern konnten 30 mit einer Intention versehen werden, während bei den anderen 24 Mustern keine eindeutige Intention zu erkennen war. Dies bedeutet eine Abdeckung von etwa 56% der Muster. Die zugeordneten Intentionen basieren allein auf den Bewegungsmustern und der Kenntnis, dass die Daten von einem Fußballspiel stammen. Wird zusätzlich noch der Ort berücksichtigt, an dem das Bewegungsmuster gefunden wurde, können weitere mit Intentionen versehen werden. Insgesamt bedeutet ein Wert von 56%, dass aus den Bewegungsmustern ein Gewinn für mobile Dienste gezogen werden kann. Jedoch reicht diese Abdeckung nicht, um in der Praxis einen **zuverlässigen** mobilen Dienst nur auf Bewegungsmustern basierend aufzubauen, da nur der Hälfte aller Bewegungen eine Intention und somit eine Anwendung zugewiesen werden kann. Stattdessen müssen weitere Kontextinformationen gesammelt und zusätzlich verwendet werden, um dem Nutzer stets die richtige Anwendung anbieten zu können.

6.4. Diskussion

Wie bereits in den Unterkapiteln zuvor angemerkt, lassen sich nicht alle gefundenen Werte für Parameter direkt auf jede Umgebung übertragen, in der mobile Dienste eingesetzt werden können. Für alle in dieser Arbeit präsentierten Ergebnisse muss stets beachtet werden, dass zur Evaluation Fußballdaten genutzt wurden. In einem anderen Umfeld - beispielsweise in der Stadt - werden abweichende Werte für die Parameter benötigt, um die gewünschten Muster zu finden. Dadurch, dass für jede Umgebung andere Werte für die Parameter gewählt werden können, kann das vorgestellte Verfahren individuell an die einzelnen Umfelder angepasst werden.

Die folgenden Eigenschaften der Bewegung eines Fußballspielers wurden in den Parametern besonders berücksichtigt und geben somit einen Rückschluss auf die vorgestellten Werte:

- viele Bewegungen: Spieler verweilen selten über eine längere Zeit am selben Platz
- schneller Wechsel zwischen Geschwindigkeiten
- abrupte Richtungswechsel

Diese Eigenschaften sind offensichtlich sehr spezifisch für Fußballspieler und erfordern insbesondere eine spezielle Segmentierung. In anderen Umfeldern müssen die Parameter auf die Eigenschaften dieses Umfelds angepasst werden. In der Stadt kann

dies zum Beispiel beinhalten, dass sich Menschen eher langsam bewegen und häufiger stehen bleiben.

Des Weiteren wurden in den Unterabschnitten 6.3.1 und 6.3.2 zwei verschiedene Segmentierungsprofile vorgestellt. Das erste Profil zielt darauf ab, möglichst oft ein Muster zu finden und viele Daten abzudecken. Das Ziel des zweiten Profils hingegen ist es, längere Segmente zu finden und somit eine genauere Vorhersage treffen zu können. Hierfür wird die Segment-Abdeckung maximiert, da dadurch weniger unbedeutende Segmente vorhanden sind. Auch hier hängt die Wahl des Segmentierungsprofils von der Domäne, in der es genutzt werden soll, ab. Ist das Ziel des Nutzers, den mobilen Dienst oft mit Mustern zu versorgen, sollte das erste Profil gewählt werden. Dieses stellt mehr Kontextinformationen bereit, die aber in der Regel weniger wert sind, da sie weniger aussagekräftig sind. Ist das Ziel jedoch, sehr aussagekräftige Informationen zu sammeln, die dafür seltener gefunden werden, ist das zweite Profil zu bevorzugen.

Insgesamt lässt sich feststellen, dass das erarbeitete Verfahren sehr domänenabhängig ist. Durch viele Parameter können in jedem beliebigem Umfeld passende Muster gefunden werden, die ein mobiler Dienst mit Hilfe anderer Informationen verwerten kann. Somit ist das Verfahren sehr anpassungsfähig und lässt sich in beinahe jedem Umfeld gewinnbringend nutzen.

7. Zusammenfassung und Ausblick

In diesem Kapitel werden die Erkenntnisse dieser Arbeit zusammengefasst und ein kurzes Fazit gezogen. Anschließend wird ein Ausblick gegeben, inwiefern das vorgestellte Verfahren in Zukunft verwendet und erweitert werden kann.

7.1. Zusammenfassung

Das Ziel dieser Arbeit war es, Bewegungsmuster zu sammeln und diese mit Intentionen zu verknüpfen, um mobile Dienste mit zusätzlichen Kontextinformationen zu versorgen. Außerdem sollte mit Hilfe von Explorationsarten gezeigt werden, wie mobile Dienste direkt von diesen Kontextinformationen profitieren können. Insgesamt teilt sich die Arbeit also in zwei Teile auf. Der erste Teil ist hierbei die Untersuchung und Verbesserung mobiler Dienste, während der zweite Teil sich mit der Trajektorie-Analyse beschäftigt.

Zunächst wurde in den Grundlagen gezeigt, dass mobile Dienste stark vom Kontext profitieren können. Außerdem wurde vorgestellt, inwiefern diese Arbeit dazu beiträgt, mobile Dienste zu verbessern. Dies geschieht durch das Sammeln zusätzlicher Kontextinformationen in Form von Bewegungsmustern. Als Ergebnis konnte hierbei festgehalten werden, dass mobile Dienste das Potential haben, das durch weiteren Kontext immer besser ausgereizt werden kann. In den Grundlagen zur Trajektorie-Analyse wurden schließlich Definitionen gegeben, die das Problem der Erkennung von Bewegungsmustern in Trajektorien beschreiben. Hierbei wurde festgestellt, dass diverse Parameter benötigt werden, um das Verfahren zu realisieren.

Im dritten Kapitel der Arbeit wurden verwandte Arbeiten betrachtet und zu dieser Arbeit abgegrenzt. Viele Arbeiten sammeln Bewegungsmuster in Form von einer Aneinanderreihung von Standorten, ohne feine Bewegungsmuster zu finden. Zum Sammeln anderer Kontextinformationen gibt es diverse andere Arbeiten, die mit dieser Arbeit kombiniert werden können, um die Erkennung von Intentionen weiter zu verbessern.

Basierend auf den Grundlagen wurden in den beiden folgenden Kapiteln die Konzepte dieser Arbeit vorgestellt. Hierbei wurde festgestellt, dass ein Zusammenhang zwischen Intention und Bewegung vorhanden ist, der jedoch nicht immer eindeutig zu erkennen ist. Es müssen also weitere Kontextinformationen zu Rate gezogen werden, um einer Bewegung stets eine eindeutige Intention zuordnen zu können. Als Explorationsarten wurden zunächst die geometrischen Formen Dreieck, Ellipse und Kreis vorgestellt, die ausreichen, bis weitere Untersuchungen und Versuche zu diesem Thema angestellt wurden. Außerdem wurde bei einem Vergleich zwischen den beiden Ansätzen Bottom-up und Top-down festgestellt, dass der Bottom-up Ansatz für diese Arbeit zu

bevorzugen ist, da hier der Aufwand geringer und die Zuverlässigkeit höher ist.

In dem Teil für die Trajektorie-Analyse wurde das erarbeitete Verfahren vorgestellt und erklärt. Dazu wurde es in unterschiedliche Phasen unterteilt, für die nacheinander geklärt wurde, wie sie umgesetzt wurden.

Im sechsten Kapitel der Arbeit wurde schließlich das erarbeitete Verfahren evaluiert, gefolgt von einer Vorstellung der Ergebnisse. Dabei wurde festgestellt, dass die erreichte Datenabdeckung von ca. 60% ausreicht, um einen Nutzen für mobile Dienste aus dem Verfahren ziehen zu können. In der darauf folgenden Diskussion wurde erklärt, dass die Wahl der Parameter stets an das entsprechende Umfeld angepasst werden können. Hieraus ergab sich, dass das erarbeitete Verfahren sehr anpassungsfähig ist.

7.2. Ausblick

Zunächst muss das Verfahren optimiert werden, um es auch auf Tablets oder Smartphones einsetzbar zu machen. Momentan ist es noch zu rechenintensiv dafür, da jedes Segment mit jedem verglichen wird. Hierzu könnte das Verfahren dahingehend verbessert werden, dass nur noch solche Segmente verglichen werden, die tatsächlich ähnlich zueinander sein könnten. Dafür könnte die Länge der Segmente betrachtet werden und in welche Quadranten sie sich ausdehnen.

