



Universität Regensburg

Nutzerzentrierte Indoor-Positionierung für smartphonegestützte Fußgängernavigation

Inaugural-Dissertation zur Erlangung der Doktorwürde
der Fakultät für Sprach-, Literatur- und Kulturwissenschaften
der Universität Regensburg

vorgelegt von
Robert Jackermeier
aus Rohr i.NB

Regensburg 2022

Eingereicht am: 14.02.2022
Tag der Disputation: 13.01.2023
Erstgutachter (Betreuer): Prof. Dr. Bernd Ludwig
Zweitgutachter: Prof. Dr. Ioannis Giannopoulos
Drittgutachter: Prof. Dr. Udo Kruschwitz

Danksagung

„Not all those who wander are lost.“

J.R.R. Tolkien

Wer umherirrt, muss nicht unbedingt verloren sein. In manchen Fällen mag es schon reichen, die Hilfe eines Navigationssystems in Anspruch zu nehmen (doch dazu später). Als ich nach dem Masterstudium eine Stelle als Doktorand am Lehrstuhl für Informationswissenschaft an der Universität Regensburg antrat, war das Ziel der Reise noch nicht abzusehen. Wie schon zuvor im Studium forschte ich im Bereich der Indoor-Positionierung, doch das konkrete Thema der vorliegenden Arbeit kristallisierte sich erst im Laufe der Zeit heraus. In den letzten Jahren wurde ich von vielen Weggefährten unterstützt, denen ich an dieser Stelle ein herzliches Dankeschön aussprechen möchte.

Zunächst gilt dies meinem Erstgutachter und Betreuer Prof. Dr. Bernd Ludwig, ohne dessen Impulse und Ideen meine Reise nie begonnen hätte. Unterwegs hat er mir die nötigen Freiräume gewährt, an meinen selbst gewählten Schwerpunkten zu forschen, und war dennoch stets zur Stelle, um gemeinsam über Lösungen für alle aufkommenden Fragen zu diskutieren. Dank der Bereitschaft von Prof. Dr. Ioannis Giannopoulos und Prof. Dr. Udo Kruschwitz, als weitere Gutachter zu fungieren, wurde letztendlich auch das Ziel der Reise erreicht.

Allen Kollegen und den studentischen Hilfskräften am Lehrstuhl sowie zahlreichen Studierenden danke ich für die Teilnahme an Navigations- und Positionierungs-Experimenten, bei denen so manche Strecke auf dem Regensburger Campus zurückgelegt wurde.

Besonders hervorheben möchte ich das mittlerweile in alle Himmelsrichtungen verstreute URwalking-Team um Christina, Manuel und Markus, dank deren Bemühungen ein stabiles, benutzbares und wissenschaftlich fundiertes Campus-Navigationssystem verfügbar ist, ohne das meine Arbeit in dieser Form nicht möglich gewesen wäre. Ich hoffe, dass sich unsere Wege irgendwann einmal wieder kreuzen.

Auch abseits der wissenschaftlichen Tätigkeit war meine Zeit am Lehrstuhl sehr lehrreich und hat meinen Horizont immens erweitert. Die angenehme Arbeitsatmosphäre im Team und die anregenden Diskussionen werden die letzten Jahre unvergesslich machen.

Und schließlich gebührt ganz besonderer Dank meiner Familie, die mich durch die Höhen und Tiefen meiner Reise begleitet hat. Ihre Unterstützung gab mir die Kraft, bis zum Schluss weiterzumachen. Insbesondere danke ich meiner Mutter und meiner Schwester für das Korrekturlesen der Dissertation.

An dieser Stelle endet meine Reise – nicht verloren, sondern mit Stolz und Freude über das Erreichte. Möge das Ergebnis dieser Arbeit dazu beitragen, dass auch andere nicht umherirren müssen, sondern zielsicher navigieren können.

Abstract

Reliable positioning is an important prerequisite for navigation systems to provide assistance in the right place and at the right time. In contrast to satellite-based positioning outdoors, no similarly ubiquitous technology is available for indoor environments. In this thesis, an indoor positioning system for smartphone-based pedestrian navigation is developed, focusing in particular on real user behaviour.

Based on the state of the art, a basic positioning system is first developed, which uses a particle filter to determine the user position within a graph-based environment model. The basic functionality as well as several extensions that allow the system to adapt to user characteristics are evaluated in two preliminary studies under controlled conditions.

In parallel, usage data is collected by the campus navigation app *URwalking* to investigate navigation behaviour relevant for the positioning system under realistic conditions. Information about user activities to be expected during navigation can be inferred from the properties of planned routes. A study on a heuristically filtered subset of the dataset ($N = 351$) provides insights into prevailing device placements as well as breaks and interruptions during navigation.

Based on these findings, a dataset containing sensor data for a variety of navigation-related activities and device placements is collected. It in turn serves as a basis for training deep-learning models for human activity recognition. After integrating the activity recognition component into the basic system, a study ($N = 69$) is conducted in order to evaluate the positioning accuracy during a navigation task on a route representative for real usage. Through use of activity information and targeted consideration of human behavior during navigation, position tracking here remains stable over the longer term even without external infrastructure.

Zusammenfassung

Zuverlässige Positionsbestimmung ist eine wichtige Voraussetzung für Navigationssysteme, um am richtigen Ort und zur richtigen Zeit Assistenz leisten zu können. Im Gegensatz zu satellitengestützter Positionierung in Außenbereichen existiert innerhalb von Gebäuden keine ähnlich ubiquitär verfügbare Technologie. Diese Arbeit handelt von der Entwicklung eines Indoor-Positionierungssystems für smartphonebasierte Fußgängernavigation, mit speziellem Fokus auf der Berücksichtigung von realem Nutzerverhalten.

Aufbauend auf dem Stand der Technik wird zunächst ein Basis-Positionierungssystem entwickelt, welches mithilfe eines Partikelfilters die Benutzerposition innerhalb eines graphbasierten Umgebungsmodells bestimmt. In zwei Vorstudien erfolgt anschließend unter kontrollierten Bedingungen die Evaluation der grundlegenden Funktionalität sowie mehrerer Erweiterungen zur Anpassung an Benutzereigenschaften.

Parallel dazu werden mithilfe der Campus-Navigations-App *URwalking* Nutzungsdaten erhoben, um das für die Positionsbestimmung relevante Navigationsverhalten der BenutzerInnen unter realistischen Bedingungen zu untersuchen. Die Merkmale der abgerufenen Routen erlauben Rückschlüsse auf die während der Navigation zu erwartenden Benutzeraktivitäten. Eine Studie an einer heuristisch gefilterten Untermenge des Datensatzes ($N = 351$) gibt unter anderem Aufschluss über vorherrschende Gerätepositionen sowie über Pausen und Unterbrechungen im Navigationsvorgang.

Basierend auf diesen Erkenntnissen wird ein Datensatz erhoben, welcher Sensordaten für eine Vielzahl von navigationsrelevanten Aktivitäten und Gerätepositionen enthält. Dieser wiederum dient als Grundlage für das Training von Deep-Learning-Modellen zur Aktivitätserkennung. Nach Integration der Aktivitätserkennungskomponente in das Basissystem wird die Positionierungsgenauigkeit während eines Navigationstasks auf einer für den realen Betrieb repräsentativen Route in einer abschließenden Studie ($N = 69$) untersucht. Durch geschickte Nutzung der Aktivitätsinformationen und gezielte Berücksichtigung menschlichen Verhaltens während der Navigation bleibt die Positionsverfolgung hier auch ohne externe Infrastruktur längerfristig stabil.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation und Problemstellung	2
1.2	Herangehensweise und Methodik	5
1.3	Begriffsklärungen	5
1.4	Veröffentlichungen	8
1.5	Aufbau der Arbeit	9
2	Vorarbeiten und technische Grundlagen	11
2.1	URwalking: Navigation auf dem Regensburger Campus	11
2.2	Umgebungsmodell: der Navigationsgraph	13
2.2.1	Graphstruktur	13
2.2.2	Landmarkenbasierte Navigation	14
2.2.3	Technische Umsetzung	15
2.3	Plattform und Sensorik	17
2.3.1	Das Android-Betriebssystem	17
2.3.2	Sensoren im Detail	17
2.3.3	Testgeräte	24
2.4	Die URwalking-App für Android	24
2.4.1	Funktionsweise und Funktionsumfang	25
2.4.2	Technische Umsetzung	26
3	Graphbasierte Indoor-Positionierung	29
3.1	Forschungsstand in der Indoor-Positionierung	29
3.1.1	Infrastrukturfreie Positionierung mit PDR	30
3.1.2	Hybride PDR-Systeme	34
3.1.3	Evaluation von Indoor-Positionierungssystemen	39
3.1.4	Indoor-Positionierung: Fazit	41
3.2	Positionstracking im Navigationsgraph	42
3.2.1	Pedestrian Dead Reckoning	42
3.2.2	Stabilisierung durch den Navigationsgraph	43

3.3	Nutzeradaption mithilfe des Partikelfilters	45
3.3.1	Grundlagen	46
3.3.2	Implementierung	46
3.3.3	Anpassungen an fußgängerspezifisches Verhalten	51
3.4	Studien zur Indoor-Positionierung	54
3.4.1	Evaluationsmethodik	55
3.4.1.1	Aufgabenorientierte Evaluation	55
3.4.1.2	Offline-Simulationsstudien	56
3.4.2	Verifizierung der grundlegenden Funktionsweise	57
3.4.2.1	Experimentaufbau und -vorbereitungen	57
3.4.2.2	Datenaufzeichnung	57
3.4.2.3	Analyse und Verifizierung der Sensordaten	59
3.4.2.4	Ergebnisse	59
3.4.2.5	Detailbetrachtungen	61
3.4.2.6	Fazit	61
3.4.3	Studie unter realistischen Bedingungen	62
3.4.3.1	Experimentaufbau	62
3.4.3.2	Datenaufzeichnung	64
3.4.3.3	Ergebnisse	64
3.4.3.4	Fazit	73
4	Positionierungsrelevantes Navigationsverhalten	75
4.1	Leitfragen der naturalistischen Studie	76
4.2	Literaturübersicht	78
4.2.1	Naturalistische Studien und Logdatenanalyse	78
4.2.2	Studien zum menschlichen Navigations- und Bewegungsverhalten	80
4.3	Der Navigations-Log-Datensatz	82
4.3.1	Erhebung der Daten	82
4.3.2	Vorverarbeitung	85
4.4	Verwendung der Navigations-App URwalking	86
4.4.1	Räumliche und zeitliche Muster	86
4.4.2	Charakteristika von Navigationssitzungen	90
4.4.3	Nutzercharakteristik	94
4.4.4	Geräte und Sensoren	96
4.5	Benutzerverhalten während der Navigation	99
4.5.1	Filterheuristik für tatsächliche Navigationssitzungen	99
4.5.2	Gehverhalten	101
4.5.3	Unterbrechungen in der App-Nutzung	105

4.5.4	Geräteposition	106
4.6	Fazit der Logdatenauswertung	108
5	Aktivitätserkennung in der Fußgängernavigation	109
5.1	Das Forschungsfeld Human Activity Recognition	111
5.1.1	Methodik	111
5.1.2	Smartphonebasierte Aktivitätserkennung	114
5.1.3	HAR in der Fußgängernavigation	115
5.1.4	Existierende HAR-Datensätze	118
5.2	Erkennung von navigationsspezifischen Aktivitäten	119
5.2.1	Aktivitäten und Gerätepositionen	120
5.2.2	Erhebung des HAR-Datensatzes	122
5.2.2.1	Die Datenerhebungs-App	122
5.2.2.2	Verwendete Sensoren	123
5.2.2.3	Vorverarbeitung des Datensatzes	124
5.2.3	Machine-Learning-Methodik	126
5.2.3.1	Modelle	126
5.2.3.2	Training und Evaluation	128
5.2.3.3	Die Klassifikator-Hierarchie	129
5.2.4	Evaluation der HAR-Modelle	129
5.2.4.1	Bestimmung der Geräteposition	131
5.2.4.2	Flache Klassifikation	132
5.2.4.3	Zweistufige Klassifikation	134
5.2.4.4	Fazit	139
5.3	Anwendung von HAR in naturalistischen Studien	140
5.3.1	Geräteposition während der Navigation	141
5.3.2	Aktivitäten während der Navigation	143
5.4	HAR zur Verbesserung der Indoor-Positionierung	144
5.4.1	Integration von HAR in das Positionierungssystem	145
5.4.2	Positionierungsexperiment mit HAR	146
5.4.2.1	Experimentaufbau und Datenerhebung	147
5.4.2.2	Datenaufbereitung	150
5.4.2.3	Hypothesen	151
5.4.2.4	Ergebnisse	151
5.4.2.5	Detailstudie	156
5.5	Fazit zur Aktivitätserkennung	157

6	Diskussion: Nutzerzentrierte Indoorpositionierungssysteme	159
6.1	Grenzen von reinen PDR-Systemen	159
6.1.1	Evaluationsmethodik	160
6.1.2	Modellierung der Umgebung	161
6.2	Navigationsverhalten und Nutzung der URwalking-App	162
6.2.1	Navigations Sitzungen allgemein	163
6.2.2	Pausen und Unterbrechungen	163
6.2.3	Geräteposition und Trageart	164
6.3	Aktivitätserkennung mithilfe von Gerätesensoren	165
6.3.1	Klassifikation der Geräteposition	165
6.3.2	Klassifikation der Aktivitäten	165
6.4	Anwendung von HAR in der Indoor-Positionierung	167
6.4.1	Trefferquote allgemein	168
6.4.2	Falschpositive und falschnegative Klassifikation	169
6.4.3	Einbindung in das Umgebungsmodell	169
7	Fazit und Ausblick	171
7.1	Zusammenfassung	171
7.2	Wissenschaftlicher Beitrag	172
7.3	Limitierungen	174
7.4	Ausblick	176
	Abbildungsverzeichnis	179
	Tabellenverzeichnis	183
	Literaturverzeichnis	185

Kapitel 1

Einleitung

In industrialisierten Gesellschaften verbringen Menschen einen Großteil ihrer Zeit innerhalb von Gebäuden – Klepeis et al. (2001, S. 15) nennen für die USA einen Wert von 86,9 %. Neben den eigenen vier Wänden teilt sich diese Zeit u. a. auf Bürogebäude und Fabriken, öffentliche Gebäude und Schulen sowie Geschäfte und Einkaufszentren auf. Sich innerhalb von Gebäuden zurechtzufinden und zu vormals unbekanntem Zielen zu navigieren, ist daher Bestandteil des Alltags vieler Menschen.

Für die Navigation im Außenbereich, insbesondere in Fahrzeugen, sind satellitengestützte Navigationssysteme (*Global Navigation Satellite Systems*, GNSS) etabliert, zunehmend auch in Form von Smartphone-Anwendungen (Brown & Laurier, 2012, S. 1621). Laut einer Erhebung aus dem Jahr 2018 verwenden 77 % der Smartphone-BenutzerInnen regelmäßig Navigations-Apps, wobei die Assistenz während des Führens von Fahrzeugen den größten Anteil ausmacht (Panko, 2018).

Smartphonebasierte Fußgängernavigation innerhalb von Gebäuden ist weit weniger verbreitet. Eine Ursache hierfür ist die Nicht-Verfügbarkeit von GNSS im Innenbereich, weswegen beispielsweise Huang, Gartner, Krisp, Raubal und Van de Weghe (2018, S. 74f.) *ubiquitous positioning*, also ein immer und überall verfügbares Positionierungssystem, als eine der Schlüsselherausforderungen im Forschungsbereich *Location Based Services* (LBS) nennen.

Neben der Indoornavigation besteht in vielen weiteren Anwendungsfeldern Bedarf nach Indoor-Positionierungsverfahren, beispielsweise bei Rettungsdiensten und Feuerwehren (Ingram, Harmer & Quinlan, 2004; Rantakokko et al., 2011), in der Industrie und Robotik, etwa für die Automation von Fabrikhallen (Cotera, Velazquez, Cruz, Medina & Bandala, 2016; W. Guan et al., 2020; Yazici, Yayan & Yücel, 2011), oder für gezieltes standortabhängiges Marketing in Einkaufszentren (van de Sanden, Willems & Brengman, 2019). Dass Indoor-Positionierung ein umkämpftes Forschungsfeld ist, lässt sich auch an den Investitionen und Bemühungen

aus der Industrie ablesen. Neben vielen Start-ups – z. B. MazeMap¹, Navigine² und Navisens³ – sind längst auch die großen Technologiekonzerne in diesem Bereich engagiert: Bereits 2013 übernahm Apple das auf WLAN-Positionierung spezialisierte Start-up WifiSLAM. Im Jahr darauf stellte Google die Tango-Plattform vor, die mithilfe von Bewegungssensoren und Tiefenkameras die Bestimmung von Geräteausrichtung und -position im Raum ermöglichte. Als eines der Anwendungsszenarien wurde Indoor-Positionierung demonstriert (Google Developers, 2014). Ebenfalls im Jahr 2014 begann Microsoft mit der Veranstaltung der Microsoft Indoor Localization Competition, einem bis 2018 jährlich stattfindenden Wettbewerb, dessen Ziel u. a. die einheitliche Evaluation verschiedener Positionierungssysteme war (Lymberopoulos et al., 2015). An den parallel ausgetragenen Wettbewerben im Rahmen der IPIN-Konferenz (*Indoor Positioning and Indoor Navigation*) nahm neben universitären Einrichtungen und anderen Forschungsinstituten beispielsweise auch ein Team von Intel teil (Potorti et al., 2020).