Um den gefundenen Bewegungsmustern später tatsächlich die richtigen Intentionen zuordnen zu können, müssen in einem nächsten Schritt Bewegungsdaten von potenziellen Nutzern gesammelt werden. Zusätzlich muss mit Fragebögen herausgefunden werden, welchen Dienst sie in welchen Momenten gebraucht haben. Alternativ können diese Informationen mit einer zusätzlichen Anwendung auf dem Smartphone gesammelt werden.

Ist das Verfahren so weit optimiert, dass es auf mobilen Geräten eingesetzt werden kann, muss ein mobiler Dienst entwickelt werden, der neben den Kontextinformationen, die durch die Bewegungsmuster gesammelt werden, noch weitere Informationen nutzt, um Anwendungen für einen Nutzer auszuwählen. Anschließend muss eine Studie durchgeführt werden, ob der mobile Dienst tatsächlich wie gewünscht funktioniert.

Nachdem ein mobiler Dienst etabliert wurde, der in der Lage ist selbstständig die richtigen Anwendungen für Nutzer auszuwählen, kann dieser stetig mit zusätzlichen Kontextinformationen erweitert werden. Hierdurch wird eine immer bessere Voraussage, welche Anwendung der mobile Nutzer als nächstes benötigt, ermöglicht.

Literaturverzeichnis

[1] ComScore, “Anzahl der Smartphone-Nutzer in Deutschland in den Jahren 2009 bis 2014,” 2014. [Online]. Available: http://de.statista.com/statistik/daten/studie/198959/umfrage/anzahl-der-smartphonenutzer-in-deutschland-seit-2010/

[2] T. u. n. M. e. B. Bundesverband Informationswirtschaft, “Anzahl der Downloads mobiler Apps in Deutschland in den Jahren 2009 bis 2012 und Prognose für 2014,” 2014. [Online]. Available: http://de.statista.com/statistik/daten/studie/168038/umfrage/anzahl-der-downloads-mobiler-apps-in-deutschland-seit-2009/

[3] M. Hettig, “Grafisches Werkzeug zur Bereitstellung von kontextbezogenen Diensten für mobile Nutzer,” Bachelorarbeit, Leibniz Universität Hannover, Fachgebiet Software Engineering, 2012.

[4] A. K. Dey, “Understanding and Using Context,” *Personal and Ubiquitous Computing*, vol. 5, no. 1, February 2001, pp. 4–7, 2001.

[5] K. Rothermel, “Kontextbezogene Systeme – die Welt im Computer modelliert,” *Digitale Visionen*, pp. 31–42, 2008.

[6] A. K. Dey and G. D. Abowd, “Towards a Better Understanding of Context and Context-Awareness,” *Computing Systems*, vol. 40, no. 3, pp. 304–307, 1999.

[7] A. Rakotonirainy, S. Loke, and G. Fitzpatrick, “Context-awareness for the mobile environment,” in *CHI2000 Workshop #11 Proposal*, The Hague, Netherlands, 2000.

[8] S. Long, R. Kooper, G. D. Abowd, and C. G. Atkeson, “Rapid Prototyping of Mobile Context-aware Applications: The Cyberguide Case Study,” in *Proceedings of the 2nd Annual International Conference on Mobile Computing and Networking*, 1996, pp. 97–107.

[9] M. Li, Y. Guo, and Y. Huang, “A Study on the Factors Affecting the Use of Mobile Travel Guide Systems During Travel,” in *The 18th Annual Graduate Education and Graduate Student Research Conference in Hospitality and Tourism*, 2013, pp. 49–54.

[10] Google, “Google Maps,” 2014. [Online]. Available: https://www.google.de/maps/preview

[11] J. Gummerus and M. Pihlström, “Context and mobile services value-in-use,” *Journal of Retailing and Consumer Services*, vol. 18, no. 6, pp. 521–533, Nov. 2011.

[12] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, “Trajectory pattern mining,” in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’07*. New York, New York, USA: ACM Press, 2007, pp. 330–339.

[13] M. Berchtold, M. Budde, D. Gordon, H. R. Schmidtke, and M. Beigl, “ActiServ: Activity Recognition Service for mobile phones,” in *International Symposium on Wearable Computers (ISWC) 2010*. IEEE, Oct. 2010, pp. 1–8.

[14] M. Budde, M. Berchtold, and M. Beigl, “Video : Activity Recognition on Mobile Phones – Why do we need it and how can it be done?” San Francisco, USA, 2011. [Online]. Available: https://www.youtube.com/watch?v=7QMH29_y6yw

[15] R. Mayrhofer, H. Radi, A. Ferscha, J. Kepler, and U. Linz, “Recognizing and Predicting Context by Learning from User Behavior,” in *The International Conference On Advances in Mobile Multimedia (MoMM2003)*, 2003, vol. 171, pp. 25–35.

[16] D. Gordon, J. Czerny, and M. Beigl, “Activity recognition for creatures of habit,” *Personal and Ubiquitous Computing*, vol. 18, no. 1, pp. 205–221, Mar. 2013.

[17] D. Ashbrook and T. Starner, “Using GPS to learn significant locations and predict movement across multiple users,” *Personal and Ubiquitous Computing*, vol. 7, no. 5, pp. 275–286, Oct. 2003.

[18] K. Laasonen, “Clustering and prediction of mobile user routes from cellular data,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3721 LNAI, 2005, pp. 569–576.

[19] D. Taniar and J. Goh, “On Mining Movement Pattern from Mobile Users,” *International Journal of Distributed Sensor Networks*, vol. 3, no. 1, pp. 69–86, 2007.

[20] G. Yavaş, D. Katsaros, O. Ulusoy, and Y. Manolopoulos, “A data mining approach for location prediction in mobile environments,” *Data & Knowledge Engineering*, vol. 54, no. 2, pp. 121–146, Aug. 2005.

[21] J. J.-C. Ying, W.-C. Lee, T.-C. Weng, and V. S. Tseng, “Semantic trajectory mining for location prediction,” in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS ’11*. New York, New York, USA: ACM Press, 2011, p. 34.

[22] V. S. Tseng and K. W. Lin, “Efficient mining and prediction of user behavior patterns in mobile web systems,” *Information and Software Technology*, vol. 48, no. 6, pp. 357–369, Jun. 2006.

[23] A. Schmidt, “Ubiquitous Computing – Computing in Context,” Dissertation, Lancaster University, England, U.K., 2002.

[24] OpenStreetMap, “Karte Hannover,” 2014. [Online]. Available: http://www.openstreetmap.de/karte.html

[25] D. Darmon, J. Sylvester, M. Girvan, and W. Rand, “Predictability of User Behavior in Social Media: Bottom-Up v. Top-Down Modeling,” in *2013 International Conference on Social Computing*. Alexandria, VA: IEEE, Sep. 2013, pp. 102–107.

[26] G. Linden, B. Smith, and J. York, “Amazon.com recommendations: item-to-item collaborative filtering,” *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, Jan. 2003.

[27] M. Buchin, A. Driemel, M. Van Kreveld, and V. Sacristan, “Segmenting trajectories: A framework and algorithms using spatiotemporal criteria,” *Journal of Spatial Information Science*, vol. 3, no. 3, pp. 33–63, Dec. 2011.

[28] S. B. Needleman and C. D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–453, Mar. 1970.

Abbildungsverzeichnis

1.1. Anzahl der Smartphone-Nutzer in Deutschland[1] 1
1.2. Anzahl der Downloads mobiler Apps in Deutschland[2] 2

2.1. Google Maps Navigation-Auswahlmaske 6

4.1. Explorationsart 1: spitzes Dreieck 14
4.2. Explorationsart 2: Ellipse 15
4.3. Explorationsart 3: Kreis 16

5.1. Übersicht über das Verfahren 19
5.2. Segmentierung nach Geschwindigkeit 21
5.3. Segmentierung nach Richtung 22
5.4. Umwandlung in eine orts- und richtungsunabhängige Form 24
5.5. Länge der Übereinstimmung zweier Trajektorien 25

A.1. Alle Muster bei optimalen Parametern für eine maximale Punkt-Abdeckung 51
A.2. Beispiel 1 bei optimalen Parametern für eine maximale Punkt-Abdeckung . 52
A.3. Beispiel 2 bei optimalen Parametern für eine maximale Punkt-Abdeckung . 52
A.4. Beispiel 3 bei optimalen Parametern für eine maximale Punkt-Abdeckung . 53
A.5. Beispiel 4 bei optimalen Parametern für eine maximale Punkt-Abdeckung . 53
A.6. Beispiel 5 bei optimalen Parametern für eine maximale Punkt-Abdeckung . 54
A.7. Beispiel 6 bei optimalen Parametern für eine maximale Punkt-Abdeckung . 54

B.1. Alle Muster bei optimalen Parametern für eine maximale Segment-Abdeckung 55
B.2. Beispiel 1 bei optimalen Parametern für eine maximale Segment-Abdeckung 56
B.3. Beispiel 2 bei optimalen Parametern für eine maximale Segment-Abdeckung 56
B.4. Beispiel 3 bei optimalen Parametern für eine maximale Segment-Abdeckung 57
B.5. Beispiel 4 bei optimalen Parametern für eine maximale Segment-Abdeckung 57
B.6. Beispiel 5 bei optimalen Parametern für eine maximale Segment-Abdeckung 58

C.1. Negativbeispiel: Alle Muster bei leicht überhöhten Werten bzw. leicht verringerter Mindestlänge 59
C.2. Negativbeispiel: Beispiel 1 bei leicht überhöhten Werten bzw. leicht verringerter Mindestlänge 60
C.3. Negativbeispiel: Beispiel 2 bei leicht überhöhten Werten bzw. leicht verringerter Mindestlänge 60
C.4. Negativbeispiel: Beispiel 3 bei leicht überhöhten Werten bzw. leicht verringerter Mindestlänge 61
C.5. Negativbeispiel: Beispiel 4 bei leicht überhöhten Werten bzw. leicht verringerter Mindestlänge 61

Tabellenverzeichnis

6.1. Variieren des Parameters *Maximaler Richtungsunterschied für Segmentierung* zur Maximierung der Punkt-Abdeckung 33
6.2. Variieren des Parameters *Maximaler Geschwindigkeitsunterschied für Segmentierung* zur Maximierung der Punkt-Abdeckung 33
6.3. Variieren des Parameters *Reichweite* zur Maximierung der Punkt-Abdeckung 34
6.4. Variieren des Parameters *Maximaler Richtungsunterschied für Segmentierung* zur Maximierung der Segment-Abdeckung 36
6.5. Variieren des Parameters *Maximaler Geschwindigkeitsunterschied für Segmentierung* zur Maximierung der Segment-Abdeckung 36
6.6. Variieren des Parameters *Reichweite* zur Maximierung der Segment-Abdeckung 37
6.7. Variieren des Parameters *Maximaler Richtungsunterschied für Sammeln* zur Maximierung der Abdeckung 38
6.8. Variieren des Parameters *Maximale Distanzabweichung für Sammeln* zur Maximierung der Abdeckung 39
6.9. Variieren des Parameters *Mindestlänge des Musters* zur Maximierung der Abdeckung 39
6.10. Variieren des Parameters *Zeittoleranz* zur Maximierung der Abdeckung . . 40

D.1. Zuordnung von Intention zu den Mustern, die mit optimalen Parametern für eine Maximierung der Punkt-Abdeckung gefunden wurden 62

Liste der Algorithmen

5.1. Segmentierung nach Geschwindigkeit (angelehnt an[27]) 21
5.2. Segmentierung nach Richtung (angelehnt an[27]) 23
5.3. Länge der Übereinstimmung zweier Trajektorien (angelehnt an [28]) . . . 26
5.4. Sammeln der Muster 27
5.5. Löschen von zu langen unterstützenden Segmenten 28
5.6. Löschen doppelt vorkommender unterstützender Segmente 29

A. Beispiel zur Maximierung der Punkt-Abdeckung

In Abbildung A.1 sind alle gefundenen Muster dargestellt. Es ist möglich, dass unterschiedliche Muster die gleiche Farbe haben.

Bei den Beispielen (Abbildungen A.2 - A.7) ist auf der linken Seite das orts- und richtungsabhängige Muster zu sehen, während auf der rechten Seite alle Segmente, in denen das Muster zu finden ist, gefärbt sind. Hierbei ist es möglich, dass unterschiedliche Segmente die gleiche Farbe haben.

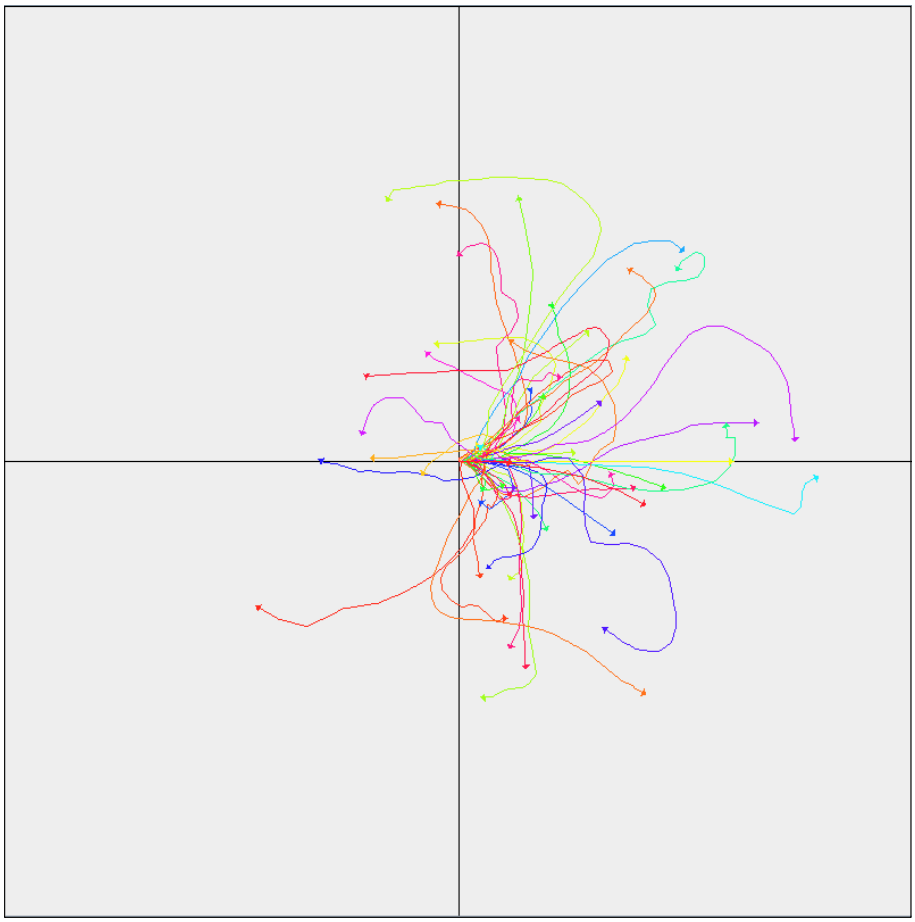

Abbildung A.1.: Alle Muster bei optimalen Parametern für eine maximale Punkt-Abdeckung

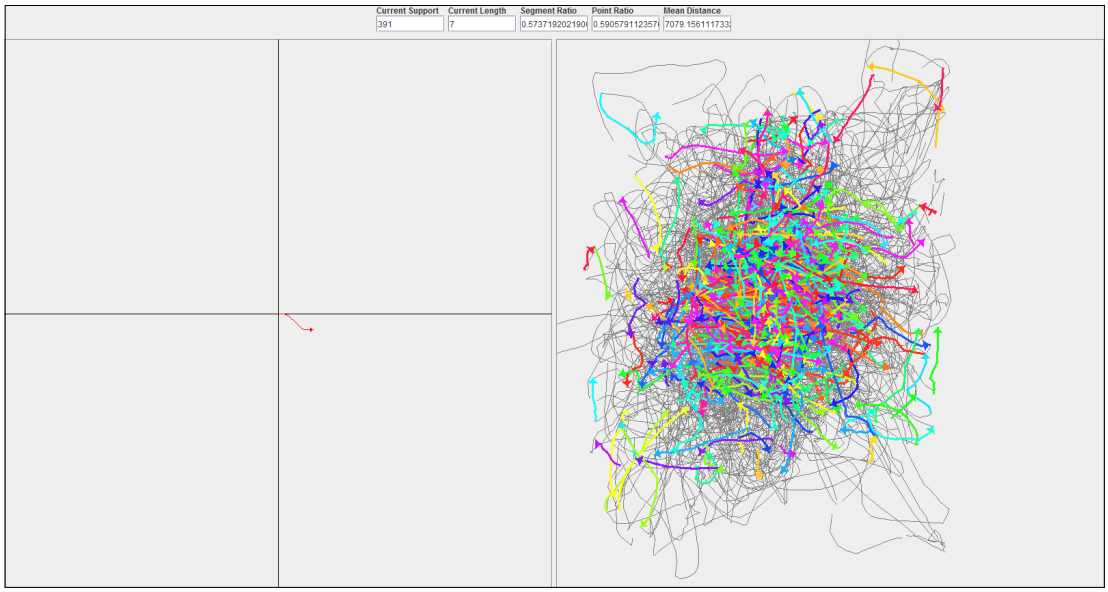

Abbildung A.2.: Beispiel 1 bei optimalen Parametern für eine maximale Punkt-Abdeckung