1.1 Motivation und Problemstellung

Für die smartphonebasierte Positionsbestimmung im Innenbereich existiert eine Vielzahl von Lösungen, basierend auf unterschiedlichen technischen Ansätzen, darunter Funktechnologien wie WLAN oder Bluetooth, Inertialsensorik und Kameras (Davidson & Piche, 2016; Gu et al., 2019). Unter kontrollierten Bedingungen wird von den besten Systemen eine Positionierungsgenauigkeit in einer Größenordnung von ca. 2–5 m erreicht (Potorti et al., 2021, S. 28).

Keine dieser technischen Verfahren kann jedoch eine ähnliche Verbreitung wie die satellitengestützten Lösungen im Außenbereich vorweisen. Eine Ursache dafür ist die deutlich aufwändigere Skalierung der Positionierungslösungen auf andere Gebäude, insbesondere beim Einsatz von infrastrukturabhängigen Verfahren wie der WLAN-Ortung (Gu et al., 2019, S. 0:10). Ein weiterer Grund sind die variablen Anforderungen hinsichtlich der Genauigkeit und Zuverlässigkeit von Positionierungssystemen, auf die beispielsweise Huang et al. (2018, S. 75f.) hinweisen. Je nach Anwendungsszenario sind unterschiedliche Aspekte wie etwa die horizontale und vertikale Positionierungsgenauigkeit oder die Frequenz und Latenz der Positionsschätzung von Bedeutung, was die Entwicklung einer *One-Size-Fits-All*-Lösung erschwert.

Um den jeweiligen Anforderungen gerecht zu werden, müssen Indoor-Positio-

¹<https://www.mazemap.com>, abgerufen am 10.01.22

²<https://navigine.com>, abgerufen am 10.01.22

³<https://navisens.com>, abgerufen am 10.01.22

nierungsverfahren demnach auf einen spezifischen Anwendungszweck hin optimiert werden. Dass dabei das Verhalten von realen BenutzerInnen einer Navigationsanwendung selten im Vordergrund steht, beklagen Fang, Li und Shaw mit den Worten »*What about people in pedestrian navigation?*« und plädieren dafür, Fußgängernavigationssysteme nicht prozessorientiert, sondern um Nutzerbedürfnisse herum zu gestalten (Fang et al., 2015, S. 135). Im dort entwickelten Modell wird das Zusammenspiel von Fußgänger, Navigationssystem und Umgebung thematisiert, wobei die zuverlässige Positionsbestimmung eine zentrale Rolle einnimmt. Die Forderung nach Berücksichtigung der Nutzerinteressen greift zudem Elemente des *User Centered Design* (Garrett, 2011; Norman, 1986) und des Manifests für Agile Softwareentwicklung (Beck et al., 2001) auf.

An der Universität Regensburg ist das Thema Indoornavigation in Form der vom Lehrstuhl für Informationswissenschaft betriebenen Campusnavigation *URwalking* vertreten. Eine Besonderheit des Systems ist die Wegführung anhand von natürlichsprachlichen Navigationsanweisungen, die sich auf markante Objekte in der Umgebung – üblicherweise als *Landmarken* bezeichnet (Golledge, 1991; Lynch, 1960) – beziehen. Eine weitere hervorzuhebende Eigenschaft ist die Modellierung der Umgebung in einer Graphstruktur, in der neben den genannten Landmarken die möglichen Zielorte sowie die begehbaren Wege auf dem Campus modelliert sind (Ullmann, 2020, S. 12–57).

Ziel dieser Arbeit ist es, ein Indoor-Positionierungssystem zu entwickeln und zu evaluieren, das auf diese Art der smartphonegestützten Fußgängernavigation zugeschnitten ist. Die Eigenheiten des menschlichen Navigationsverhaltens werden dabei als Chancen betrachtet, mithilfe derer die Positionsverfolgung zuverlässiger gestaltet werden kann. Im Rahmen der Arbeit werden daher ebenfalls die für ein Indoor-Positionierungssystem relevanten menschlichen Verhaltensweisen und Aktivitäten während der Fußgängernavigation untersucht. Dies wird dadurch ermöglicht, dass es es sich – im Gegensatz zu den meisten Forschungsprototypen aus der Literatur – bei *URwalking* um ein im Realbetrieb verwendbares System mit tatsächlicher Nutzerbasis handelt. Im Laufe der Arbeit werden deswegen auch immer die für den praktischen Einsatz der entwickelten Lösung relevanten Aspekte berücksichtigt.

Angesichts der Ressourcen, die weltweit für die Forschung zur Indoor-Positionierung aufgewendet werden, kann es freilich nicht der Anspruch dieser Arbeit sein, das Problem ein für alle Mal zu lösen. Stattdessen liegt der Fokus auf spezifischen Möglichkeiten, mithilfe der Berücksichtigung von menschlichem Navigationsverhalten eine für den vorliegenden Anwendungsfall adäquate Positionsbestimmung zu erreichen. Die Randbedingungen dieser Arbeit, die durch die lokalen Gegebenheiten und vorherige Projekte vorgegeben sind, werden nachfolgend

in Abschnitt 1.3 sowie in Kapitel 2 näher beschrieben.

Eine der Grundannahmen dieser Arbeit ist, dass für den Zweck der Fußgänger-navigation nicht zwangsweise eine Positionierungsgenauigkeit im Meter-Bereich vonnöten ist. Vielmehr wird es als ausreichend betrachtet, das Betreten und Verlassen von Routenabschnitten zu identifizieren, um dafür korrekte Navigationsanweisungen zu generieren und somit Navigationsassistenz zu leisten. Je nach Art der Unterteilung der Route können Abschnittswechsel auf unterschiedliche Weise bestimmt werden. Werden Routenabschnitte durch Abbiegungen, Türen, Treppen, o. Ä. begrenzt, so kann dies mittels Aktivitätserkennung (*Human Activity Recognition*, HAR) identifiziert werden (siehe Kapitel 5). Ist dies nicht der Fall, so wird die Position anhand von Schritterkennung und sog. *Pedestrian Dead Reckoning* (PDR) verfolgt (siehe Kapitel 3).

Nach diesem einleitenden Überblick lassen sich die Forschungsfragen dieser Arbeit formulieren, mit denen sich die folgenden Kapitel auseinandersetzen werden:

- Inwieweit kann bereits ein grundlegendes graphbasiertes Indoor-Positionierungssystem die Position von FußgängerInnen im Innenbereich verfolgen? Was sind die Grenzen eines rein PDR-basierten Systems?
- Welche Verhaltensweisen legen BenutzerInnen eines smartphonebasierten Indoor-Navigationssystems an den Tag und wie wirken sich diese auf ein Indoor-Positionierungssystem aus?
- Wie können Nutzeraktivitäten während der Navigation zuverlässig mithilfe von Gerätesensoren erkannt werden?
- Wie lässt sich die Genauigkeit eines Indoor-Positionierungssystems durch gezielte Berücksichtigung von ortsbezogenen Aktivitäten verbessern?

Abschließend soll an an dieser Stelle kurz der Bezug zum Fach Informationswissenschaft hergestellt werden: Üblicherweise ist die Entwicklung von Positionierungssystemen im universitären Kontext bei den Ingenieurwissenschaften, bei der Geoinformatik o. Ä. angesiedelt. In der vorliegenden Arbeit liegt der Fokus jedoch nicht darauf, die technischen Möglichkeiten auszureizen; vielmehr steht der Benutzer des Systems im Mittelpunkt. Dies entspricht dem Selbstverständnis der Informationswissenschaft als angewandte Informatik mit Fokus auf der Interaktion zwischen Mensch und Computer. Wie Kattenbeck, Müller, Ohm und Ludwig (2015) darlegen, ist insbesondere das Thema Informationsverhalten in der Forschung zu Fußgänger-navigation in verschiedenen Facetten gegenwärtig. Auch die

in der Arbeit eingesetzten, im folgenden Abschnitt näher beschriebenen Methoden decken ein breites Spektrum der informationswissenschaftlichen Methodik ab.

1.2 Herangehensweise und Methodik

Um die gestellten Ziele zu erreichen, stützt sich die Arbeit methodisch auf mehrere Standbeine.

Das entwickelte Positionierungssystem basiert auf einem stochastischen Filterverfahren (Partikelfilter), welches als Kernstück des Verfahrens die Verarbeitung unsicherer Information (in Form von Smartphone-Sensordaten) ermöglicht und diese in Kombination mit einem Modell der Umgebung zu einer Positionsschätzung verarbeitet. Die Evaluation des Systems findet mithilfe von Simulationsstudien statt, für welche zuvor in kontrollierten Navigationsexperimenten die nötigen Sensordaten erhoben werden, wobei Wert auf möglichst hohe Realitätsnähe gelegt wird.

Um Einblicke in das Verhalten realer NutzerInnen von Fußgängernavigationssystemen zu erlangen, werden naturalistische Log-Analysen durchgeführt. Diese bieten gegenüber Labor- oder In-Situ-Studien den großen Vorteil, dass das System in gewohnter Weise ohne eine künstliche Experimentsituation benutzt werden kann, und ermöglichen es, Daten von einer Vielzahl von Personen zu erheben (Dumais, Jeffries, Russell, Tang & Teevan, 2014). Zur Auswertung der Logdaten kommen Verfahren der deskriptiven und der Inferenzstatistik zum Einsatz.

Die anschließenden Studien zur Aktivitätserkennung während der Navigation werden mithilfe von Verfahren des maschinellen Lernens (*Machine Learning*, ML), insbesondere des *Deep Learning*, durchgeführt, welche den aktuellen Stand der Technik darstellen.

Gegen Ende der Arbeit ergänzen sich die verwendeten Methoden gegenseitig: Zum einen werden mithilfe von trainierten ML-Modellen die Analysen der naturalistischen Daten verfeinert, wodurch eine genauere Aufschlüsselung der dort beobachteten Phänomene möglich wird. Zum anderen finden die ML-Modelle Einzug in das Positionierungssystem und extrahieren dort mehr Informationen aus den Sensordaten, um die Positionsbestimmung zu verbessern.

1.3 Begriffsklärungen

Ausgehend vom Titel der Arbeit – *Nutzerzentrierte Indoor-Positionierung für smartphonegestützte Fußgängernavigation* – werden in diesem Abschnitt die Eckpunkte

der Arbeit abgesteckt sowie einige Begrifflichkeiten definiert und präzisiert.

Nutzerzentriert

Der Aspekt der Nutzerzentrierung kommt im Verlauf dieser Arbeit in unterschiedlicher Weise zum Tragen.

Wie eingangs bereits erwähnt, liegt der Fokus nicht darauf, durch technische Mittel höchstmögliche Positionierungsgenauigkeit zu erreichen. Arbeiten, die dieses Ziel verfolgen, sind zumeist nicht ohne erheblichen Infrastruktur-Aufwand skalier- oder einsetzbar. Stattdessen soll durch die Beschränkung auf handelsübliche Smartphones und den Verzicht auf zusätzliche Infrastruktur eine für die breite Masse nutzbare Lösung entstehen. Anstelle technischer Ortungsverfahren stehen vielmehr die während der Navigation von den BenutzerInnen an den Tag gelegten Verhaltensweisen im Vordergrund, mithilfe derer der Fortschritt entlang einer geplanten Route verfolgt werden kann.

Ein weiterer Aspekt der Nutzerzentrierung kommt bei der Evaluation des Systems zu tragen: Anstatt den Positionsfehler durch metrische Distanzen auszudrücken, welche im Innenbereich ohne zusätzliche Informationen über die Gebäudearchitektur nicht immer sinnvoll interpretierbar sind, kommt eine angepasste Evaluationsmetrik zur Anwendung. Diese ist konzeptuell auf die landmarkenbasierte Navigation abgestimmt und ermöglicht eine bessere Einschätzung, inwiefern der Benutzer den Navigationstask erfüllen kann.

Nutzerzentrierung meint jedoch im Rahmen dieser Arbeit nicht, dass auf Sensorik verzichtet wird und die Lokalisierung rein durch Interaktion zwischen Nutzer und Navigationsanwendung stattfindet, wie dies beispielsweise Winter et al. (2019, S. 61:15f.) beschreiben. Ebenso ist kollaborative Positionierung, in der mehrere NutzerInnen durch das Teilen von Informationen voneinander lernen und somit ihre Position im Raum und ihr Verhältnis zueinander besser einschätzen können (vgl. Winter et al., 2019, S. 61:14f.), nicht Bestandteil dieser Arbeit.

Indoor-Positionierung

Die im Rahmen dieser Arbeit entwickelte Indoor-Positionierungslösung zielt auf Gebäude oder Gebäudekomplexe ab, in denen aufgrund von Ausmaßen und Komplexität der Aufwand für die Einrichtung und den Betrieb eines Indoor-Navigationssystems angemessen ist. In diese Kategorie fallen beispielsweise Bürogebäude, Messehallen, Einkaufszentren, Flughäfen und Bahnhöfe und nicht zuletzt auch öffentliche Einrichtungen wie Universitäten. Davon abzugrenzen ist die Positionsbestimmung innerhalb kleinerer Gebäude oder Wohnungen, in denen keine Navigation in diesem Sinne stattfinden kann. Die Forschungsbemühungen in die-

sem Bereich konzentrieren sich vor allem auf den Kontext von *Ambient Assisted Living* (siehe z. B. Álvarez García, Barsocchi, Chessa & Salvi, 2013).

Weiterhin wird vorausgesetzt, dass die Navigation in einer zuvor bekannten, abgeschlossenen Umgebung stattfindet, für welche ein Umgebungsmodell existiert, in dem begehbare Wege und weitere relevante Merkmale – darunter beispielsweise Eingangs- und Durchgangstüren, Treppen und Aufzüge sowie ggf. Landmarken – verzeichnet sind. SLAM-Verfahren (*Simultaneous Localization and Mapping*), d. h. das Erkunden von unbekanntem Umgebungen mit gleichzeitiger Positionierung (siehe z. B. Abdelnasser et al., 2016; Faragher & Harle, 2013) sind hingegen nicht Teil dieser Arbeit.

Die benötigte Positionierungsgenauigkeit orientiert sich an den Anforderungen, die der geplante Anwendungszweck mit sich bringt. Im vorliegenden Fall sollte das System zu jedem Zeitpunkt so genaue Informationen über die Position des Nutzers haben, dass zuverlässig Navigationsanweisungen gegeben werden können. Die exakte Position innerhalb eines kleinen Raumes oder gar Sub-Meter-Genauigkeit wie in Robotik-Anwendungen ist daher nicht notwendig. In der von Richter, Winter, Richter und Stirling (2013, S. 94) vorgeschlagenen Granularitäts-Skala für Standortbeschreibungen entspricht dies dem *Room-Level*, von Winter et al. (2019, S. 61:11) näher definiert als eine Größenordnung von 10 m.

Smartphonegestützt

Um eine breite Verfügbarkeit einer Indoor-Positionierungslösung zu gewährleisten, bietet sich das Smartphone an, welches im vergangenen Jahrzehnt enorme Verbreitung erreicht hat. Nach einer aktuellen Erhebung benutzen 86 % der Deutschen über 14 Jahre ein Smartphone; in den Altersgruppen unter 50 Jahren liegt die Verbreitung sogar bei mehr als 95 % (Tenzer, 2021b). Noch im Jahr 2012 nutzte nur etwas mehr als jede dritte befragte Person ein Smartphone (Tenzer, 2021a).

Besonderes Augenmerk liegt in dieser Arbeit darauf, nur handelsübliche Hardware zu verwenden, mit Sensoren, die mittlerweile – mit Ausnahme von günstigen Einsteigergeräten – zur Standardausstattung gehören. Dies schließt Technologien wie beispielsweise das seit dem Apple iPhone 11 verfügbare Ultra-Wide-Band oder spezielle Kameras zur Tiefenmessung (wie etwa im eingangs erwähnten Google Tango) aus. Ebenfalls keine nähere Betrachtung finden in dieser Arbeit zusätzliche am Fuß oder Körper befestigte Inertialsensoren. Diese bieten zwar prinzipbedingt eine hohe Genauigkeit (Godha & Lachapelle, 2008; Jimenez Ruiz, Seco Granja, Prieto Honorato & Guevara Rosas, 2012), sind jedoch nicht mit dem Grundsatz der breiten Verfügbarkeit vereinbar. Aus dem gleichen Grund finden auch Lösungen, die auf zusätzliche Infrastruktur innerhalb von Gebäuden wie etwa Bluetooth-

Beacons (siehe z. B. Xin-Yu Lin et al., 2015) angewiesen sind, keine Anwendung.

Als Entwicklungsplattform dient das mobile Betriebssystem Android, das sich gegenüber dem Konkurrenten iOS von Apple vor allem durch größere Offenheit im Hinblick auf den Zugang zu Sensorik, aber auch durch die Möglichkeit der plattformunabhängigen Entwicklung auszeichnet. Genauer zur Android-Plattform und zur Sensorik findet sich in Abschnitt 2.3.

Fußgängernavigation

Im Rahmen dieser Arbeit liegt der Fokus auf FußgängerInnen, die sich durch klar identifizierbare Schritte gehend fortbewegen. Dies ist eine Grundvoraussetzung für das später vorgestellte PDR-Verfahren, bedeutet allerdings gleichermaßen, dass andere Fortbewegungsarten wie beispielsweise Rollstühle nicht berücksichtigt werden können.

Die Navigation innerhalb des modellierten Indoor-Areals erfolgt entlang einer im Voraus geplanten und dem Positionierungssystem bekannten Route. Diese ist in Abschnitte aufgeteilt, mit jeweils korrespondierenden Navigationsanweisungen, die – sofern sie vom Benutzer befolgt werden – zu Fortschritt entlang der Route und schließlich zum Erreichen des Ziels führen.