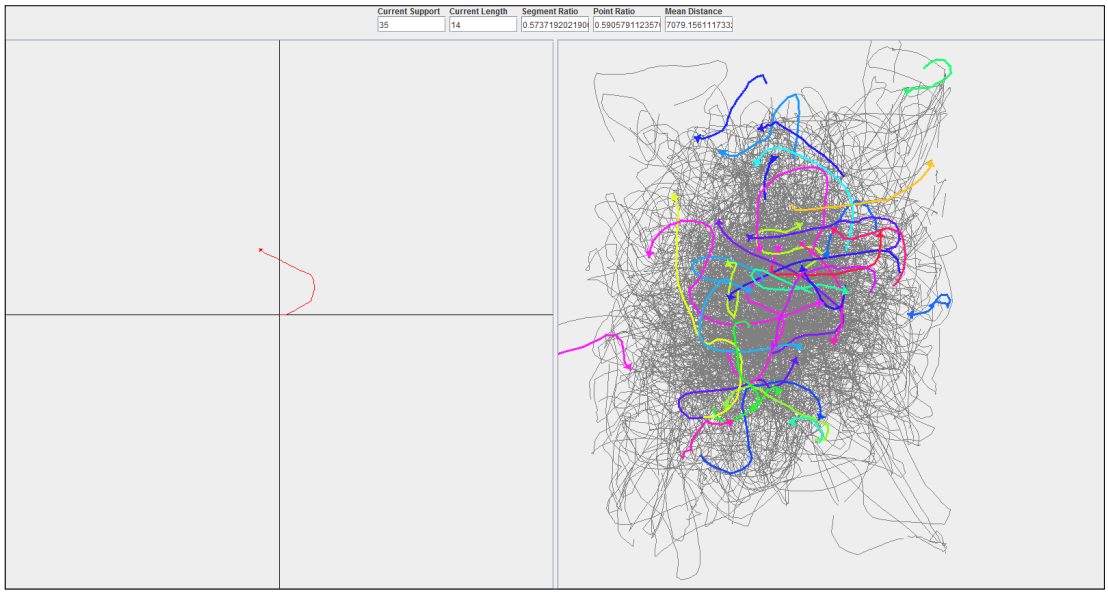

Abbildung A.3.: Beispiel 2 bei optimalen Parametern für eine maximale Punkt-Abdeckung

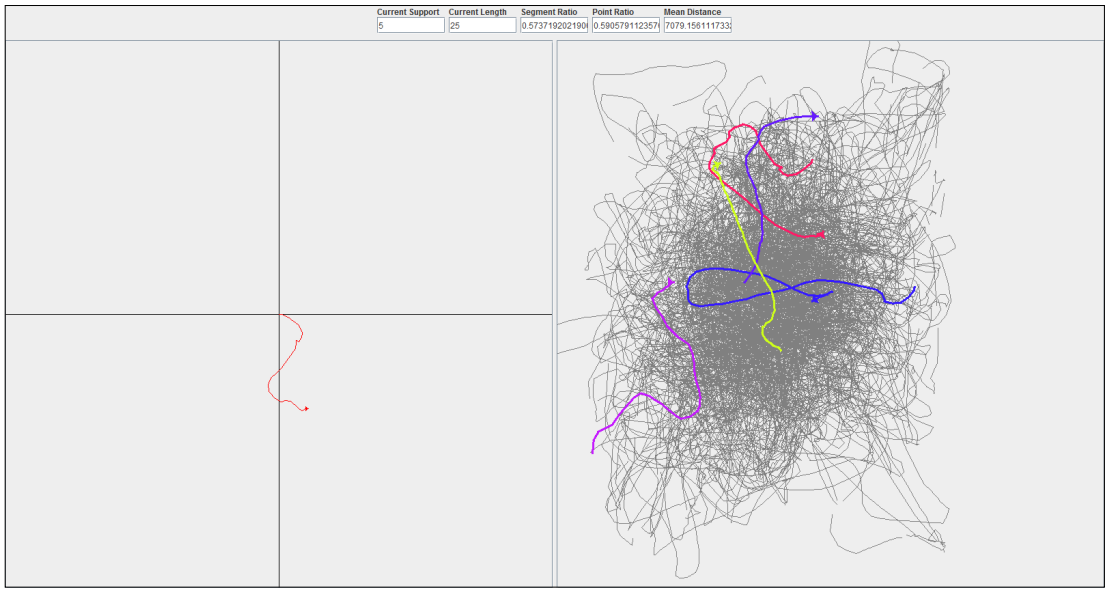

Abbildung A.4.: Beispiel 3 bei optimalen Parametern für eine maximale Punkt-Abdeckung

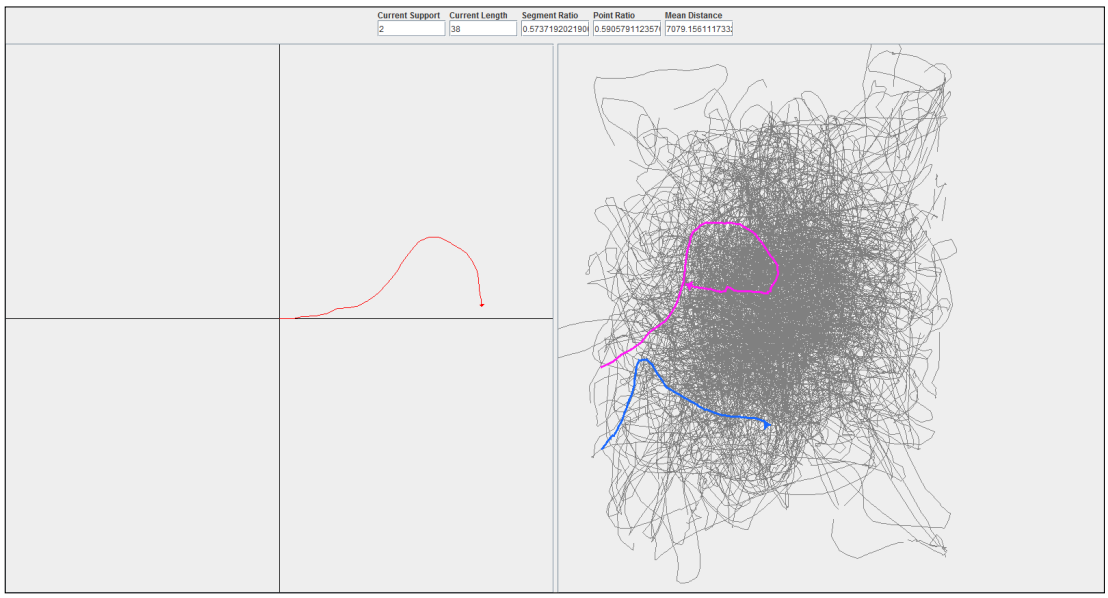

Abbildung A.5.: Beispiel 4 bei optimalen Parametern für eine maximale Punkt-Abdeckung

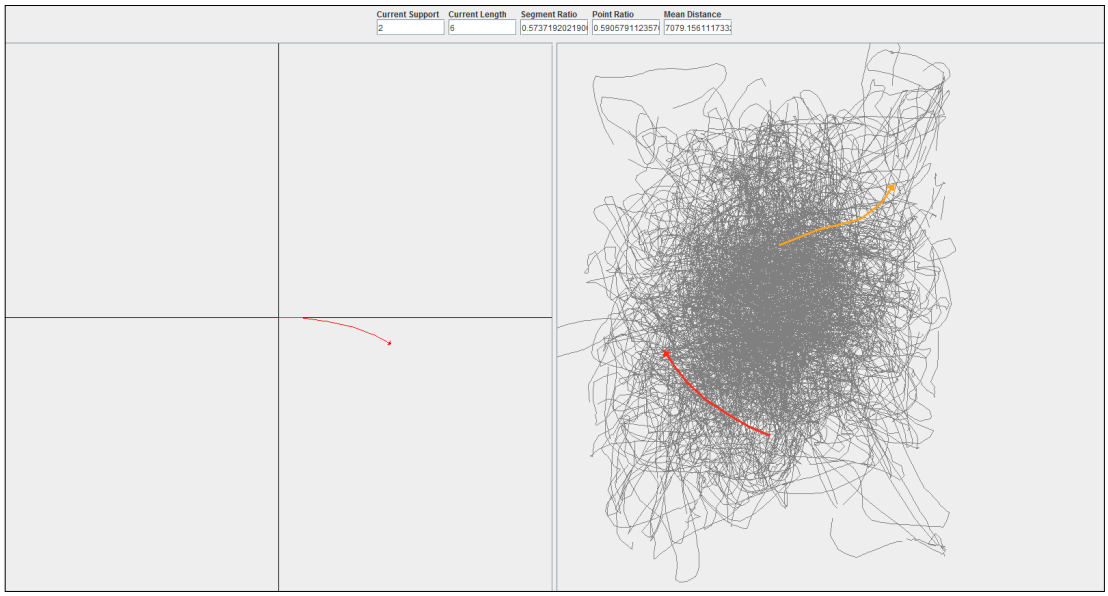

Abbildung A.6.: Beispiel 5 bei optimalen Parametern für eine maximale Punkt-Abdeckung

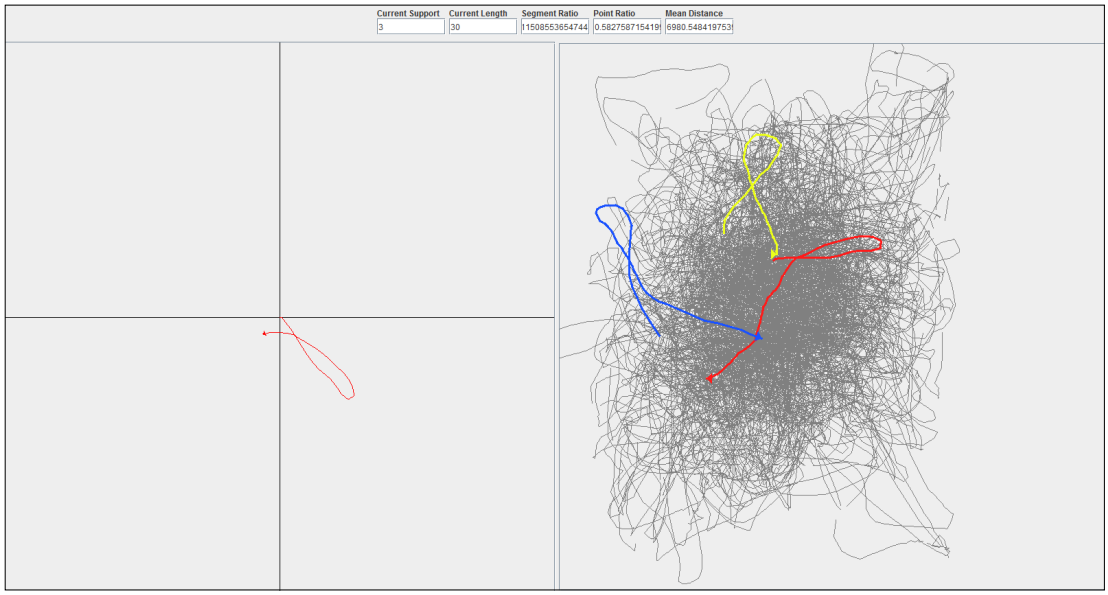

Abbildung A.7.: Beispiel 6 bei optimalen Parametern für eine maximale Punkt-Abdeckung

B. Beispiel zur Maximierung der Segment-Abdeckung

In Abbildung B.1 sind alle gefundenen Muster dargestellt. Es ist möglich, dass unterschiedliche Muster die gleiche Farbe haben.

Bei den Beispielen (Abbildungen B.2 - B.6) ist auf der linken Seite das orts- und richtungsabhängige Muster zu sehen, während auf der rechten Seite alle Segmente, in denen das Muster zu finden ist, gefärbt sind. Hierbei ist es möglich, dass unterschiedliche Segmente die gleiche Farbe haben.

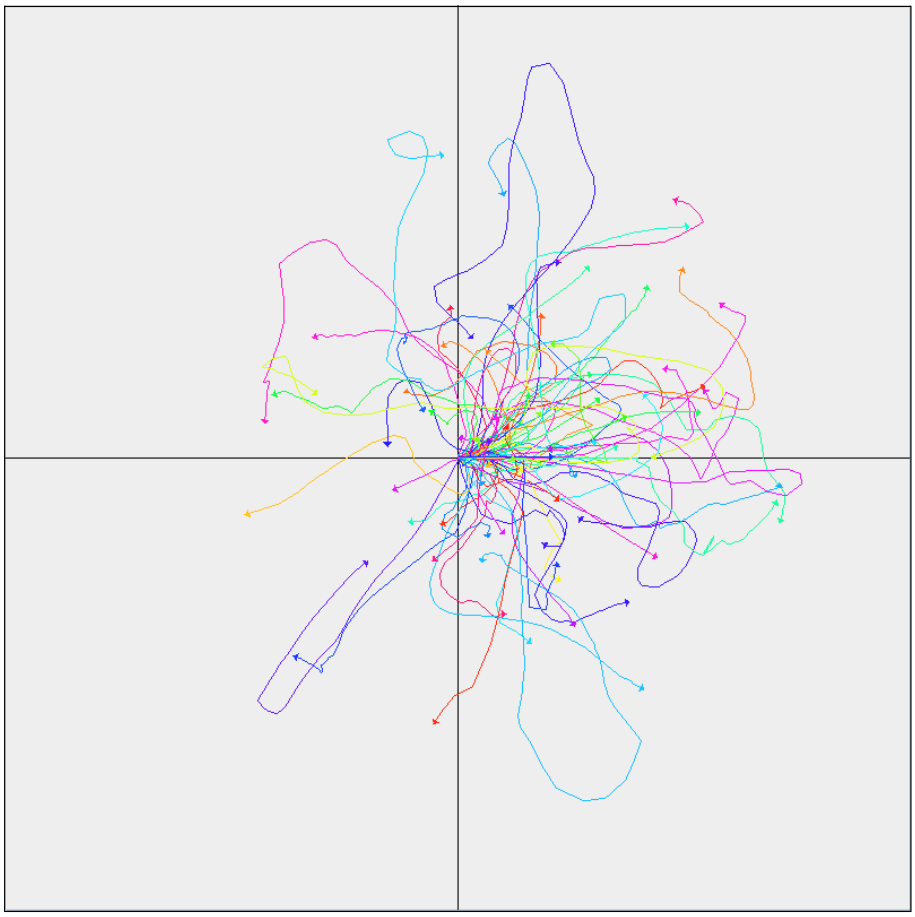

Abbildung B.1.: Alle Muster bei optimalen Parametern für eine maximale Segment-Abdeckung

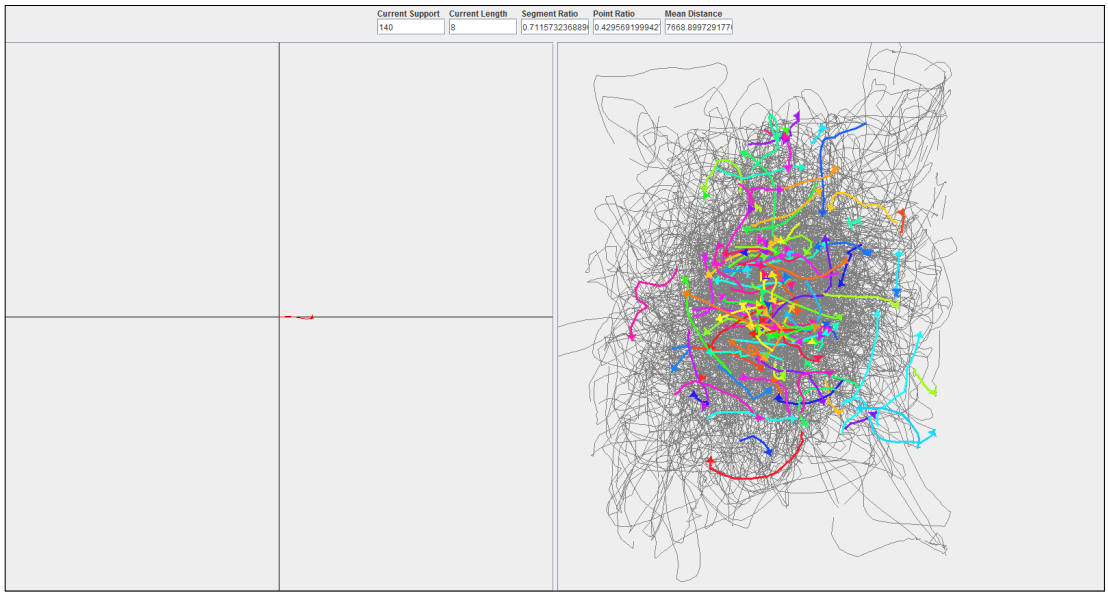

Abbildung B.2.: Beispiel 1 bei optimalen Parametern für eine maximale Segment-Abdeckung