1.4 Veröffentlichungen

Teilergebnisse dieser Dissertation wurden vorab in verschiedenen Zeitschriften und auf Konferenzen veröffentlicht. Weiterhin sind die Ergebnisse meiner 2016 der Fakultät für Sprach-, Literatur- und Kulturwissenschaft der Universität Regensburg vorgelegten Masterarbeit eingegangen.

- *Interaktive Indoor-Lokalisierung für Fußgänger*
Die erste Version des in Kapitel 3 beschriebenen graphbasierten Positionierungssystems wurde im Rahmen meiner Masterarbeit entwickelt und seitdem mehrfach überarbeitet (Jackermeier, 2016).
- *Task-Oriented Evaluation of Indoor Positioning Systems*
In einem Konferenzbeitrag auf der *Location Based Services (LBS) 2018* wurde gezeigt, dass das graphbasierte Positionierungskonzept unter kontrollierten Bedingungen funktioniert. Zudem wurde hier die aufgabenorientierte Evaluationsmethodik eingeführt (Jackermeier & Ludwig, 2018b).
- *Exploring the limits of PDR-based indoor localisation systems under realistic conditions*

Aufbauend auf dem Konferenzbeitrag wurden weitere Studien als Zeitschriftenartikel im *Journal for Location Based Services* veröffentlicht (Jackermeier & Ludwig, 2018a). Unter anderem wurde dabei gezeigt, wie durch periodisches Zurücksetzen der Position eine Langzeitstabilität erreicht werden kann.

- *Door Transition Detection for Long-Term Stability in Pedestrian Indoor Positioning*
Auf der Fachkonferenz IPIN 2019 wurde die Strategie für die Positionskorrektur beim Durchschreiten von Türen durch die Erkennung von Mustern in der Inertialsensorik vorgestellt (Jackermeier & Ludwig, 2019a).
- *User Behaviour in the Wild: Analysing Naturalistic Log Data of a Pedestrian Navigation App*
Auf der LBS 2019 wurden die Ergebnisse der Auswertung von naturalistischen Logdaten der Regensburger Campus-Navigations-App URwalking veröffentlicht (Jackermeier & Ludwig, 2019b). Der Fokus lag dabei auf dem Nutzerverhalten während der Navigation und dessen Auswirkungen auf die Positionsbestimmung. Teilergebnisse der Studie wurden zuvor bereits als Poster auf der COSIT (*Conference on Spatial Information Theory*) 2019 vorgestellt.
- *Smartphone-Based Activity Recognition in a Pedestrian Navigation Context*
Im Open-Access-Journal *Sensors* wurde ein Artikel zum Thema Aktivitätserkennung in der Fußgängernavigation veröffentlicht (Jackermeier & Ludwig, 2021). Darin wurde u. a. das Konzept der hierarchischen Klassifikation und die Anwendung der HAR-Modelle auf naturalistische Logdaten thematisiert.

Im folgenden Abschnitt wird deutlich, wie die soeben beschriebenen Publikationen den Verlauf der nächsten Kapitel bedingen.

1.5 Aufbau der Arbeit

Die Dissertation ist wie folgt gegliedert: In Kapitel 2 werden zunächst existierende Vorarbeiten aus dem URwalking-Projekt beschrieben, die für das Verständnis der Arbeit nötig sind, allen voran das graphbasierte Umgebungsmodell des Regensburger Campus. Weiterhin werden die technischen Grundlagen, insbesondere die verwendete Sensorik auf der Android-Plattform, vorgestellt.

Kapitel 3 beginnt mit einer Einführung in den aktuellen Forschungsstand in der Indoor-Positionierung. Danach folgt eine detaillierte Beschreibung der Basis-Version des entwickelten Positionierungssystems, welches mithilfe eines Partikelfilters und PDR die Position auf dem Graphmodell verfolgt. In den damit durchgeführten Vorstudien wird die grundlegende Funktionalität verifiziert und der realistische Experimentaufbau erprobt.

Anschließend liegt in Kapitel 4 der Fokus auf naturalistischen Studien zur Untersuchung des Benutzerverhaltens während der Fußgängernavigation. Hierfür werden die über einen Zeitraum von mehreren Semestern erhobenen Logdaten der Campus-Navigations-App URwalking analysiert, um die Anforderungen an ein Indoor-Positionierungssystem unter realen Bedingungen besser einschätzen zu können.

Kapitel 5 stellt darauf aufbauend HAR zur Erkennung von Aktivitäten während der Navigation vor. Anhand eines dafür erhobenen Datensatzes werden unterschiedliche HAR-Ansätze erprobt, um anschließend mit den resultierenden Modellen die naturalistischen Studien um einige Aspekte zu ergänzen. Die Integration der HAR-Modelle in das Indoor-Positionierungssystem führt schließlich wieder zur Kernthematik der Arbeit zurück. Eine abschließende Evaluation sowie detaillierte Fallstudien demonstrieren die Vorteile der HAR-Lösung.

In Kapitel 6 werden die Ergebnisse aller Studien im Bezug auf die Forschungsfragen der Arbeit reflektiert, um aus den so gewonnenen Erkenntnissen Empfehlungen für das Design eines Indoor-Positionierungssystems abzuleiten.

Den Abschluss bildet Kapitel 7 mit einer kurzen Zusammenfassung, der Einordnung des wissenschaftlichen Beitrags und einer Diskussion der Limitationen der Arbeit. Am Ende erfolgt ein Ausblick auf offene Probleme und mögliche Anknüpfungspunkte für zukünftige Arbeiten.

Kapitel 2

Vorarbeiten und technische Grundlagen

Die vorliegende Arbeit ist in einen größeren Forschungskontext an der Universität Regensburg eingebettet und entstand parallel zu mehreren Forschungsprojekten. Die von dort stammenden Vorarbeiten, auf denen diese Arbeit aufbaut, und die dadurch gegebenen Rahmenbedingungen werden in diesem Kapitel vorgestellt. Weiterhin werden technische Grundlagen erläutert, die für alle weiteren Kapitel von Relevanz sind.

Im Einzelnen handeln die folgenden Abschnitte von den Forschungstätigkeiten im Bereich Fußgängernavigation an der Universität Regensburg, insbesondere dem Projekt *URwalking* (Abschnitt 2.1) und der dort aufgebauten Datenbasis, die auch in dieser Arbeit Verwendung findet (Abschnitt 2.2). Anschließend wird die Android-Plattform, mit speziellem Fokus auf der dort verfügbaren Sensorik, vorgestellt (Abschnitt 2.3), welche die Basis für die mobile Campus-Navigations-App (Abschnitt 2.4) darstellt.

2.1 URwalking: Navigation auf dem Regensburger Campus

Der nach offiziellen Angaben 150 ha große Campus (Universität Regensburg, 2021) besteht aus mehreren großen Gebäudekomplexen sowie kleineren Gebäuden (siehe Abbildung 2.1). Überdachte Übergänge und Tiefstraßen ermöglichen es, die meisten Ziele auf dem Campus ohne das Betreten von Außenflächen zu erreichen. Die auf dem Luftbild erkennbare repetitive Anordnung der Gebäude, insbesondere im südlichen Teil des Campus, setzt sich im Innenbereich fort und erschwert somit die Orientierung und das Auffinden von Räumen.

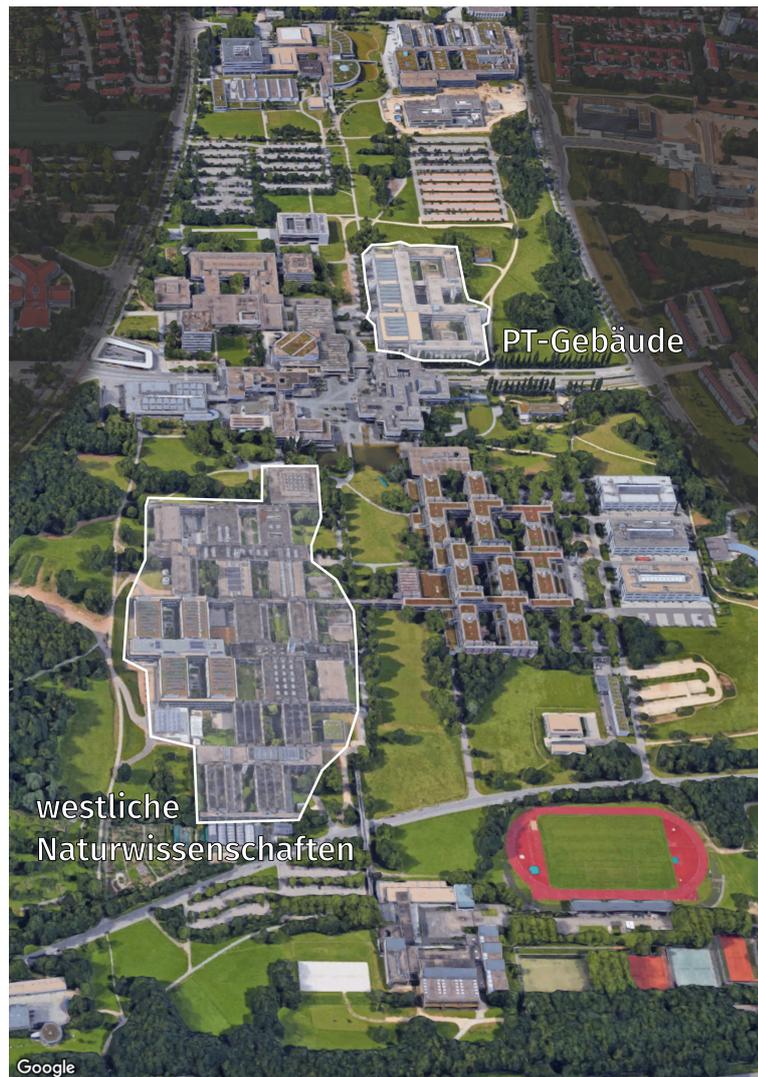


Abbildung 2.1: Luftbild des Regensburger Campus in Richtung Norden.
In den markierten Bereichen finden die später beschriebenen Experimente statt.
(Quelle: Google Maps; eigene Bearbeitung)

Der Regensburger Campus ist daher ein sehr gut geeignetes Areal für die Erforschung und Entwicklung von Fußgängernavigationssystemen. Am Lehrstuhl für Informationswissenschaft wird diese Thematik seit 2011 unter der Leitung von Bernd Ludwig im Rahmen des Projekts *URwalking* bearbeitet. Neben diversen Seminaren und Abschlussarbeiten wurden in diesem Kontext auch die Forschungsprojekte NADINE (»Navigation im öffentlichen Personenverkehr mit modularer Dienste-Architektur zur Einbindung in externe Applikationen«) und DIVIS (»Digitale Indoor-Verortung mittels Inertial-Sensorik und Landmarken«) durchgeführt. Im Laufe der Jahre entstand somit eine komplette Campus-Navigations-

lösung, bestehend aus detailliertem Umgebungsmodell, spezialisiertem Routing-server, Web-Applikation¹ und Android-App².

2.2 Umgebungsmodell: der Navigationsgraph

Das im Rahmen des URwalking-Projekts erstellte Modell des Regensburger Campus dient in dieser Arbeit als Datenbasis für das Positionierungssystem. Ursprünglich für die Pfadplanung entwickelt, bildet es die begehbaren Bereiche auf dem Campus in Form eines Graphen ab. Die nachfolgende Beschreibung beschränkt sich auf die für die Positionsbestimmung relevanten Aspekte, für weitere Details sei auf Ullmann (2020, S. 12–57) verwiesen.

2.2.1 Graphstruktur

Auf Grundlage der offiziellen Gebäudepläne und Grundrisse wurden die begehbaren Bereiche in den Gebäuden und Außenbereichen des Campus manuell mithilfe einer webbasierten Oberfläche annotiert (Ullmann, 2020, S. 53). Die daraus entstandene Graphstruktur, im Folgenden *Navigationsgraph* genannt, ist eine vereinfachte Repräsentation der Realität: An Entscheidungspunkten und möglichen Zielen befinden sich Knoten unterschiedlichen Typs, darunter Büros, Hörsäle, Raum- und Flurtüren sowie nicht näher definierte Stützpunkte (Ullmann, 2020, S. 29 ff.). Die Verbindungen dazwischen, d. h. die Korridore, Treppen oder auch die Wege in den Außenbereichen, sind als Kanten mit entsprechendem Typ modelliert (siehe Abbildung 2.2). Kanten verbinden nicht nur Knoten innerhalb einer Ebene, sondern können auch zwei Stockwerke miteinander verknüpfen. Ihr Typ gibt wiederum an, durch welche Art diese Verbindung stattfindet, beispielsweise durch Treppen, Aufzüge oder Rampen.

Größere, flächige Indoor-Areale, in denen für FußgängerInnen weitgehende Bewegungsfreiheit herrscht, stellen zunächst ein Problem für diese Art der Modellierung dar. Würden alle Zugänge – in manchen Fällen mehrere Dutzend – miteinander verbunden, zöge dies eine kombinatorische Explosion von teils nur geringfügig abweichenden Verbindungswegen mit sich. Daher ist es sowohl für die Routenberechnung als auch für die spätere Positionierung sinnvoll, diese Areale speziell zu behandeln. Technisch umgesetzt ist dies als ein zentraler Knoten (im Weiteren *Areanode* genannt), mit dem sternförmig alle Zugänge zur Fläche direkt verbunden sind (siehe z. B. links in Abbildung 2.2). Die konvexe Hülle der somit

¹<https://urwalking.ur.de>, abgerufen am 10.01.2022

²<https://play.google.com/store/apps/details?id=de.ur.urwalking.urwalking4>, abgerufen am 10.01.2022

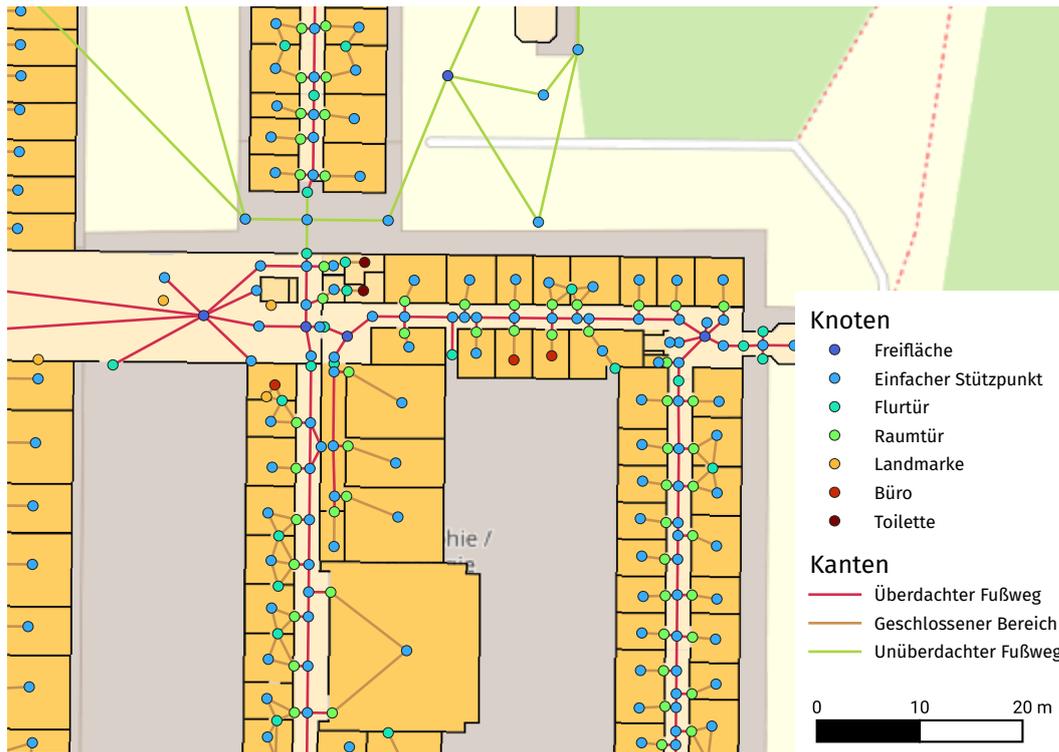


Abbildung 2.2: Visualisierung des graphbasierten Umgebungsmodells.
Kartenmaterial von OpenStreetMap (Haklay & Weber, 2008)

verbundenen Knoten spannt ein Polygon auf, innerhalb dessen freie Bewegung möglich ist (vgl. Ullmann, 2020, S. 46 f.).

Letztendlich entsteht eine hybride Graph-/Polygon-Repräsentation als Umgebungsmodell. Durch Verwendung dieser Abstraktion ist später keine Verortung an beliebigen Punkten im Raum möglich. Gleichzeitig führt diese Abstraktion zu weniger Komplexität und geringerem Rechenaufwand, da anstelle eines kontinuierlichen ein diskreter Raum zum Einsatz kommt.

2.2.2 Landmarkenbasierte Navigation

Wie eingangs in dieser Arbeit bereits erwähnt, liegt im URwalking-Projekt ein besonderer Fokus auf der Wegführung mithilfe von Landmarken, d. h. für den menschlichen Benutzer auffälligen Objekten, auf die in Navigationsanweisungen verwiesen werden kann (Golledge, 1991; Lynch, 1960). Nach Caduff und Timpf (2008, S. 250) kann jedes leicht erkennbare Objekt als Landmarke dienen, sofern es sich als Referenzpunkt eignet und salient ist, d. h. aufgrund seiner Eigenschaften relativ zur Umgebung hervorsteht. Der Landmarkenbegriff wird beispielsweise von H. Wang, Elgohary und Choudhury (2012) auch im Zusammenhang mit der

Indoor-Positionierung verwendet und meint dann diejenigen Orte, die mithilfe von Sensorik erkennbar sind. Für die vorliegende Arbeit ist vor allem die Schnittmenge dieser Definitionen relevant, d. h. Landmarken wie etwa Durchgangstüren oder Treppen, die im Verlauf der Route passiert werden müssen und spezifische Bewegungsmuster verursachen.