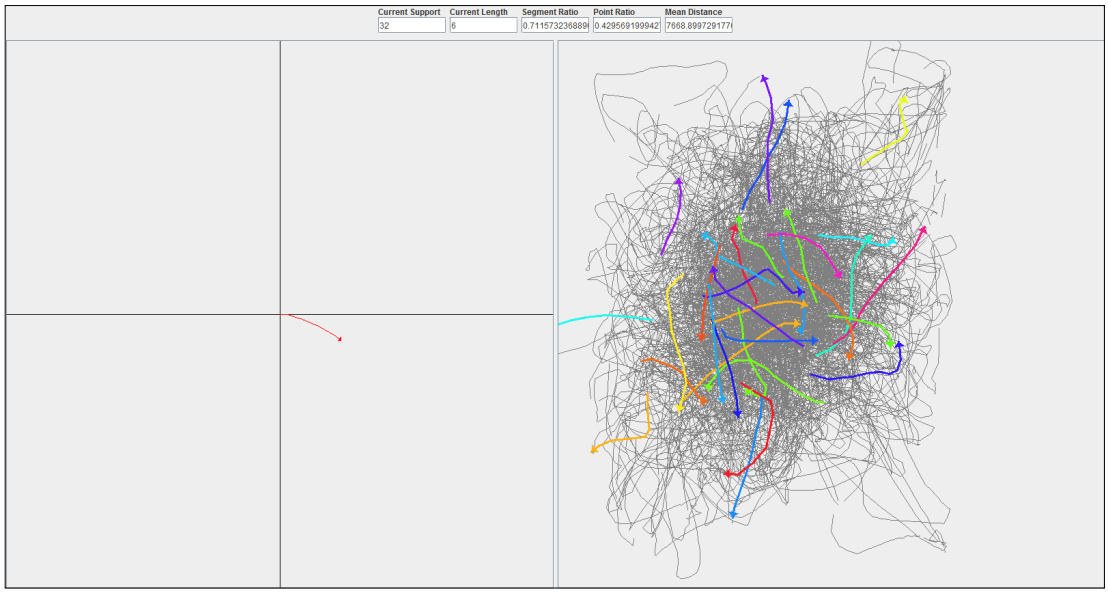

Abbildung B.3.: Beispiel 2 bei optimalen Parametern für eine maximale Segment-Abdeckung

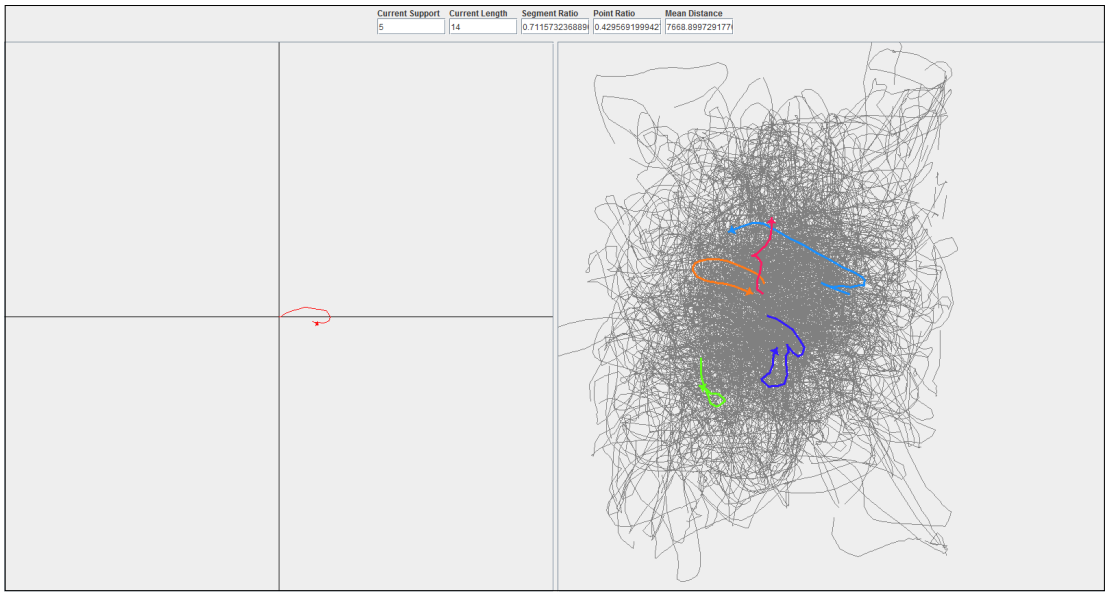

Abbildung B.4.: Beispiel 3 bei optimalen Parametern für eine maximale Segment-Abdeckung

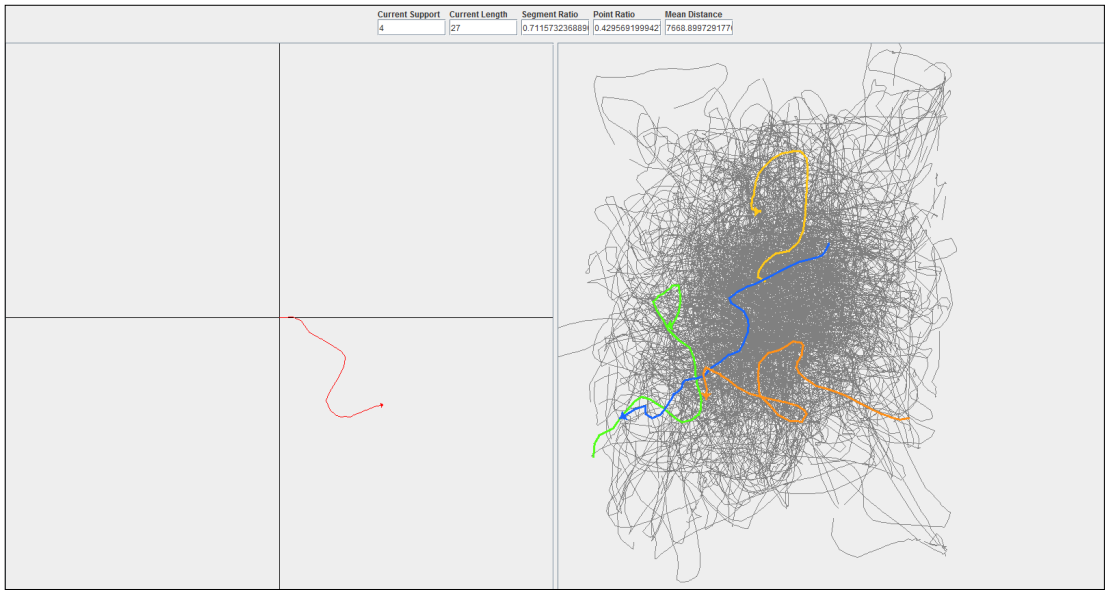

Abbildung B.5.: Beispiel 4 bei optimalen Parametern für eine maximale Segment-Abdeckung

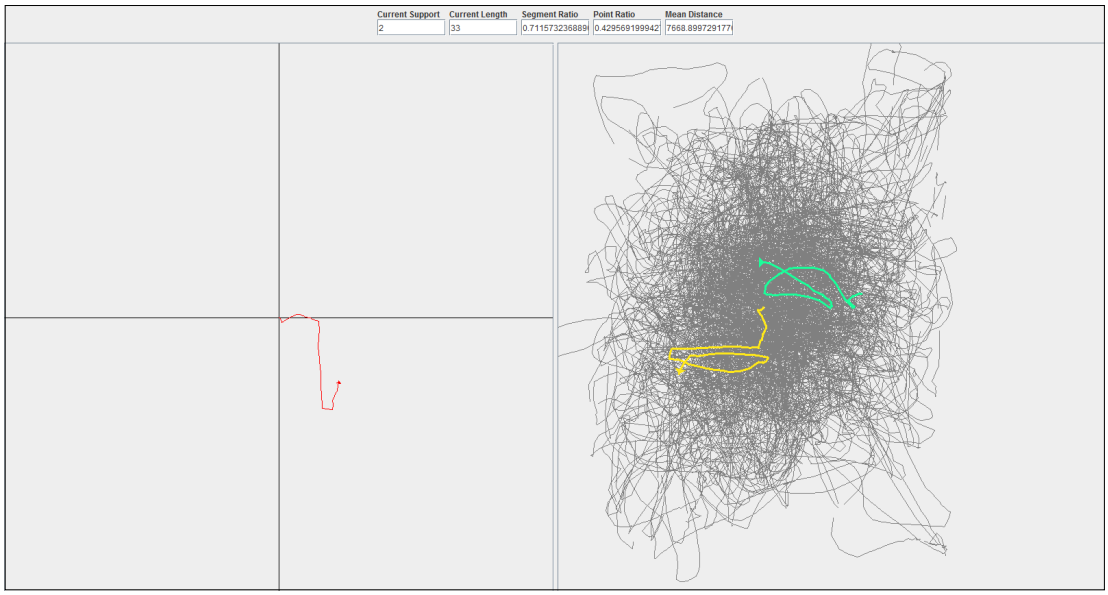

Abbildung B.6.: Beispiel 5 bei optimalen Parametern für eine maximale Segment-Abdeckung

C. Negativbeispiel für falsche Parameterwahl

In Abbildung C.1 sind alle gefundenen Muster des Negativbeispiels dargestellt. Es ist möglich, dass unterschiedliche Muster die gleiche Farbe haben.

Bei den Beispielen (Abbildungen C.2 - C.5) ist auf der linken Seite das orts- und richtungsabhängige Muster zu sehen, während auf der rechten Seite alle Segmente, in denen das Muster zu finden ist, gefärbt sind. Hierbei ist es möglich, dass unterschiedliche Segmente die gleiche Farbe haben.

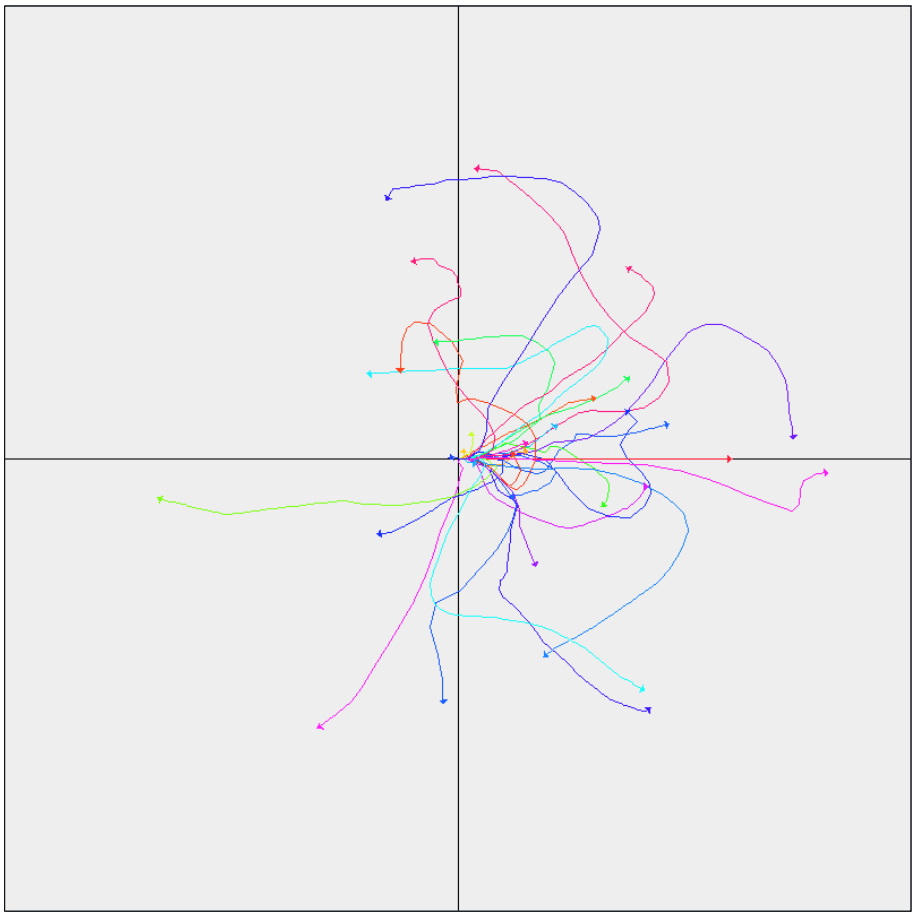

Abbildung C.1.: Negativbeispiel: Alle Muster bei leicht überhöhten Werten bzw. leicht verringerter Mindestlänge

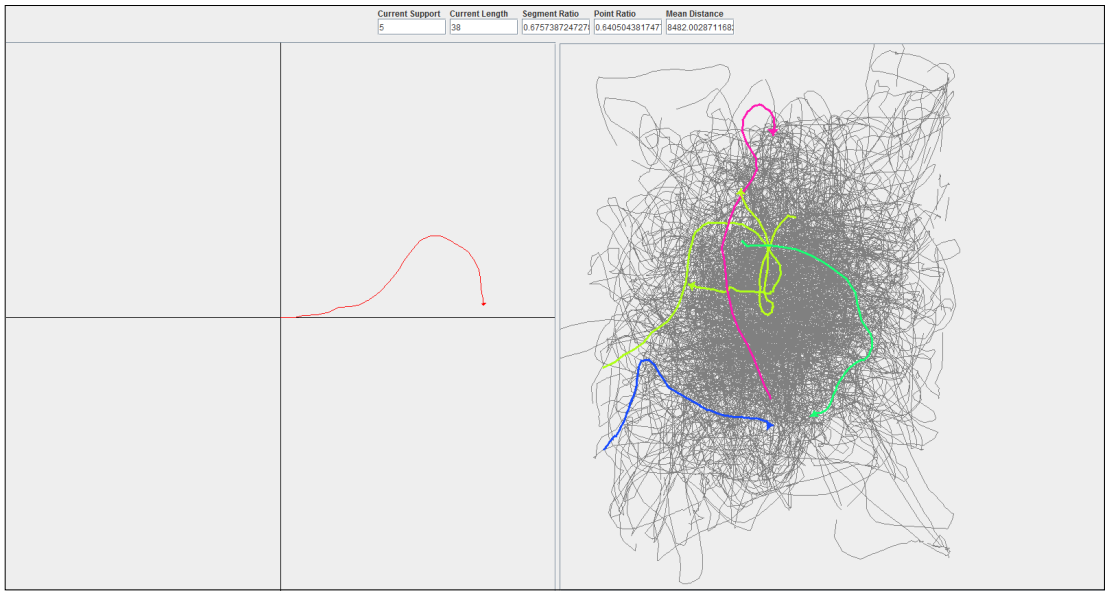

Abbildung C.2.: Negativbeispiel: Beispiel 1 bei leicht überhöhten Werten bzw. leicht verringerter Mindestlänge