2.2.3 Technische Umsetzung

Koordinatensysteme und Geo-Referenzierung

Die Knoten im Navigationsgraph wurden ursprünglich lokal anhand ihrer x- und y-Koordinaten auf den aus den Gebäudeplänen generierten Bilddateien verortet. Um globale Koordinaten zu erhalten, wurden einige zusätzliche Knoten manuell georeferenziert. Mithilfe der dadurch berechneten Transformationsmatrizen erfolgte für jedes Stockwerk eine Umrechnung von lokalen in Gauß-Krüger-Koordinaten (EPSG:31468) (vgl. Ullmann, 2020, S. 47–52). Ein Vorteil dieser Repräsentation für die spätere Verwendung im Positionierungssystem ist, dass die Position in Form von Rechtswert und Hochwert in einem kartesischen Koordinatensystem in der Einheit Meter angegeben ist. Auf diese Weise können in der gleichen Einheit ausgedrückte Schrittvektoren ohne weitere Konvertierung direkt addiert werden. Globale GPS-Koordinaten (EPSG:4326) werden intern im Positionierungssystem nicht verwendet, dienen jedoch als Schnittstelle für die Darstellung der Position auf der Karte.

Daten-Repräsentation

Die durch das Modellierungswerkzeug erstellte Datenbasis liegt ursprünglich im XML-Format vor (Ullmann, 2020, S. 41 f.). In einer hierarchischen Anordnung ist jedes Gebäude zunächst in einer separaten Datei mit dem Wurzelement `graph` gespeichert, die die Stockwerke (`level`) des Gebäudes enthält, in welchen sich wiederum alle dortigen Knoten (`node`) und Kanten (`edge`) befinden. Verbindungen zwischen Stockwerken werden mithilfe von `levelEdge`-Tags auf der Hierarchieebene der Stockwerke realisiert, während Verbindungen zwischen Gebäuden mithilfe spezieller Knotentypen (`link`) umgesetzt sind. Je Stockwerk sind zusätzlich die zur Umrechnung der Bildkoordinaten nötigen Elemente der Transformationsmatrizen enthalten. Listing 2.1 veranschaulicht diesen Aufbau an einem gekürzten Beispiel (adaptiert von Ullmann, 2020, S. 42–46).

Die Repräsentation der Daten im XML-Format ist für den Einsatz auf Mobilgeräten nicht optimal, da bei jedem Start der Anwendung aufwändiges Parsen nötig ist, um auf die enthaltenen Informationen zuzugreifen. Für eine kompaktere und