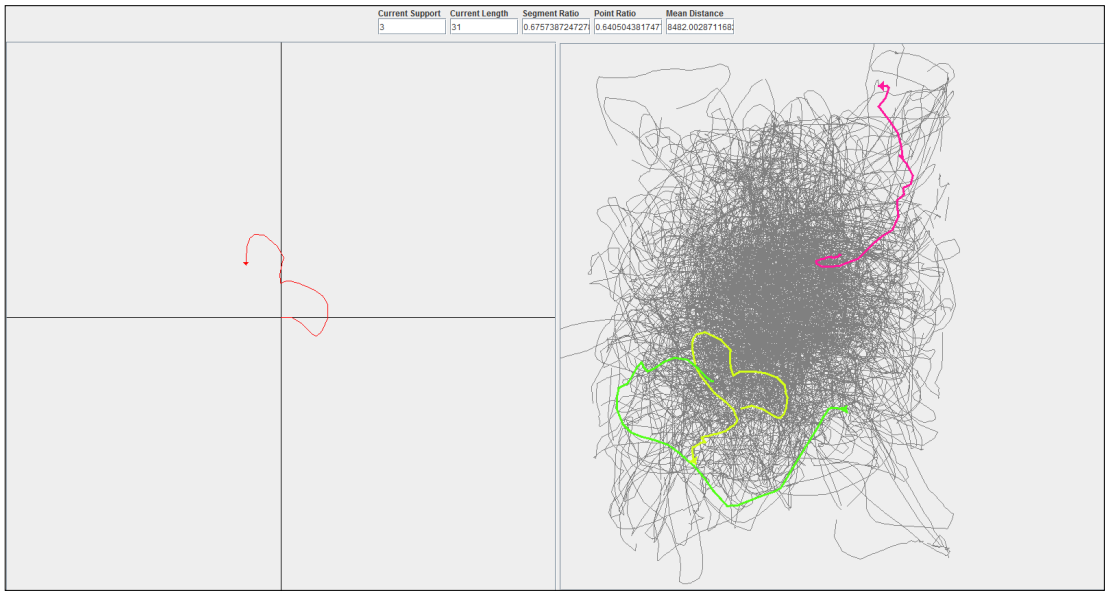

Abbildung C.3.: Negativbeispiel: Beispiel 2 bei leicht überhöhten Werten bzw. leicht verringerter Mindestlänge

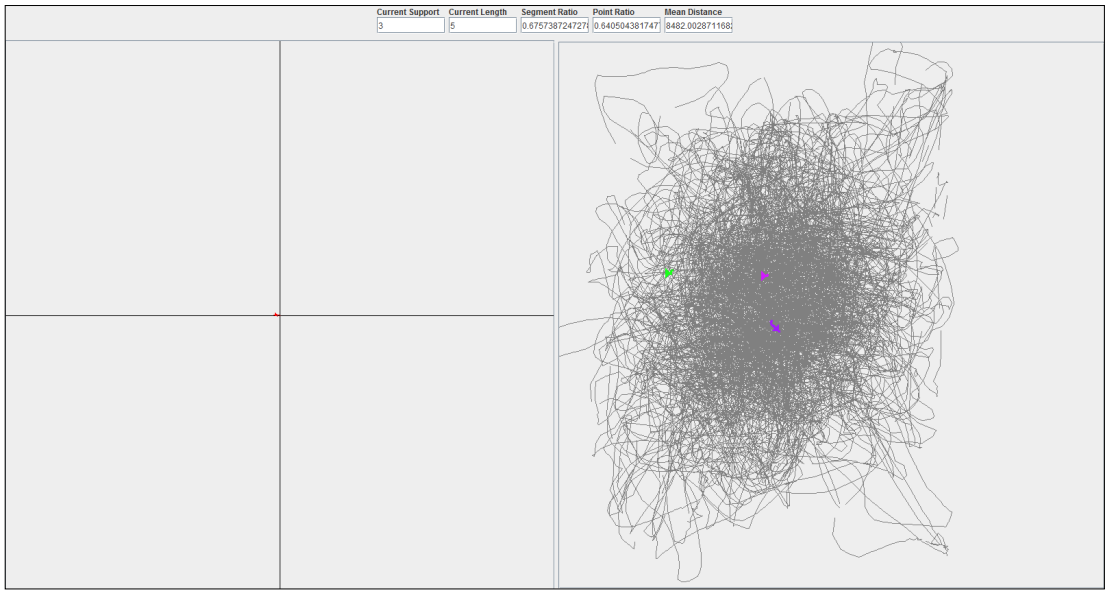

Abbildung C.4.: Negativbeispiel: Beispiel 3 bei leicht überhöhten Werten bzw. leicht verringerter Mindestlänge

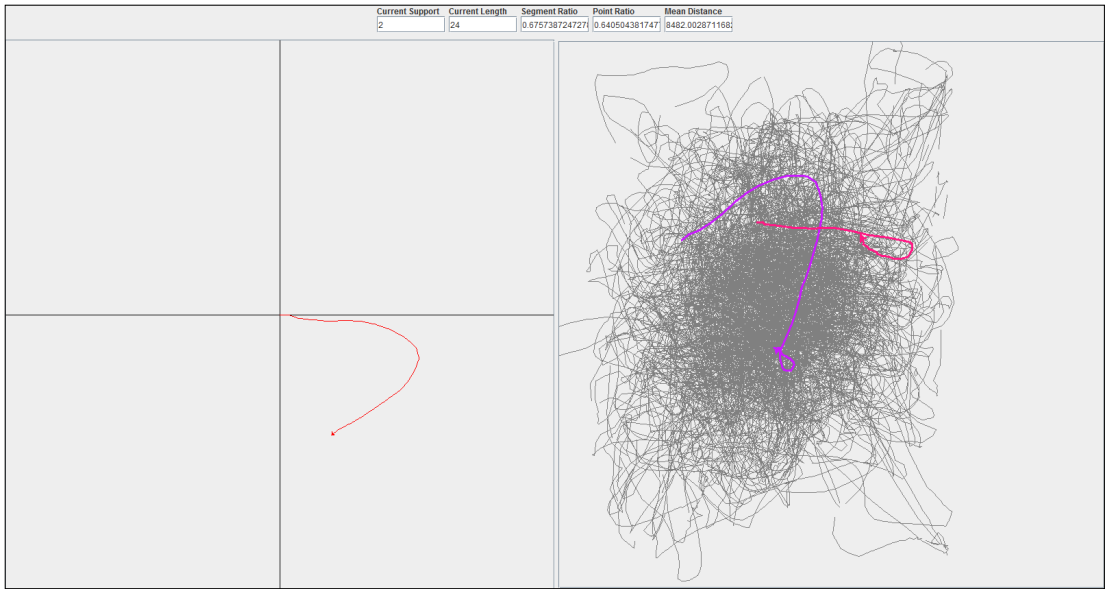

Abbildung C.5.: Negativbeispiel: Beispiel 4 bei leicht überhöhten Werten bzw. leicht verringerter Mindestlänge

D. Qualitative Auswertung der gefundenen Muster

Tabelle D.1.: Zuordnung von Intention zu den Mustern, die mit optimalen Parametern für eine Maximierung der Punkt-Abdeckung gefunden wurden

Muster Nr.	Intention
1	verschieben
2	nicht eindeutig
3	verschieben
4	verschieben
5	Tor schießen oder verhindern
6	nicht eindeutig
7	nicht eindeutig
8	Position halten und warten
9	Ball zurückerobern
10	Ball zurückerobern
11	nicht eindeutig
12	nicht eindeutig
13	Tor schießen oder verhindern
14	verschieben
15	nicht eindeutig
16	nicht eindeutig
17	Ball zurückerobern
18	nicht eindeutig
19	Ball zurückerobern
20	nicht eindeutig
21	nicht eindeutig
22	Ball zurückerobern
23	Position halten und warten
24	Tor schießen oder verhindern
25	nicht eindeutig
26	verschieben
27	Ball zurückerobern
28	nicht eindeutig
29	nicht eindeutig
30	Tor schießen oder verhindern
31	Ball zurückerobern
32	nicht eindeutig
33	nicht eindeutig

34	aufrücken
35	nicht eindeutig
36	Tor schießen oder verhindern
37	nicht eindeutig
38	Ball zurückerobern
39	Position halten und warten
40	Tor schießen oder verhindern
41	Ball zurückerobern
42	nicht eindeutig
43	nicht eindeutig
44	Ball zurückerobern
45	verschieben
46	nicht eindeutig
47	Ball zurückerobern
48	Ball zurückerobern
49	nicht eindeutig
50	nicht eindeutig
51	nicht eindeutig
52	nicht eindeutig
53	Ball zurückerobern
54	verschieben

E. Inhalt der CD

Der gedruckten Version dieser Bachelorarbeit liegt eine CD mit folgendem Inhalt bei:

- Bachelorarbeit im PDF- und LATEX-Format
- Quellcode des entwickelten Prototyps
- Screenshots gefundener Muster
- während der Evaluation erstellte Tabellen

Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig und ohne fremde Hilfe verfasst und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel verwendet habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfungsamt vorgelegen.

Hannover, den 28.08.2014

_____

Raphael Voges