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<graph>
  <level id="1" storey="0" mapfile="pt-eg.jpg" width="1754" height="2481">
    <node id="1" x="1004" y="1469" type="Office" name="Beispielbüro" roomid="3.0.67 A"/>
    <node id="2" x="1004" y="1488" type="Entry"/>
    <edge id="1" source="1" sink="2" type="1"/>
    <node id="1" x="70" y="80" type="link" filename="Campusplan" level="1" nodeid="407"/>
    <!-- ... -->
    <pixelToWGS84>
      <xlat>-7.305428E-9</xlat>
      <ylat>-8.2814381E-7</ylat>
      <wlat>49.000349</wlat>
      <xlon>1.2586323E-6</xlon>
      <yln>-9.9242146E-9</yln>
      <wlon>12.095523</wlon>
    </pixelToWGS84>
    <pixelToGK>
      <xrw>0.092100550</xrw>
      <yrr>-0.000612225</yrr>
      <wrw>4507095.63</wrw>
      <xhw>-0.000612225</xhw>
      <yhw>-0.092100550</yhw>
      <whw>5429228.05</whw>
    </pixelToGK>
  </level>
  <level id="2" storey="1" mapfile="pt-og1.jpg" width="1754" height="2481">
    <node id="1" x="1024" y="469" type="1"/>
    <!-- ... -->
  </level>
  <leveledge id="1" sourcelevel="2" sinklevel="1" sourcenode="1" sinknode="1" type="Stairs"/>
  <!-- ... -->
</graph>
```

Listing 2.1: Gekürztes Beispiel der XML-Repräsentation eines Gebäudes

performantere Repräsentation der Daten wurde im Rahmen des DIVIS-Projekts ein relationales Datenbankschema entwickelt. Zum Einsatz kommt dabei das von Android standardmäßig unterstützte SQLite in Version 3.

Die Konvertierung erfolgt mithilfe eines Programms, welches aus der hierarchischen XML-Struktur (Gebäude \Rightarrow Stockwerke \Rightarrow Knoten/Kanten) die relationale Struktur erzeugt. An dieser Stelle findet ebenso die erwähnte Umrechnung in Gauß-Krüger-Koordinaten statt, mit denen das Positionierungssystem intern arbeitet, wodurch sich die Anzahl der zur Laufzeit nötigen Konvertierungen zwischen Koordinatensystemen verringert.

Abbildung 2.3 zeigt das resultierende Datenbankschema in leicht vereinfachter Form. Wichtigste strukturelle Änderung gegenüber dem Ausgangszustand ist die Verflachung der Hierarchie, infolge derer auch Stockwerks- und Gebäudeverbindungen als reguläre Kanten (mit entsprechendem Typ) modelliert sind. Dies erleichtert im weiteren Verlauf das Durchschreiten des Graphs während der Positionsbestimmung.

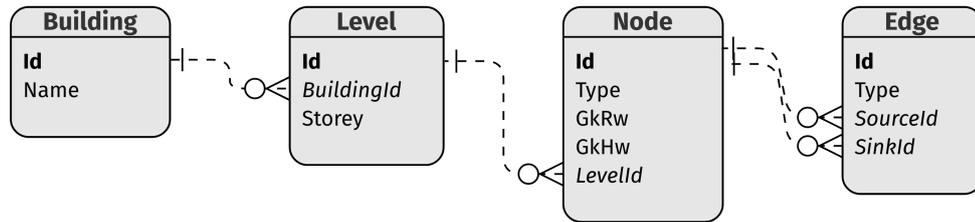


Abbildung 2.3: SQLite-Schema des Navigationsgraphs

2.3 Plattform und Sensorik

Neben den Vorarbeiten aus dem URwalking-Projekt stellen auch die mobilen Plattformen und die dort verfügbare Sensorik Randbedingungen für diese Arbeit dar.

2.3.1 Das Android-Betriebssystem

Android und iOS sind zum aktuellen Zeitpunkt mit 72,2 % respektive 27,0 % weltweitem Marktanteil die beiden einzigen relevanten mobilen Betriebssysteme (Stat-Counter, 2021). Die Forschung in dieser Arbeit findet mithilfe der Android-Plattform statt, welcher aufgrund ihrer größeren Offenheit der Vorzug gegeben wurde. Dies schließt auch die plattformunabhängige Unterstützung von Entwicklertools für Windows, Linux und macOS sowie die bessere Verfügbarkeit von günstigen Entwicklergeräten ein.

2.3.2 Sensoren im Detail

Im Folgenden werden die für die Indoor-Positionierung relevanten Sensoren der Android-Plattform näher beschrieben. Die Anzahl der unterstützten Sensortypen hat im Laufe der Zeit deutlich zugenommen, von ursprünglich sechs Sensoren in API-Level 3 (Android 1.5) auf mehr als zwei Dutzend im aktuellen API-Level 30 (Android 11) (Google Developers, 2021d).

Zu unterscheiden ist dabei zwischen zwei Arten (vgl. Google LLC & Open Handset Alliance, 2020): Hardware-Sensoren (*base sensors*) wie etwa der Beschleunigungssensor, das Gyroskop oder der Magnetfeldsensor messen grundlegende physikalische Phänomene. Darauf aufbauend sind eine Vielzahl von Sensoren verfügbar, welche per Software die Messwerte der zugrundeliegenden Sensoren verarbeiten (sog. *composite sensors*). Beispiele hierfür sind der Schritterkenner oder

der Rotationsvektor. Die Entwicklung von Algorithmen zur Low-Level-Sensorfusion ist nicht Teil dieser Arbeit, stattdessen wird soweit wie möglich auf diese existierenden Software-Sensoren zurückgegriffen.

Die Messwerte der Sensoren beziehen sich auf ein lokales Koordinatensystem, dessen x-Achse bei frontaler Betrachtung des Geräts von links nach rechts und dessen y-Achse von unten nach oben verlaufen. Die z-Achse verläuft senkrecht zur Bildelebene und zeigt in Richtung des Betrachters (siehe Abbildung 2.4).

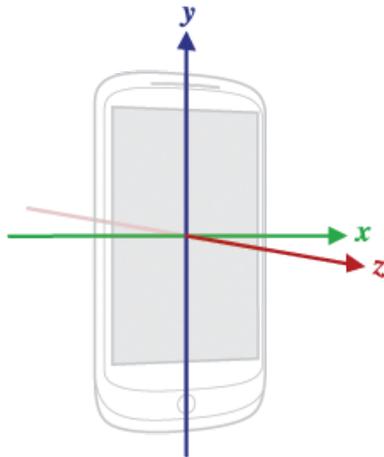


Abbildung 2.4: Von der Sensor-API verwendetes Koordinatensystem
(Google LLC & Open Handset Alliance, 2020)

Der Zugriff auf manche Sensortypen setzt zusätzliche Berechtigungen voraus, die vom Benutzer gewährt werden müssen. Für die zur absoluten Positionsbestimmung fähigen Sensoren bzw. Technologien WLAN, Bluetooth und GNSS sind die Berechtigungen `FINE_LOCATION` bzw. `COARSE_LOCATION` erforderlich. Ab API-Level 30 ist zusätzlich `BACKGROUND_LOCATION` nötig, um die Positionierung aufrechtzuerhalten, wenn die Anwendung nicht im Vordergrund läuft. Für den Zugriff auf den Schritterkenner wird ab API-Level 29 (Android 10) die Berechtigung `ACTIVITY_RECOGNITION` vorausgesetzt.

Die konkrete Implementierung in einer Android-App erfolgt mithilfe des Systemdiensts `SensorManager`, welcher über die Funktion `getDefaultSensor` den korrespondierenden Sensor zurückgibt. Für diesen können anschließend mithilfe der Funktion `registerListener` Updates bezogen werden. Das `SensorEventListener`-Interface erfordert das Überschreiben der beiden Funktionen `onSensorChanged` und `onAccuracyChanged`, welche die eigentlichen Sensordaten respektive Informationen über deren Genauigkeit liefern.

Die gemeldete Genauigkeit wird in mehreren Stufen angegeben, wobei eine 0 (`SENSOR_STATUS_UNRELIABLE`) auf einen völlig unzuverlässigen Sensor hinweist und 1 bis 3 für niedrige (`SENSOR_STATUS_ACCURACY_LOW`), mittlere (`SENSOR_`

STATUS_ACCURACY_MEDIUM) und hohe (SENSOR_STATUS_ACCURACY_HIGH) Genauigkeit stehen (Google Developers, 2021c).

Beim Registrieren der Sensor-Updates kann mithilfe des Parameters `samplingPeriodUs` die gewünschte Aktualisierungsrate explizit in Mikrosekunden oder mithilfe von Konstanten (`SENSOR_DELAY_NORMAL`, `SENSOR_DELAY_UI`, `SENSOR_DELAY_GAME` oder `SENSOR_DELAY_FASTEST`) angegeben werden. Seitens des Systems ist jedoch nicht garantiert, dass die so festgelegte Rate erreicht oder konstant eingehalten wird. Die maximale Updaterate variiert zudem je nach Sensortyp und konkret verbauter Gerätehardware.

Nach diesen allgemeinen Erläuterungen folgt nun der Überblick über die einzelnen Sensoren:

Beschleunigungssensor

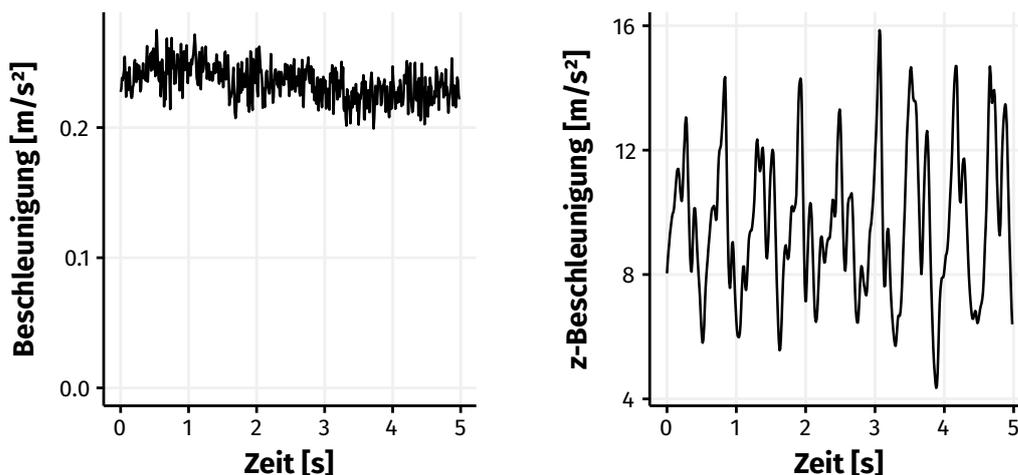
Den ersten Teil der Inertial Measurement Unit (IMU) bildet der Beschleunigungssensor. Das reguläre Accelerometer (in der Android-API als `TYPE_ACCELEROMETER` bezeichnet) gibt die Beschleunigung inklusive der Erdbeschleunigung entlang der drei Achsen des Geräts (siehe Abbildung 2.4) an. Treten außer der Erdbeschleunigung keine weiteren Kräfte auf, kann somit auch die Lage, d. h. der Neigungs- und Rollwinkel des Geräts, bestimmt werden. Daneben existiert als Software-Sensor das sog. lineare Accelerometer (`TYPE_LINEAR_ACCELERATION`), welches die reine Beschleunigung des Geräts ohne die Erdbeschleunigung angibt.

Zu beachten ist, dass die Genauigkeit der verbauten Sensoren nicht dazu ausreicht, durch Integration von der Beschleunigung auf die Geschwindigkeit oder von dort aus gar – durch nochmaliges Integrieren – auf die Position zu schließen. Dies lässt sich am anschaulichsten experimentell dadurch zeigen, dass der Betrag der Beschleunigung eines still liegenden Geräts in der Regel ungleich der Erdbeschleunigung von $9,81 \text{ m/s}^2$ ist bzw. die lineare Beschleunigung ungleich 0 m/s^2 . Ein Beispiel hierfür ist in Abbildung 2.5a zu sehen.

Aufgrund der beim Gehen auftretenden periodischen Beschleunigungskräfte (Abbildung 2.5b) eignet sich das Accelerometer für die Implementierung von Schritterkennungs-Methoden. Im Rahmen dieser Arbeit wird eine derartige Rückfalllösung immer dann verwendet, falls kein nativer Sensor für die Schritterkennung vorhanden ist. Weiterhin dienen die drei Achsen des linearen Accelerometers später als Features für die Aktivitätserkennung.

Gyroskop

Als zweiter Teil der IMU erlaubt das Gyroskop (`TYPE_GYROSCOPE`) die Erkennung von Rotationen um die drei Geräteachsen, die Messwerte liegen in der Einheit



(a) Magnitude der linearen Beschleunigung bei stationärem Gerät

(b) Beschleunigung in z-Richtung während des Gehens

Abbildung 2.5: Beispieldaten des Beschleunigungssensors

rad/s vor.

Prinzipbedingt reagiert das Gyroskop sehr responsiv auf Änderungen der Geräteausrichtung. Wird die Winkelgeschwindigkeit des Gyroskops zur Berechnung der Geräteausrichtung verwendet, tritt wiederum die oben thematisierte Problematik der Integration auf. Bereits geringe Ungenauigkeiten äußern sich dann in einer driftenden Richtungsangabe.

Im weiteren Verlauf dieser Arbeit werden die drei Achsen des Gyroskops als Features für die Aktivitätserkennung verwendet.

Magnetfeldsensor

Der Magnetfeldsensor (TYPE_MAGNETIC_FIELD) liefert die Richtung und Stärke des umgebenden Magnetfelds entlang der drei Geräteachsen in der Einheit μT . Im Idealfall bestehen die Messwerte vorrangig aus dem Erdmagnetfeld, wodurch sie wie bei einem herkömmlichen Magnetkompass zur Bestimmung der Geräteausrichtung in Bezug auf den magnetischen Nordpol verwendet werden können. Störende Einflüsse, etwa durch Stahlbeton, Metalltüren oder Aufzugschächte, sorgen jedoch dafür, dass der Magnetfeldsensor für die Richtungsbestimmung im Innenbereich nur bedingt geeignet ist (Subbu, Gozick & Dantu, 2013).

Im Vergleich zum Gyroskop reagiert der Magnetfeldsensor deutlich träger auf Richtungsänderungen. Die gemessene Feldstärke fließt später als Feature in die Aktivitätserkennung mit ein.

Rotationsvektor und Game-Rotationsvektor

Um die spezifischen Nachteile von Accelerometer, Gyroskop und Magnetometer auszugleichen, stellt Android den sog. Rotationsvektor-Sensor (`TYPE_ROTATION_VECTOR`) bereit, in dem per Software die Messwerte der obigen Sensoren integriert werden. Unter Idealbedingungen wird die absolute Ausrichtung des Geräts anhand des Magnetometers bestimmt, während schnelle Bewegungen mittels des Gyroskops und Accelerometers erkannt werden können. Als Messwert liefert der Rotationsvektor vier Komponenten eines Quaternions, aus welchen die Ausrichtung des Geräts in einem globalen Koordinatensystem berechnet werden kann.

In der Praxis ist die Inklusion des Magnetfeldsensors im Rotationsvektor oft problematisch. Fehlerhafte Kalibrierung oder die oben angesprochenen störenden Umgebungseinflüsse können falsche Richtungsangaben verursachen. Ein Beispiel dafür zeigt Abbildung 2.6 aus einem später detaillierter beschriebenen Positionierungs-Experiment (Abschnitt 3.4.3). Die in blau abgebildeten Trajektorien wurden durch Addition der rohen Schrittvektoren erzeugt, deren Richtung wiederum auf dem Rotationsvektor basiert. Wie zu sehen ist, existiert eine große Diskrepanz zwischen den so aufbereiteten Daten und dem Routenverlauf (nummerierte Abschnitte). Dies gilt nicht nur im Hinblick auf die absolute Richtung, sondern auch bezogen auf die Wiedergabetreue der Abbiegungen und der geraden Abschnitte.

Abhilfe schafft der sog. Game-Rotationsvektor-Sensor (`TYPE_GAME_ROTATION_VECTOR`, im Folgenden als GRV bezeichnet), welcher ohne den Magnetfeldsensor operiert und lediglich Accelerometer und Gyroskop fusioniert. Mit ihm kann keine absolute Richtung mehr festgestellt werden, die Schätzung der relativen Richtungsänderung ist jedoch unabhängig von externen Faktoren. In Abbildung 2.6 sind die mithilfe des GRV entstandenen Trajektorien in Grüntönen dargestellt. Ohne den störenden Einfluss des Magnetfeldsensors wird der Routenverlauf – unter Nichtbeachtung der absoluten Richtung – deutlich besser approximiert.

Im weiteren Verlauf dieser Arbeit wird der GRV für die Bestimmung der Schritt-richtung und als Feature für die Aktivitätserkennung verwendet.

Schrittkenner

Der Schrittkenner (`TYPE_STEP_DETECTOR`) ist ein auf dem Accelerometer basierender Software-Sensor, der bei jedem gegangenen Schritt auslöst. Schritt-richtung und Länge werden dabei nicht erfasst und müssen separat bestimmt werden.

Unter der Annahme, dass eine vom Hersteller auf das jeweilige Gerät abgestimmte Lösung zuverlässiger arbeitet als eine eigene Implementierung auf Basis des Accelerometers, wird der dedizierte Schrittkenner im Rahmen dieser Arbeit

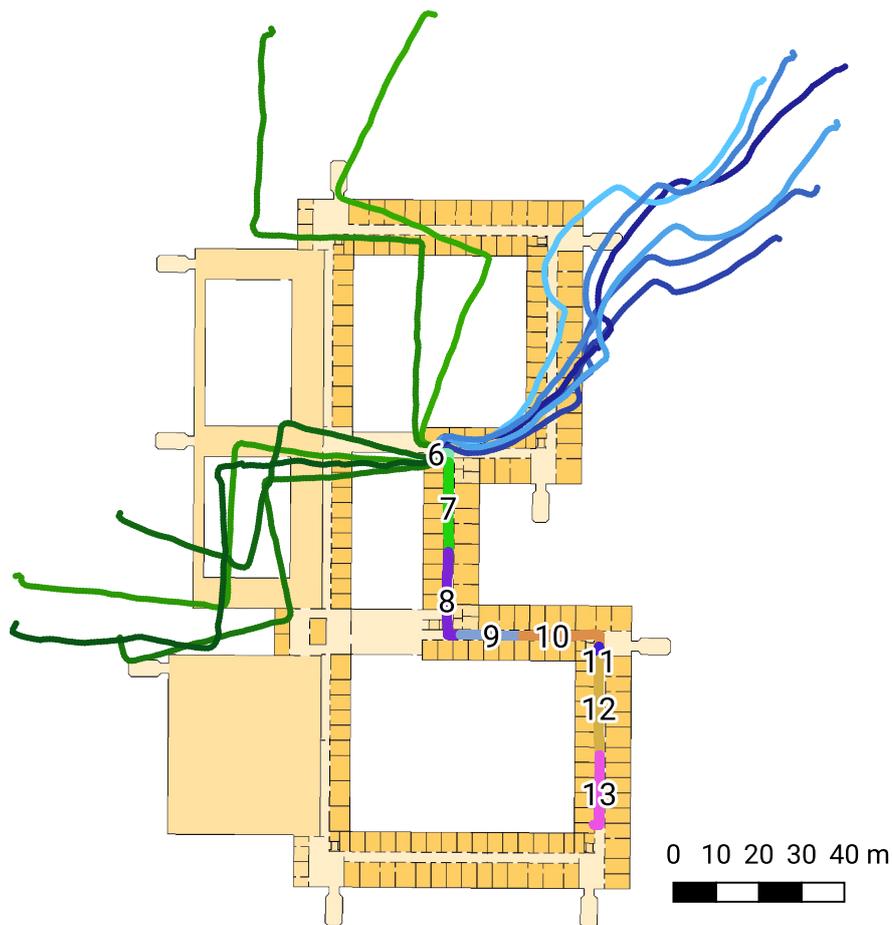


Abbildung 2.6: Rohe Schritt-Trajektorien des Rotationsvektors (blau) und des Game-Rotationsvektors (grün) sowie die tatsächlich gegangene Route

in den kontrollierten Experimenten als zentrales Element des PDR-Verfahrens (siehe Abschnitt 3.2.1) verwendet.

Barometer

Mit dem Barometer (TYPE_PRESSURE) wird der Luftdruck in hPa bestimmt, welcher Rückschlüsse auf Höhenänderungen – etwa zwischen verschiedenen Stockwerken – zulässt. Das Beispiel in Abbildung 2.7 zeigt Luftdruckdaten, die beim Treppensteigen (zwei Stockwerke nach unten) aufgezeichnet wurden. Darin ist gut zu sehen, wie nach etwa 10s der ursprünglich nahezu konstante Luftdruck steigt, bis ca. bei Sekunde 40 wieder ein Plateau erreicht wird. Dazwischen sind ebenso die Treppenabsätze auf halber Höhe sowie das Zwischenstockwerk erkennbar.

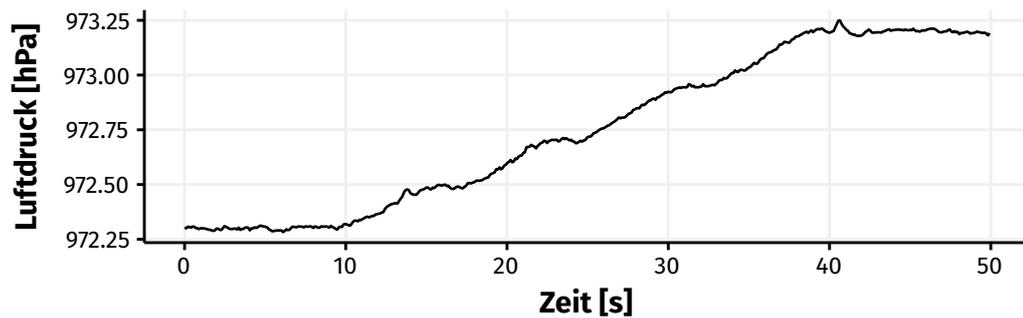


Abbildung 2.7: Luftdruckveränderung beim Wechseln von Stockwerken

Im Rahmen dieser Arbeit werden mit dem Barometer relative Luftdruckveränderungen gemessen, welche später als Feature für die Aktivitätserkennung dienen.

Funktechnologien

Der Vollständigkeit halber seien hier weitere Technologien aufgeführt, die häufig in Positionierungssystemen zum Einsatz kommen, in dieser Arbeit jedoch aufgrund der beschriebenen Randbedingungen nur eine geringe Rolle spielen.

Auf der Android-Plattform ist es – die entsprechenden Berechtigungen vorausgesetzt – grundsätzlich möglich, auf WLAN-Signalstärken zuzugreifen, welche anschließend zur Positionsschätzung verwendet werden können (mehr dazu in Abschnitt 3.1.2). In neueren Android-Versionen wurde die Möglichkeit, selbst WLAN-Scans auszulösen, jedoch deutlich eingeschränkt. Aktuell (seit Android-Version 10) besteht ein Limit von vier Scans innerhalb eines Zeitraums von zwei Minuten, welches sich lediglich zu Entwicklungszwecken aufheben lässt (Google Developers, 2021e). Der Nutzen für eine breit einsetzbare Positionierungslösung ist daher beschränkt. Im URwalking-System findet WLAN lediglich bei der initialen Positionsbestimmung Anwendung.

Die Nichtverfügbarkeit satellitengestützter Systeme innerhalb von Gebäuden motiviert nicht nur diese Arbeit, sondern den Forschungsbereich Indoor-Positionierung generell. Die Positionsbestimmung erfolgt anhand der Laufzeitunterschiede der von den Satelliten ausgesendeten Signale, weswegen für optimale Funktion eine ungestörte Sichtverbindung zu den Satelliten erforderlich ist. Da dies innerhalb von Gebäuden nicht der Regelfall ist, kann die Position nicht oder nur mit geringer Genauigkeit berechnet werden. Während GNSS in reinen Indoor-Systemen also nicht sinnvoll einsetzbar ist, kann in gemischten Indoor-/Outdoor-Umgebungen damit beispielsweise das Betreten von Gebäuden erkannt werden, um den Startpunkt für die Indoor-Positionierung zu bestimmen.

Auf der Android-Plattform ist es unüblich, mit rohen GNSS-Daten zu arbei-

ten. Stattdessen wird üblicherweise der von den *Google Play Services* bereitgestellte *Fused Location Provider* (Google Developers, 2021a) für die Positionsbestimmung verwendet. In diesem werden aus Gründen der Energieeffizienz die Signale unterschiedlicher Sensoren, darunter GNSS und WLAN, in einem proprietären Verfahren fusioniert.

Fazit

Wie dieser technische Überblick zeigt, besitzt jeder Sensortyp eigene Vor- und Nachteile, die ihn für bestimmte Positionierungsmethoden geeignet machen. Beispiele aus der Literatur für den konkreten Einsatz in Positionierungssystemen werden in Abschnitt 3.1 beschrieben.

Die tatsächliche Verfügbarkeit der vorgestellten Sensoren ist abhängig vom spezifischen Endgerät und der Android-Version. Ein Überblick über die Verfügbarkeit unter der Nutzerschaft der entwickelten Campus-Navigations-App erfolgt in Abschnitt 4.4.4.

2.3.3 Testgeräte

Für den Großteil der im weiteren Verlauf der Arbeit beschriebenen Studien dienten Google-Pixel-Smartphones als Testgeräte. Die Pixel-Serie löste 2016 die ebenfalls von Google mitentwickelten Nexus-Geräte ab und kann als Referenzimplementierung für die Android-Plattform betrachtet werden.

Die wichtigsten technischen Daten bilden ein Vierkern-Prozessor (Qualcomm Snapdragon 821), 4 GB Arbeitsspeicher sowie ein 5,0 Zoll großer Full-HD-Bildschirm (Google, 2021a). Weiterhin sind alle im vorigen Abschnitt beschriebenen und für die Indoor-Positionierung benötigten Sensoren vorhanden. Softwareseitig wurden die Geräte mit Android 7.1 ausgeliefert. Für die Experimente wurde die jeweils aktuelle Android-Version verwendet.

Die damalige gehobene Hardware-Ausstattung des ersten Google-Pixel-Smartphones ist im Vergleich mit aktuellen Geräten nur noch im Mittelfeld anzusiedeln. Für die Entwicklung und Evaluation möglichst breit anwendbarer Positionierungslösungen ist es daher gut geeignet.

2.4 Die URwalking-App für Android

Die Campus-Navigations-App für die Android-Plattform stellt innerhalb des URwalking-Projekts neben der webbasierten Version ein zusätzliches Assistenzangebot dar. Größter Vorteil der nativen Umsetzung als Android-App gegenüber einer

mobilen Web-App ist die Möglichkeit, auf Gerätesensoren zuzugreifen und somit ein Indoor-Positionierungssystem zu integrieren. Die App wurde ursprünglich als Prototyp für die Verwendung in einem Forschungsseminar entwickelt und anschließend für die Veröffentlichung im Google Play Store vorbereitet, wo sie seit Herbst 2018 verfügbar ist. Zu Beginn des Wintersemesters 2021/22 existieren laut Play-Store-Statistik auf ca. 1700 Geräten aktive Installationen. Der Großteil der Implementierung stammt von mir als Autor; einige Features wurden von Michael Meckl als studentische Hilfskraft beigesteuert.

Die URwalking-App dient nicht nur als Testplattform für die in Kapitel 3 vorgestellte Indoor-Positionierungslösung, sondern auch zur Erforschung des Navigationsverhaltens unter realistischen Bedingungen und dessen Implikationen für den Betrieb eines Positionierungssystems (siehe Kapitel 4). Zu diesem Zweck werden die BenutzerInnen beim ersten Start der Anwendung gefragt, ob sie der freiwilligen und anonymen Übermittlung ihrer Sensor- und Nutzungsdaten zustimmen.

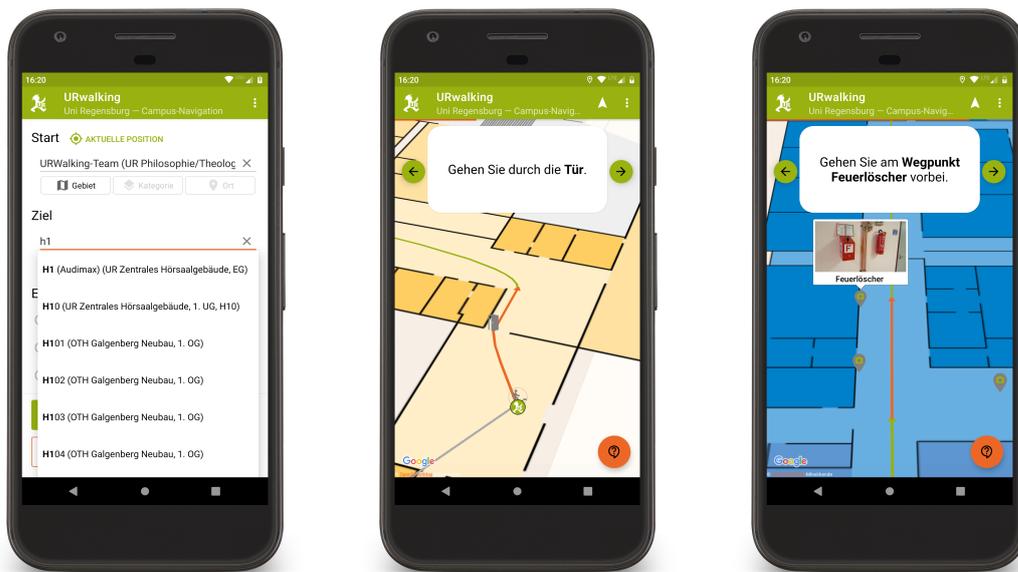
2.4.1 Funktionsweise und Funktionsumfang

Der Funktionsumfang der App entspricht größtenteils der Web-Applikation. NutzerInnen sind in der Lage, nach Zielen auf dem Campus zu suchen (siehe Abbildung 2.8a) oder diese in einer nach Gebäuden und Kategorien geordneten Hierarchie zu selektieren. Die Startposition wird entweder ebenfalls manuell vom Benutzer ausgewählt oder aber anhand der vom Android-System geschätzten Position grob automatisch bestimmt. Weiterhin befinden sich auf dem Regensburger Campus verteilt georeferenzierte QR-Codes, welche beim Scan auf die URwalking-Webseite bzw. -App führen und dort den aktuellen Standort als Startpunkt vorauswählen. Eine Reihe von optional selektierbaren Präferenzen ermöglicht u. a. das Bevorzugen von barrierefreien oder im Innenbereich gelegenen Routen.

Nach Auswahl von Start und Ziel und der anschließenden Routenberechnung beginnt der eigentliche Navigationsvorgang. Die Benutzeroberfläche teilt sich grob in die Kartendarstellung und die aktuelle Navigationsanweisung auf. Der momentan zu absolvierende Routenabschnitt ist farblich hervorgehoben (Abbildung 2.8b). Von den Anweisungen referenzierte Landmarken sind auf der Karte markiert und können durch Antippen als Foto dargestellt werden (Abbildung 2.8c).

Durch Drücken der seitlichen Pfeil-Buttons oder durch eine Wischgeste kann jederzeit zwischen den Routenanweisungen gewechselt werden, wodurch sich auch der Kartenausschnitt dorthin bewegt. Letzterer kann während der Navigation innerhalb der Grenzen des Campus frei bewegt werden, auch das Zoomen oder Schrägstellen der Karte mit Zwei-Finger-Gesten ist möglich.

Die Funktionalität wurde später durch einen dedizierten Kartenmodus erwei-



(a) Zielsuche mit Auto-Vervollständigung

(b) Route und Benutzerposition

(c) Anzeige von Landmarkenfotos

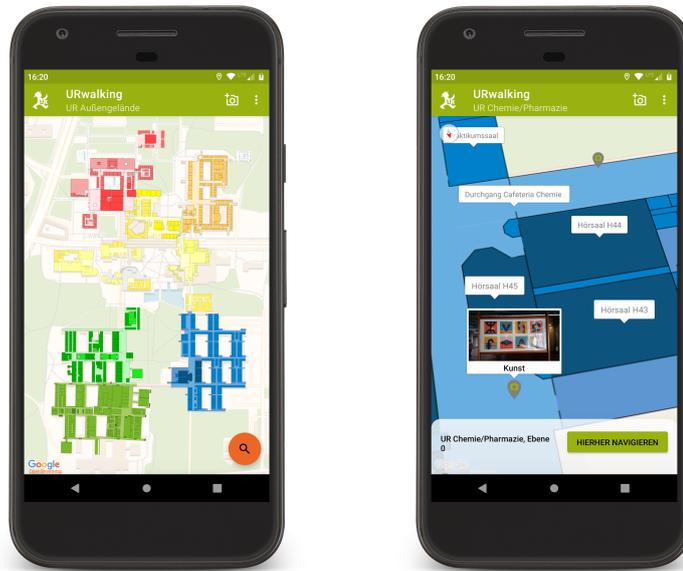
Abbildung 2.8: Screenshots der URwalking-App (Suche und Navigation)

tert, der auch ohne Eingabe von Start und Ziel die Orientierung auf dem Campus ermöglicht (Abbildung 2.9a). Darin können Räume, Landmarken und beliebige andere Orte betrachtet und als Ziel für die Navigation ausgewählt werden (Abbildung 2.9b).

2.4.2 Technische Umsetzung

Die App ist größtenteils in der Programmiersprache Kotlin geschrieben, welche sich neben Java seit der offiziellen Unterstützung durch Google im Jahr 2017 für die Android-Entwicklung etabliert hat. Während der Entwicklung wurde darauf geachtet, wo immer möglich die offiziellen Gestaltungsrichtlinien zu beachten. Im Sinne einer einheitlichen User Experience wurde beispielsweise die Kartendarstellung mithilfe der Google-Maps-Bibliothek für Android gelöst, welche die bekannten Gesten für das Manipulieren der Karte unterstützt. Da die Außenbereiche des Campus in OpenStreetMap detaillierter modelliert sind, werden diese als Overlay über der Google-Karte angezeigt. Darüber wiederum befindet sich das URwalking-Indoor-Kartenmaterial.

Der Navigationsgraph wird in Form der in Abschnitt 2.2.3 beschriebenen SQLite-Datenbank mit der App ausgeliefert und kann durch einen Update-Mechanismus aktualisiert werden. Die Routenberechnung findet auf dem URwalking-Pfad-



(a) Übersicht über den Campus

(b) Detailansicht:
Landmarken und Räume

Abbildung 2.9: Screenshots der URwalking-App (Kartenmodus)

planungs-Server (Ullmann, 2020, S. 96) statt, mit dem über die bereitgestellte JSON-Schnittstelle kommuniziert wird. Die URwalking-App enthält prototypisch das im Rahmen dieser Arbeit entwickelte Indoor-Positionierungssystem, das im folgenden Kapitel näher beschrieben wird.

Kapitel 3

Graphbasierte Indoor-Positionierung

Nachdem nun die Randbedingungen der Arbeit bekannt sind – insbesondere der Navigationsgraph als Umgebungsmodell für die URwalking-Navigation und die verfügbare Sensorik auf der Android-Plattform –, widmet sich dieses Kapitel der graphbasierten Indoor-Positionierung.

Zunächst wird in einem Überblick über die Literatur der Stand der Technik in der Indoor-Positionierung vorgestellt, mit besonderem Fokus auf das im Anwendungsgebiet der Fußgängernavigation vorherrschende PDR-Prinzip. Anschließend erfolgt die Beschreibung des in dieser Arbeit entwickelten Basis-Systems, bestehend aus dem an den Navigationsgraph adaptierte PDR-Verfahren und der darauf operierenden Partikelfilter-Implementierung. In zwei Evaluationsstudien werden schließlich die grundlegende Funktionsweise des Verfahrens verifiziert und das realistische Experiment-Setup sowie einige Anpassungen an die spezifischen Gegebenheiten der Indoor-Positionierung erprobt.

3.1 Forschungsstand in der Indoor-Positionierung

Wie eingangs in dieser Arbeit erwähnt, besteht ein Bedarf nach Indoor-Positionierungslösungen in verschiedensten Kontexten. Je nach Umgebung und Einsatzgebiet herrschen unterschiedliche Voraussetzungen u. a. hinsichtlich der verfügbaren Technologien, des finanziellen Aufwands und der Möglichkeit zur Installation zusätzlicher Infrastruktur.

In diesem Überblick über die vorhandene Literatur liegt der Fokus auf Indoor-Positionierungsverfahren für die Fußgängernavigation unter Berücksichtigung der in den vorigen Kapiteln etablierten Randbedingungen. Kurz zusammen-

gefasst schließt dies die Verwendung handelsüblicher Smartphones als Sensorik-Plattform, die Beschränkung auf bereits vorhandene Gebäude-Infrastruktur sowie ein graphbasiertes Umgebungsmodell ein. Verfahren, die diese Kriterien nicht erfüllen, werden im Folgenden nur kurz thematisiert, um diese Arbeit innerhalb des umfangreichen Forschungsbereichs Indoor-Positionierung zu verorten.

In den folgenden Abschnitten wird zunächst der Stand der Technik in der infrastrukturfreien Positionierung mittels PDR beschrieben. Darauf aufbauend folgt die Vorstellung von hybriden Verfahren, welche zusätzliche Sensoren oder Informationen für die Positionierung heranziehen. Der letzte Abschnitt widmet sich der Evaluationsmethodik für Positionierungssysteme und vergleicht häufig in der Literatur verwendete Experiment-Setups und Distanzmetriken.

3.1.1 Infrastrukturfreie Positionierung mit PDR

Infrastrukturfreie Positionierungsverfahren zeichnen sich nach Winter et al. (2019, S. 61:2) dadurch aus, dass außer dem zu ortenden Gerät – im vorliegenden Fall das handelsübliche Smartphone – keine externe, in der Umgebung verbaute Sensorik oder Hardware benötigt wird. Dies schließt ebenfalls den Anspruch ein, die Position ohne eine Internetverbindung, d. h. nur durch die auf dem Gerät vorhandene Rechenleistung zu bestimmen.

Unter Berücksichtigung dieser Kriterien eignen sich insbesondere PDR-Verfahren bzw. sog. *Step-and-Heading-Systeme* (SHS) als technische Lösungen und sind daher Kernbestandteile vieler infrastrukturfreier Indoor-Positionierungssysteme. In ihnen werden Schrittvektoren, deren Länge und Richtung es möglichst akkurat zu schätzen gilt, ausgehend von einer bekannten Startposition aufaddiert und somit neue Positionsschätzungen generiert.

Schritterkennung

Für die Entscheidung, ob und wann ein Schritt getätigt wurde, kann in modernen Geräten der entsprechende, bereits in Abschnitt 2.3.2 beschriebene Sensor verwendet werden. Weiterhin sind Verfahren üblich, die Schritte anhand der Rohdaten des Accelerometers bestimmen, beispielsweise durch Anwendung von Schwellenwerten, Erkennung von Spitzenwerten oder Nulldurchgängen, Template-Matching oder Spektralanalyse (Harle, 2013, S. 1284). Letztgenannte Verfahren eignen sich vor allem für längere Gehphasen mit gleichbleibender Geschwindigkeit.

Die Auswirkung unterschiedlicher Gerätepositionen auf die Schritterkennung wurde von Brajdic und Harle (2013) untersucht. Hierfür gingen 27 Testpersonen, mit einem oder mehreren Smartphones in der Hand oder verschiedenen Taschen

ausgestattet, in variabler Geschwindigkeit entlang einer geraden Teststrecke. In den meisten Gerätepositionen tendieren die evaluierten Schritterkennungs-Algorithmen zur Nicht-Erkennung von Schritten; lediglich bei einem in der Handtasche getragenen Gerät führen – so die Erklärung der Autoren – zusätzliche Schwingungen zu einer Überschätzung der Schrittzahl. Die deutlichsten Abweichungen treten beim Tragen des Geräts in der hinteren Hosentasche auf.

Schrittlängenschätzung

Die menschliche Schrittlänge variiert unter anderem abhängig von Körpergröße und -gewicht, Geschlecht sowie Gehgeschwindigkeit und Umgebung (Winter et al., 2019, S. 61:12). Damit eine PDR-Positionsschätzung auf längere Zeit möglichst korrekt bleibt, muss die Schrittlängen-Variation berücksichtigt werden. Für eine Übersicht über zahlreiche für diesen Zweck in der Literatur entwickelte Verfahren sei auf Diez, Bahillo, Otegui und Otim (2018) verwiesen. Grob lässt sich zwischen biomechanischen Ansätzen, welche die menschliche Physiologie modellieren, sowie parametrischen, meist auf Regression basierenden Verfahren unterscheiden.

Erstere arbeiten durch Analyse von Daten fest am Körper fixierter Sensoren, welche beispielsweise die Bewegung der Hüfte oder den Winkel des Oberschenkels messen, um zu einer akkuraten Schätzung der Schrittlänge zu gelangen (siehe z. B. Munoz Diaz, 2015, S. 9162 ff.). Da im vorliegenden Fall keine derartigen Annahmen getroffen bzw. Vorgaben gemacht und keine zusätzlichen Sensoren verwendet werden, sind diese Verfahren hier nicht anwendbar.

Das häufig in der Literatur verwendete parametrische Verfahren von Weinberg (2002) schätzt die Schrittlänge anhand der Formel

$$l = K \cdot \sqrt[4]{a_{max} - a_{min}} \quad (3.1)$$

aus den maximal (a_{max}) und minimal (a_{min}) gemessenen Beschleunigungswerten in z-Richtung während des Schritts. K ist ein einstellbarer Faktor, den beispielsweise Y. Zhou, Zheng, Xiong und Chen (2017) auf 0,46 setzen. Shin, Park, Kim, Hong und Lee (2007) präsentieren eine Lösung, bei der die Schrittlänge als eine Linearkombination von Schrittfrequenz f und Varianz der Beschleunigung a berechnet wird (mit α , β und γ als datengetrieben gelernte Parameter):

$$l = \alpha \cdot f + \beta \cdot \text{var}(a) + \gamma \quad (3.2)$$

Eine von Tian, Salcic, Wang und Pan (2016) vorgeschlagene Alternative modelliert die Schrittlänge als proportional zur Quadratwurzel der Schrittfrequenz, mit Fak-

toren für geschlechtsspezifische (k) und körpergrößenbedingte (h) Unterschiede:

$$l = k \cdot h \cdot \sqrt{f_s} \quad (3.3)$$

Anhand dieser Beispiele wird deutlich, dass eine Schätzung der Schrittlänge ohne zusätzliche Informationen oder Annahmen in der Praxis nur begrenzt möglich ist.

Bestimmung der Schrittrichtung

Neben der Schrittlänge ist die Schrittrichtung die zweite wichtige Größe, die es dem PDR-System erlaubt, die Positionsschätzung aufrechtzuerhalten.

Wie bereits in Abschnitt 2.3.2 beschrieben, ist die vom Magnetkompass gemessene Richtungsangabe innerhalb von Gebäuden häufig durch Umwelteinflüsse gestört. Während dies unter Umständen als Teil eines hybriden Systems – siehe dazu den folgenden Abschnitt 3.1.2 – eine zur Positionsbestimmung dienliche Information ist, wird dadurch die PDR-Positionierung deutlich erschwert.

Zudem besteht auch hier die Problematik, dass die Gehrichtung nur indirekt bestimmt werden kann, da die Smartphone-Ausrichtung und -Position in Bezug auf den Körper nicht fixiert sind. Für die Lösung des Problems schlagen Combettes und Renaudin (2015) beispielsweise die Hauptkomponentenanalyse der 3D-Beschleunigungsdaten und die Modellierung von Längs- und Seitwärtsbewegungen vor. Alternative Verfahren setzen auf eine Fusion aus Gyroskop und Accelerometer, um Richtungsänderungen zu erkennen (Roy, Wang & Roy Choudhury, 2014).

Die in Smartphones verbauten MEMS-Sensoren sind nicht auf höchste Präzision ausgelegt und liefern daher auf längere Zeit keine stabilen Richtungsangaben. Durch Annahmen über die architektonische Struktur von Gebäuden – z. B. einer rechtwinkligen Anordnung von geraden Korridoren – können die Auswirkungen dieser Drift heuristisch reduziert oder eliminiert werden (Borenstein & Ojeda, 2010; Jimenez, Seco, Zampella, Prieto & Guevara, 2011).

Map-Matching

Ist wie im vorliegenden Fall ein Modell der Umgebung vorhanden, kann die Richtungs- bzw. Positionskorrektur anhand der dort enthaltenen Informationen über begehbare Bereiche erfolgen, wobei je nach Art des Modells unterschiedliche Map-Matching-Verfahren zum Einsatz kommen. Für die Implementierung von Map-Matching-Verfahren bieten sich stochastische Filter (siehe Abschnitt 3.1.2) an.

In detaillierten architektonischen Modellen, in denen die genauen Positionen

und Ausmaße von Wänden verzeichnet sind, ist eine Kollisionsabfrage möglich. Schneidet ein Schrittvektor eine Wand, so ist die dazugehörige Positionsschätzung nicht valide (Beauregard, Widyawan & Klepal, 2008; Perttula et al., 2014).

Aus Gebäudeplänen generierte Graph- oder Gitterstrukturen diskretisieren den Raum, wobei sich die verwendete Granularität auf die erreichbare Positionierungsgenauigkeit auswirkt. Je nach Anwendungsfall überwiegen jedoch die Vorteile, u. a. in Form von höherer Effizienz und Stabilität. Dies zeigen bereits Lin Liao, Fox, Hightower, Kautz und Schulz (2003), indem sie aus einem Gebäudeplan einen sog. Voronoi-Graph generieren, welcher nach geringer manueller Nachbearbeitung strukturell nicht unähnlich zum URwalking-Navigationsgraph ist. Die Autoren machen weiterhin deutlich, dass sich mit diesem Ansatz übliche menschliche Bewegungsmuster in Gebäuden gut abbilden lassen und sich die Positionierungs-Performanz verbessert.

Ein weiteres Beispiel für ein Graphmodell findet sich bei Hilsenbeck, Bobkov, Schroth, Huitl und Steinbach (2014) mit einer Kantenlänge, die in etwa der Schrittlänge entspricht, während F. Ebner, Fetzer, Deinzer, Köping und Grzegorzek (2015) auf ein festes Gitter von 0,2 m Abstand setzen. In ihrem Vergleich von Graph- und Meshstruktur bescheinigen Fetzer, Ebner, Bullmann, Deinzer und Grzegorzek (2018) dem Mesh eine höhere Langzeitstabilität. Eine Variante des graphbasierten Map-Matching präsentieren Herrera et al. (2014), indem sie OSM-Indoor-Kartenmaterial annotieren, um ausgehend von den Knoten und Kanten des Graphs mithilfe von Gaußschen Verteilungen die Aufenthaltswahrscheinlichkeiten von FußgängerInnen zu modellieren.

Stockwerkswechsel

Für eine Positionsverfolgung nicht nur innerhalb einer Ebene, sondern auch über mehrere Stockwerke hinweg, ist es erforderlich, Stockwerkswechsel zu erkennen. Hierfür bietet sich, wie bereits in Abschnitt 2.3.2 beschrieben, das Barometer an, mithilfe dessen anhand von Luftdruckveränderungen Rückschlüsse auf die Höhe bzw. Höhenänderungen möglich sind.

Da Luftdruckwerte wetterbedingt schwanken, werden in der Praxis probabilistische Modelle angewendet (Banerjee, Agarwal & Sharma, 2015) oder zusätzlich Inertialsensoren herangezogen (Xu, Wei, Zhu & Yang, 2017). Um die Problematik von zeitlich variablen Luftdruckwerten zu umgehen, setzen H. Ye, Gu, Tao und Lu (2016) auf ein Fingerprinting-Verfahren, in dem aktuelle Referenzwerte auf einem zentralen Server vorgehalten werden.

Um die Art des Stockwerkswechsels zu bestimmen, bieten sich HAR-Verfahren an, welche später in Kapitel 5 gesondert behandelt werden.

Beispiele für PDR-Systeme

Abschließend sollen an dieser Stelle einige Positionierungssysteme vorgestellt werden, die auf den soeben beschriebenen Bausteinen basieren. Teilweise lässt sich dabei der Anspruch kompletter Unabhängigkeit von externer Infrastruktur nicht aufrechterhalten.

Carboni, Manchinu, Marotto, Piras und Serra (2015) operieren unter ähnlichen Randbedingungen wie die vorliegende Arbeit. Auch sie erkennen an, dass durch einen geschickten Einsatz von Inertialsensorik und digitalem Kartenmaterial eine ausreichend hohe Genauigkeit für den Anwendungszweck der Fußgängernavigation erreicht werden kann. Für die Festlegung der Initialposition sowie periodische Checkpoints kommen zusätzlich QR-Codes zum Einsatz.

Hsu et al. (2018) stellen ein PDR-System vor, welches lediglich auf Accelerometer und Gyroskop angewiesen ist. Die Initialisierung und Kalibrierung erfolgt mithilfe von »calibration marks« genannten Pfeilen am Boden. Das Verfahren erfordert das Tragen des Geräts in Gehrichtung vor dem Körper, kämpft aber dennoch mit Drift-Problemen, da kein Map-Matching zum Einsatz kommt.

Ein Beispiel für heuristische Richtungskorrektur in einem PDR-System findet sich bei Giorgi, Frigo und Narduzzi (2017). Nach einer vorherigen Kalibrierung des Magnetfeldsensors darf sich das Gerät auch in der Tasche befinden, die Disambiguierung der Gehrichtung erfolgt anhand der vier dominanten Richtungen des Gebäudes im 90°-Abstand. Da sich jedoch auch die zur Evaluation verwendeten Teststrecken strikt an diese Einschränkung halten und nur aus 90°-Abbiegungen bestehen, erscheint der Nutzen unter realen Bedingungen fragwürdig.

Ju, Park und Park (2018) schlagen eine Weiterentwicklung der heuristischen Drift-Eliminierung vor, bei der die gemäß der Gebäuderotation dominanten Richtungen selektiv für einzelne Abschnitte während der Navigation zur Richtungskorrektur herangezogen werden. Mit diesem »multiple virtual tracking« genannten Verfahren reduziert sich der am Ende der Teststrecken gemessene Positionsfehler deutlich.

3.1.2 Hybride PDR-Systeme

Für eine robuste und zuverlässige Positionsbestimmung haben sich hybride Systeme etabliert, die weitere Sensoren oder Informationen integrieren, um die Schätzungen der PDR-Komponente zu stützen und somit Langzeitstabilität zu erreichen. Dies führt in den meisten Fällen zu Abhängigkeiten von externer Infrastruktur.

Zusätzliche Sensorik

Am häufigsten wird in hybriden Systemen auf Funk-Technologien wie WLAN oder Bluetooth als zusätzliche Komponente zurückgegriffen. Schon kurz nach dem Aufkommen von WLAN-Geräten und -Infrastruktur wurde das Potenzial für darauf basierende Ortungsverfahren erkannt (Bahl & Padmanabhan, 2000). Bei Verwendung handelsüblicher Smartphones als Empfangsgeräte werden zumeist Fingerprinting-Verfahren eingesetzt, bei welchen zunächst eine Karte (die sog. *Radiomap*) der im Gebäude empfangenen WLAN-Access-Points mitsamt ihren jeweiligen Signalstärken (*Received Signal Strength*, RSS) aufgebaut wird. Im Live-Betrieb erfolgt dann ein Abgleich der aktuellen RSS-Daten mit der Radiomap, wofür in der Literatur verschiedene Verfahren und Ähnlichkeitsmaße verwendet werden, darunter Nearest-Neighbor-Verfahren (Bahl & Padmanabhan, 2000), Clustering (Youssef & Agrawala, 2005) und genetische Algorithmen (Chintalapudi, Padmanabha Iyer & Padmanabhan, 2010). In der jüngeren Vergangenheit kamen vor allem ML-Methoden zum Einsatz, u. a. Random Forests (Y. Wang, Xiu, Zhang & Yang, 2018), Neuronale Netze (Song et al., 2019) und Autoencoder (Abbas, Elhamshary, Rizk, Torki & Youssef, 2019). Größter Nachteil der Fingerprinting-Verfahren ist die zeitintensive Erstellung der Radiomap sowie die Notwendigkeit, sie bei Änderungen an der WLAN-Infrastruktur zu aktualisieren. Anstelle des RSS-Verfahrens können in neuerer WLAN-Hardware mithilfe sog. *Channel State Information* (CSI) deutlich genauere Positionierungsergebnisse erzielt werden (Ma, Zhou & Wang, 2019).

Eine weitere häufig zur Positionierung verwendete Funktechnologie ist Bluetooth bzw. die weiterentwickelte Variante *Bluetooth Low Energy* (BLE). Popularisiert u. a. durch Apples iBeacons, lässt sich hier anhand der Signalstärke (RSS) die Distanz zwischen Sender und Empfänger genauer als bei Benutzung von WLAN messen und durch Trilateration zu einer Positionsschätzung verarbeiten (Xin-Yu Lin et al., 2015). Die flächendeckende Ausstattung eines Gebäudes mit Beacons, um eine ausreichende Genauigkeit zu gewährleisten, ist jedoch mit hohen initialen Kosten verbunden, da im Gegensatz zu WLAN in der Regel nicht auf existierende Infrastruktur zurückgegriffen werden kann.

In den letzten Jahren fokussiert sich die Forschung zu funkgestützter Indoor-Ortung zunehmend auf Ultra-Wideband-Technologie, nicht zuletzt wiederum aufgrund der Unterstützung in Apples iPhones und AirTags. UWB ermöglicht prinzipbedingt nochmals höhere Genauigkeit als BLE und WLAN, jedoch bei geringerer Reichweite (Zafari, Gkelias & Leung, 2019, S. 9 f.), wodurch sich die Anforderungen hinsichtlich der aufzubauenden Infrastruktur verschärfen. Auch hier findet die Positionsbestimmung anhand RSS oder Trilateration statt (siehe z. B.

Garcia, Poudereux, Hernandez, Urena & Gualda, 2015; Gigl, Janssen, Dizdarevic, Witrissal & Irahauten, 2007).

Neben Funktechnologien werden auch optische Verfahren erforscht, die beispielsweise anhand von Aufnahmen der Smartphone-Kamera in einer Bilddatenbank oder einem detaillierten Modell der Umgebung mithilfe von Image-Retrieval-Methoden die Position bestimmen (siehe z. B. van Opdenbosch et al., 2014). Aufgrund der großen Datenmengen, die nicht auf Mobilgeräten vorgehalten werden können, ist hier in der Regel eine Client-Server-Lösung nötig. Andere kameragestützte Verfahren arbeiten infrastrukturabhängig mithilfe von speziell manipulierten LED-Leuchtmitteln, welche hochfrequente, für das menschliche Auge nicht wahrnehmbare Muster ausstrahlen, um Informationen über den aktuellen Standort zu übertragen (Kuo, Pannuto, Hsiao & Dutta, 2014; Luo, Fan & Li, 2017). Auch mit akustischen Verfahren wie etwa *RoomSense* von Rossi, Seiter, Amft, Buchmeier und Tröster (2013) lassen sich durch Messung und Abgleich der Reflexions-Charakteristik Räume oder Standorte wiedererkennen.

Zuletzt sind in diesem Zusammenhang noch magnetfeldgestützte Ansätze zu nennen, welche lokale Magnetfeld-Anomalien, die beispielsweise durch Stahlbeton, Metalltüren oder Aufzugsschächte verursacht werden, für die Umsetzung von Positionierungs-Verfahren nutzen (Subbu et al., 2013).

Voraussetzung für den Einsatz der in diesem Abschnitt vorgestellten Verfahren ist einerseits die zeitintensive – und teils nur mit Spezial-Equipment durchführbare – Erstellung von Umgebungsmodellen, sei es in Form von WLAN-Radiomaps oder laservermessenen 3D-Modellen. Hinzu kommt in vielen Fällen die Notwendigkeit der mit hohen Kosten verbundenen Installation zusätzlicher Hardware, beispielsweise als Bluetooth-Beacons oder UWB-Sender. Da diese Voraussetzungen im vorliegenden Fall nicht erfüllt sind, werden derartige infrastrukturabhängige Ansätze im weiteren Verlauf dieser Arbeit nicht verfolgt, sondern dienen lediglich als Vergleichsobjekt und Illustration des Stands der Technik.

Für den Regensburger Campus liegt zwar eine Karte der verbauten WLAN-Accesspoints vor, nicht jedoch eine flächendeckende Radiomap. Während u. a. im Rahmen des DIVIS-Projekts an der Positionsbestimmung mit WLAN und Bluetooth geforscht wurde, ist im URwalking-System WLAN lediglich für die grobe Initialpositionierung im Einsatz. Zur Stabilisierung der Positionsbestimmung wird in der vorliegenden Arbeit stattdessen ein anderer Ansatz in Form von HAR verfolgt (siehe Kapitel 5).

Stochastische Filterverfahren

Alle bisher vorgestellten Verfahren nehmen die Positionsschätzung anhand unsicherer Information vor. PDR-Systeme können die Schrittlänge und -richtung nur zu einem gewissen Grad genau bestimmen, wodurch sich mit jedem getätigten Schritt der potenzielle Fehler akkumuliert. Um die Positionierung zu stabilisieren, sind zusätzliche Informationen z. B. in Form eines Umgebungsmodells nötig, mit welchem die PDR-Position mithilfe von Map-Matching abgeglichen wird. Auch infrastrukturabhängige Verfahren messen die Position im Rahmen der physikalischen Grenzen nur mit einer begrenzten Genauigkeit und sind oft auf mehrere aufeinanderfolgende Messungen angewiesen, um die Unsicherheit der Schätzung zu reduzieren.

Um die unsicheren Informationen mehrerer Quellen zu einer Positionsschätzung zu verarbeiten, werden in der Literatur verschiedene Arten von stochastischen Filtern verwendet. Als De-Facto-Standard für diesen Zweck hat sich das Partikelfilter – zuerst von Gordon, Salmond und Smith (1993) unter der Bezeichnung *Bootstrap-Filter* beschrieben – etabliert, welches auch in dieser Arbeit zum Einsatz kommt.

Partikelfilter eignen sich als Filterverfahren für nichtlineare Systeme und repräsentieren die A-Posteriori-Verteilung durch Stichproben, die namensgebenden Partikel. Auf diese Weise können sie sich beliebigen nicht-parametrischen Funktionen annähern, was einen wichtigen Vorteil gegenüber anderen Verfahren wie etwa dem Kalman-Filter darstellt, welches lediglich Gaußsche Verteilungen approximieren kann (Thrun, Burgard & Fox, 2006, S. 96 f.). In ihrer Grundform sind Partikelfilter einfach zu implementieren und flexibel anwendbar (Doucet, Freitas & Gordon, 2001, S. 4), beispielsweise im Bereich Computer Vision als *Condensation-Algorithmus* (Isard & Blake, 1998) oder auch unter der Bezeichnung *Sequential Monte Carlo* (Liu & Chen, 1998). Technische Details und die Implementierung im URwalking-System werden später in Abschnitt 3.3.2 beschrieben.

Beispiele für hybride Systeme

Mit Literaturbeispielen für hybride Indoor-Positionierungssysteme ließen sich dutzende Seiten füllen, weswegen im Folgenden hauptsächlich solche Systeme vorgestellt werden, die mit den in dieser Arbeit herrschenden Randbedingungen kompatibel sind, d. h. smartphonebasiert und mit PDR als zentraler Technologie. Das Hauptaugenmerk liegt zudem auf Partikelfilter-Verfahren, welche die Umgebung in Form einer Graphstruktur repräsentieren.

Ein Beispiel für die Verwendung des umgebenden Magnetfelds zur Stabilisierung von PDR findet sich bei Xie, Gu, Tao, Ye und Lu (2016). Nach der Ver-

messung des Magnetfelds innerhalb eines Universitätsgebäudes mit einer Auflösung von 0,1 m erreicht das Partikelfilter-Verfahren eine Positionierungsgenauigkeit von 1–2 m, wobei keine Vorgaben hinsichtlich der Geräteausrichtung gemacht werden.

Von Menke und Zakhor (2015) werden PDR, WLAN, sowie bild- und magnetfeldbasierte Positionierung in einem multimodalen System fusioniert. Nach initialer Ortung durch WLAN und Bildmaterial erfolgt das Tracking mithilfe eines Partikelfilters, welches eine durchschnittliche Genauigkeit von 2,6 m erreicht.

Ein frühes Beispiel für die Verwendung eines Graphs zur Stabilisierung von PDR-basierter Positionierung findet sich bei Kim, Kim, Yoon und Kim (2012). In ihrem hybriden System, welches neben Schritterkennung auch Magnetfeldanomalien zur Positionsbestimmung heranzieht, wird die Bewegungsrichtung anhand der Kantenrichtung im Graph bestimmt. Das modellierte Testareal besteht dabei aus lediglich vier rechteckig angeordneten Kanten, die Übergänge an den Eck-Knoten werden mithilfe des Gyroskops detektiert. In der Evaluation wird auf einer Teststrecke von 156 m Länge ein mittlerer Fehler von knapp über 1 m erreicht.

Nurminen, Ristimäki, Ali-Loytty und Piche (2013) stellen ein hybrides Partikelfilter-Verfahren vor, das für die PDR-Korrektur auf WLAN und Gebäudepläne setzt. In einer Weiterentwicklung werden die Gebäudepläne durch eine Graphstruktur ersetzt, wodurch sich in der Evaluation auf vier verschiedenen Teststrecken der Positionsfehler deutlich verringert (Nurminen, Koivisto, Ali-Loytty & Piche, 2014).

Hilsenbeck et al. (2014) verfolgen in ihrem System die Idee, für die Positionierung innerhalb von Gebäuden unnötige Freiheitsgrade zu reduzieren, indem sie aus Gebäudeplänen automatisiert Graphen generieren. Die auch im URwalking-Kontext bereits angesprochene Problematik offener Areale wird durch Auffüllen mit einer engmaschigen Graphstruktur gelöst. Das darauf operierende Positionierungssystem, in welchem PDR und WLAN mittels Partikelfilter fusioniert werden, benötigt weniger Partikel als vergleichbare Verfahren und arbeitet somit ressourcenschonender.

Auch F. Ebner et al. (2015) erzeugen aus Gebäudeplänen engmaschige Graphstrukturen, welche es erlauben, Fußgängerbewegungen mithilfe von Random Walks zu modellieren. Zur Stabilisierung der Schätzung dienen WLAN und Bluetooth, Stockwerkswechsel werden vom Barometer erkannt. Die Evaluation auf einer 220 m langen Route ergibt im besten Fall einen mittleren Positionsfehler von 3,75 m, wobei die Berechnungen nicht in Echtzeit stattfinden.

Fetzer et al. (2018) präsentieren eine Positionierungslösung u. a. auf PDR- und WLAN-Basis, die sich in kurzer Zeit (weniger als 120 min) in einer neuen Umgebung ausrollen lässt, und demonstrieren dies anhand eines historischen Gebäudes.

Der Partikelfilter operiert in diesem Fall auf einer Mesh-Struktur, die als Weiterentwicklung der Graphstruktur aus F. Ebner et al. (2015) hervorgeht. Der mittlere Positionsfehler des Systems liegt – je nach konkreter Konfiguration und Teststrecke – zwischen 3,2 und 13,4 m.

Das Konzept des *Semantic Augmented Route Network Graph* genannten Umgebungsmodells von Y. Zhou et al. (2017) ist vergleichbar mit dem in der vorliegenden Arbeit verwendeten Navigationsgraph. Auch dort sind geometrische, topologische und semantische Informationen in einem hybriden Graph vereint, dementsprechend arbeitet die PDR-Implementierung auf Basis eines Partikelfilters unter ähnlichen Voraussetzungen. Insbesondere ist auch hier an manchen Orten eine komplett freie Bewegung erlaubt – analog zu den Indoor-Freiflächen in der vorliegenden Arbeit. Durch Korrektur der PDR-Position mithilfe des Graphs wird ohne zusätzliche Sensorik die Positionierungsgenauigkeit um 36,4 % verbessert.

Ein weiteres Beispiel für partikelfiltergestütztes Positionstracking in einem hybriden Graph-/Polygon-Modell findet sich bei Ferris, Haehnel und Fox (2006). Während die verwendete Sensorik – WLAN und GSM-Informationen – nicht auf die vorliegende Problemstellung übertragbar ist, finden sich dort jedoch Details zum Umgang mit derartigen Umgebungsmodellen.

3.1.3 Evaluation von Indoor-Positionierungssystemen

Wie die Vorstellung einiger hybrider Systeme im vorigen Abschnitt gezeigt hat, wird die Positionierungsgenauigkeit in der Literatur oft in Form des mittleren Positionsfehlers, d. h. der Distanz zwischen geschätzter Position und tatsächlichem Standort (*Ground Truth*), angegeben. Im Folgenden werden weitere gängige Metriken beschrieben, die in der Vergangenheit bei der Evaluation von smartphonebasierten Positionierungssystemen zum Einsatz kamen.

Zuvor jedoch erfolgt ein Überblick über die in der Literatur übliche Evaluationsmethodik, mit Fokus auf die Realitätsnähe und Übertragbarkeit auf einen Navigationskontext.

Evaluationsmethodik

Für die Evaluation von Lokalisierungs- und Trackingssystemen existiert seit einigen Jahren der ISO/IEC-18305-Standard (International Organization for Standardization, 2016). Weiterhin bieten die jährlich stattfindenden Indoor-Positionierungswettbewerbe im Rahmen der IPIN-Konferenz einen Einblick in jeweils aktuelle Evaluationsmethodik.

Wie Bousdar Ahmed, Diez, Diaz und Garcia Dominguez (2020, S. 4) darlegen, wird der Vergleich unterschiedlicher Positionierungslösungen meist durch sequen-

tielles Testen unter (im Idealfall) gleichen Bedingungen oder durch Aufzeichnung eines Datensatzes und anschließenden Offline-Berechnungen gelöst. Seltener werden mehrere Systeme gleichzeitig von einer Person evaluiert. Offline-Studien, bei denen vor Ort die für spätere Experimente benötigten Sensordaten erhoben werden, haben einige Vorteile gegenüber In-Situ-Studien mit Echtzeit-Positionierung und werden auch in der vorliegenden Arbeit bevorzugt. Einmal aufgezeichnete Sensordaten können hierbei in mehreren Experimenten wiederverwendet werden, etwa um die Systemkonfiguration oder -parametrisierung im Nachhinein zu verändern bzw. zu optimieren. Da immer die gleichen Input-Daten vorhanden sind, ist – eine deterministische Implementierung des Positionierungs-Algorithmus vorausgesetzt – die Vergleichbarkeit der Ergebnisse gewährleistet.

Der ISO-Standard definiert 14 verschiedene Evaluationsszenarien, u. a. mit unterschiedlichen Bewegungsarten, darunter auch Laufen, Seitwärts- und Rückwärtsgehen. Bousdar Ahmed et al. (2020, S. 6) identifizieren am häufigsten Szenario 1, d. h. das Gehen entlang eines vordefinierten Pfads, bzw. Szenario 2, mit zusätzlichem Pausieren an für die Erhebung der Ground Truth verwendeten Referenzpunkten. Letzteres Szenario entspricht dem Vorgehen in älteren IPIN-Wettbewerben (siehe z. B. Torres-Sospedra et al., 2017), während in neueren Ausgaben durch den Verzicht auf künstliche Pausen höhere Realitätsnähe erreicht wird (Potorti et al., 2020, 2021). Weiterhin wurde versucht, durch zusätzliche Schwierigkeiten wie etwa durch das Simulieren eines Telefongesprächs, während dessen das Gerät ans Ohr gehalten wurde, eine realistischere Datengrundlage zu schaffen (Torres-Sospedra et al., 2018).

Das selbstständige Annotieren der Position an zuvor festgelegten Referenzpunkten durch die zu lokalisierende Person ist laut Bousdar Ahmed et al. (2020, S. 6) die häufigste Art der Evaluation. Andere, seltener genutzte Möglichkeiten zur Generierung einer Ground Truth sind die Positionsbestimmung durch bereits existierende, sekundäre Positionierungssysteme (Le Scornec, Ortiz & Renaudin, 2017) oder mithilfe von später vermessenen Video-Aufzeichnungen (vgl. z. B. Chen et al., 2015; Hoffmann, Schauer, Schonfeld & Kraus, 2017).

Distanzmetriken und Genauigkeitsmaße

Um die vom Positionierungssystem geschätzten Positionen mit der Ground Truth abzugleichen und daraus eine Kennzahl für die Genauigkeit des Gesamtsystems zu errechnen, wurden in der Vergangenheit verschiedene Metriken verwendet. Auch hier lohnt wieder ein Blick auf die Indoor-Positionierungs-Wettbewerbe der vergangenen Jahre:

In der ersten *Microsoft Indoor Localization Challenge* wurde das arithmetische

Mittel des Positionsfehlers (d. h. der euklidischen Distanz zwischen Schätzung und Ground Truth) zum Ranking der Teilnehmer herangezogen (Lymberopoulos et al., 2015). Schon damals merkten die Autoren an, dass durch die Reduzierung auf einen einzelnen Wert die Performance-Charakteristik von Positionierungssystemen nur schwer beurteilt werden kann.

Um Positionierungssysteme in größeren Umgebungen zu evaluieren, wird bei den im Rahmen der IPIN-Konferenz durchgeführten Wettbewerben bei fehlerhafter Stockwerks- oder Gebäudebestimmung zum 2D-Positionsfehler zusätzlich eine Strafdistanz – meist 15 m – addiert (Potorti, Barsocchi, Girolami, Torres-Sospedra & Montoliu, 2015; Potorti et al., 2020). Das finale Ranking der Teilnehmer basiert auf dem 75%-Quantil des resultierenden Werts.

Mendoza-Silva, Torres-Sospedra und Huerta (2017) schlagen eine Metrik vor, die anstelle der euklidischen Distanz den tatsächlich innerhalb des Gebäudes zurückzulegenden Pfad zwischen geschätzter Position und Ground Truth berücksichtigt. Auf diese Weise werden Situationen, in denen sich die Positionsschätzung hinter Wänden oder anderen Hindernissen befindet, realitätsgetreuer und – so die Autoren – näher am subjektiv vom Benutzer empfundenen Fehler repräsentiert. In einer weiteren Arbeit werden darauf aufbauend mehrere Verfahren, die u. a. auf Graph- und Meshmodellen basieren, anhand eines realen Wettbewerbs-Datensatzes verglichen (Mendoza-Silva et al., 2020). Als ein weiterer Vorteil dieses Ansatzes wird der Wegfall der recht arbiträr bestimmten Strafdistanz für falsche Stockwerke genannt.

Der oben genannte ISO-Standard sieht neben diversen 2D- und 3D-Varianten des euklidischen Fehlers für Fälle, in denen die so erreichbare Granularität ausreicht, eine zonenbasierte Metrik vor (Potorti, Crivello, Barsocchi & Palumbo, 2018, S. 3). Auch Pulkkinen und Verwijnen (2015) diskutieren die Problematik rein euklidischer Distanzmaße und weisen darauf hin, dass es für manche Anwendungen wichtiger ist, zuverlässig den Raum zu identifizieren, in dem sich ein Objekt befindet, anstatt dessen genaue Position innerhalb des Raums möglichst exakt zu bestimmen. Die später in Abschnitt 3.4.1.1 vorgestellte Evaluationsmethodik in der vorliegenden Arbeit folgt dieser Argumentation.

3.1.4 Indoor-Positionierung: Fazit

Zusammenfassend existiert trotz substanzieller Bemühungen keine Indoor-Positionierungslösung, die – analog zu satellitengestützten Systemen im Außenbereich – zuverlässig »out of the box« auf handelsüblichen Geräten funktioniert. Vielversprechende Forschungsprototypen, basierend auf hybriden Sensor-Architekturen, erfordern oft zusätzlichen Aufwand finanzieller oder technischer Natur, um ei-

ne flächendeckende Positionsbestimmung im Innenbereich zu ermöglichen. Forschung in diesem Gebiet ist daher nach wie vor höchst relevant – dies gilt heute ebenso wie zu Beginn dieses Projekts vor etwa fünf Jahren.

3.2 Positionstracking im Navigationsgraph

Nach der Übersicht über den Stand der Technik folgt nun die Beschreibung der technischen Umsetzung des Positionierungssystems im URwalking-Projekt für den Regensburger Universitätscampus. Für die Positionsverfolgung ist ein Verfahren nötig, um den aus Abschnitt 2.2 bekannten Navigationsgraph zu durchschreiten. Dies erfolgt durch Mapping der vom Smartphone erkannten Schritte auf die Graphstruktur. Im Folgenden werden zunächst Funktionsweise und Ablauf des PDR-Verfahrens näher erläutert, auf dessen Basis der anschließend vorgestellte Algorithmus für das Durchschreiten des Graphs arbeitet.

3.2.1 Pedestrian Dead Reckoning

Pedestrian Dead Reckoning oder Fußgänger-Totpunktnavigation bezeichnet die Positionsbestimmung von einer bekannten Position aus, die in regelmäßigen Abständen – im vorliegenden Fall bei jedem erkannten Schritt – durch Addition des Positionsvektors aktualisiert wird (Beauregard & Haas, 2006, S. 29). Für grundlegendes Positionstracking mittels PDR bestehen also zweierlei Voraussetzungen: Wissen um die Startposition sowie korrekt erkannte Schritte – inklusive ihrer Richtung und Länge.

Startposition

Im Rahmen dieser Arbeit wird ein Navigationskontext angenommen, wodurch als Startposition der Beginn der Route gegeben ist. Ob und wie oft diese Annahme in der Realität zutrifft, wird in Abschnitt 4.4.1 näher untersucht. Auf welche Weise der Startpunkt exakt festgelegt wurde, ist im Weiteren unerheblich. In der Navigations-App URwalking kann er – wie in Abschnitt 2.4.1 beschrieben – entweder aus der Liste der möglichen Start- bzw. Zielorte ausgewählt oder anhand der vom System bereitgestellten Position bestimmt werden.

Positionsverfolgung

Ausgehend von der bekannten Startposition wird die Positionsschätzung durch Addition von Schrittvektoren aufrechterhalten. Aus Abschnitt 2.3.2 sind bereits die für die Schritterkennung relevanten Sensoren bekannt: Der Schritterkennner

selbst liefert Informationen, ob und wann Schritte stattfinden; für die Richtungsbestimmung wird der GRV herangezogen. Die Schrittlänge wird zunächst als ein konstanter aus der Literatur abgeleiteter Erwartungswert von 0,65 m angenommen (Öberg, Karsznia & Öberg, 1993, S. 221 ff.). In Abschnitt 3.3.3 wird eine Möglichkeit zur Adaption an nutzerspezifische Schrittlängen vorgestellt.

Das PDR-Positionsupdate in einem zweidimensionalen euklidischen Raum lässt sich anhand eines einfachen Beispiels illustrieren. Ausgehend von einer Position (x_{t-1}/y_{t-1}) ergeben sich die neuen Koordinaten (x_t/y_t) nach folgender Formel (mit l_t als Länge und θ_t als Richtung des aktuellen Schrittes):

$$x_t = x_{t-1} + l_t \cdot \cos \theta_t \quad (3.4)$$

$$y_t = y_{t-1} + l_t \cdot \sin \theta_t \quad (3.5)$$

Im vorliegenden Fall sind jedoch – mit Ausnahme von gesondert behandelten flächigen Indoor-Bereichen – die Freiheitsgrade der Bewegung im Raum durch den Navigationsgraph eingeschränkt. Eine Lösung für die Positionsverfolgung mittels PDR unter diesen Voraussetzungen wird im nächsten Abschnitt vorgestellt.

Im obigen Modell wurde vereinfachend angenommen, dass Schrittrichtung und -länge exakt bestimmt werden können. In der Realität sind beide Messungen jedoch mit Unsicherheit behaftet, wodurch sich im Laufe der Zeit – wie in Dead-Reckoning-Verfahren üblich – der potenzielle Positionsfehler akkumuliert. Abschnitt 3.3 beschreibt eine Lösung für diese Problematik in Form des Partikelfilters.

3.2.2 Stabilisierung durch den Navigationsgraph

Der in Abschnitt 2.2 beschriebene Navigationsgraph schränkt die Bewegung auf die begehbaren Bereiche ein und dient somit zur Stabilisierung der Positionsschätzung. Dies erfordert es jedoch, das naive PDR-Modell zu erweitern, sodass anhand der erkannten Schritte das Durchschreiten des Graphs möglich wird. Zu diesem Zweck wird der aktuelle Zustand nicht als xy-Koordinaten gespeichert, sondern in Form des aktuellen Knotens und des Fortschritts zu den jeweiligen Nachbarknoten. Dies bildet zugleich die Grundlage für das spätere Zustandsmodell im Partikelfilter.

Dass sich mithilfe von Graphstrukturen die Stabilität und Genauigkeit der Positionsschätzung erhöhen lassen, haben beispielsweise Nurminen et al. (2014) und Y. Zhou et al. (2017) gezeigt. Die Verfahren von Hilsenbeck et al. (2014) und F. Ebner et al. (2015) sind am besten auf das vorliegende Problem übertragbar, arbeiten jedoch unter der Annahme einer feiner aufgelösten Graphstruktur. Wie

Abbildung 2.2 gezeigt hat, sind viele der Kanten im URwalking-Modell mehrere Meter lang, wodurch nicht bei jedem Schritt ein Übergang von einem Knoten zum Nachbarknoten erfolgt. Aus diesem Grund wird der Zustand durch eine diskrete Wahrscheinlichkeitsverteilung über alle angrenzenden Knoten ergänzt. Das Auswählen der korrekten Kante für eine gegebene Schrittrichtung (bzw. eine Abfolge von mehreren Schritten) kann als Klassifikationsproblem formuliert werden. Zum Einsatz kommt hierfür ein naiver Bayes-Klassifikator, der nach jedem Schritt anhand der gemessenen Schrittrichtung θ die wahrscheinlichste Kante E bestimmt. Nach dem Satz von Bayes gilt:

$$P(E|\theta) = \frac{P(E)p(\theta|E)}{p(\theta)} \quad (3.6)$$

Solange kein Schritt registriert wurde, ist die A-Priori-Wahrscheinlichkeit der Kanten $P(E)$ gleichverteilt und liegt im Falle von N möglichen Kanten bei $1/N$. Die Komponente $p(\theta)$, d. h. die Wahrscheinlichkeit, eine bestimmte Schrittrichtung zu messen, ist unabhängig von der jeweiligen Kante und hat keinen Einfluss auf das Verhältnis der A-Posteriori-Wahrscheinlichkeiten.

$p(\theta|E)$ entspricht der bedingten Wahrscheinlichkeit für das Messen von Schrittrichtung θ beim Gehen entlang Kante E . Bei jedem erkannten Schritt wird für alle Kanten die Winkeldifferenz zwischen Kanten- und Schrittwinkel bestimmt. Die Wahrscheinlichkeit wird, um die mit Rauschen behaftete Schrittrichtungsmessung zu berücksichtigen, anhand einer Normalverteilung bestimmt, die als Erwartungswert die Kantenrichtung hat. Somit kann die Berechnung der A-Posteriori-Wahrscheinlichkeit $P(E|\theta)$ erfolgen.

Die Kante mit der höchsten Wahrscheinlichkeit stellt die aktuelle Hypothese über den Zustand auf dem Graph dar. Überschreitet die zurückgelegte Strecke die Länge dieser Kante, so wird zum nächsten Knoten gewechselt, welcher nun als neuer Ausgangsknoten dient. Da der nächste Knoten in der Regel nicht exakt erreicht wird, wird die überschüssige Distanz bei der Initialisierung angerechnet. Unter der Annahme, dass keine abrupte Richtungsänderung erfolgt, werden die A-Priori-Wahrscheinlichkeiten nun anhand der vorherigen Kantenrichtung initialisiert. Das soeben beschriebene Verfahren ist in Algorithmus 3.1 zusammengefasst.

Die zwischengeschaltete Verteilung der Aufenthaltswahrscheinlichkeiten auf den einzelnen, momentan relevanten Kanten erlaubt es, Trajektorien abzubilden, bei denen der Ort der Abbiegung bzw. Richtungsänderung nicht exakt mit einem Knoten auf dem Graph zusammenfällt. Abbildung 3.1 illustriert dies anhand eines fiktiven Beispiels, bei dem erst kurz nach einem Entscheidungspunkt ein Richtungswechsel stattfindet, wodurch sich die wahrscheinlichste Kante und somit die Positionsschätzung ändert.

Algorithmus 3.1: Durchschreiten des Navigationsgraphs

- 1: Wähle Anfangsknoten n_0
- 2: Initialisiere zurückgelegte Distanz d
- 3: Erzeuge diskrete Verteilung e über angrenzende Knoten
- 4: **Für** jeden Schritt mit Länge l und Richtung θ :
- 5: Bestimme Differenz zwischen θ und angrenzenden Kanten
- 6: Aktualisiere e (Bayes)
- 7: Addiere Schrittlänge l zu zurückgelegter Distanz d
- 8: **Falls** $d >$ Länge der wahrscheinlichsten Kante in e :
- 9: Springe zum nächsten Knoten
- 10: **Gehe zu 2**, initialisiere anhand Längenüberschuss und Richtung

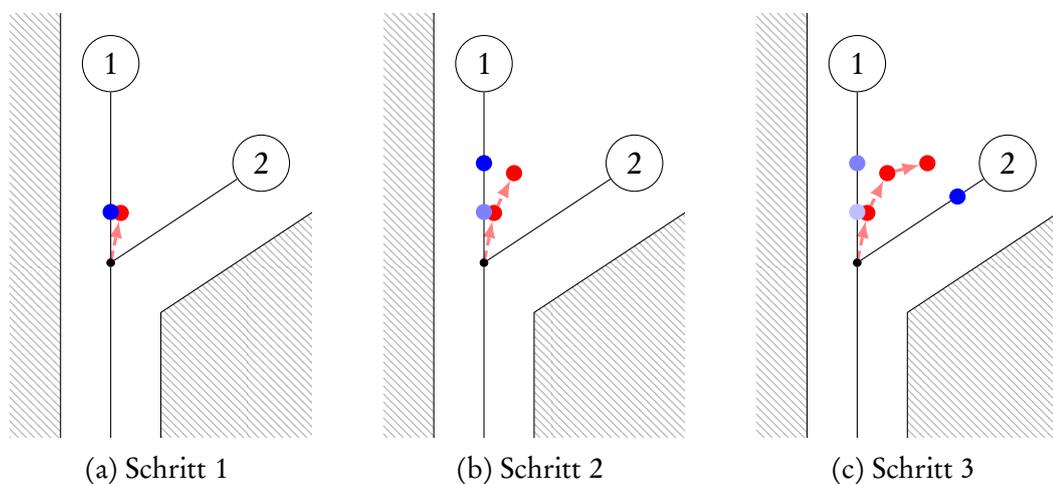


Abbildung 3.1: Durchschreiten des Navigationsgraphs (rot: Schrittvektoren; blau: geschätzte Position auf der wahrscheinlichsten Kante)

3.3 Nutzeradaption mithilfe des Partikelfilters

Das Partikelfilter bildet das Kernstück des entwickelten Verfahrens und ermöglicht Langzeitstabilität sowie die Anpassung an individuelle Benutzereigenschaften. Grundidee hinter dem Einsatz des stochastischen Filterverfahrens ist, den in Abschnitt 3.2.2 beschriebenen Algorithmus zu stabilisieren, indem nicht nur eine einzige Hypothese über den aktuellen Zustand aufrechterhalten wird, sondern eine Vielzahl davon.

In diesem Abschnitt werden zunächst die methodischen Grundlagen des Partikelfilters erläutert, gefolgt von der konkreten Implementierung im URwalking-System. Zuletzt werden speziell für den vorliegenden Anwendungszweck der Fußgänger-Navigation vorgenommene Anpassungen beschrieben.

3.3.1 Grundlagen

Partikelfilter ermöglichen die Schätzung eines nicht unmittelbar beobachtbaren Zustands auf Basis von unsicheren Beobachtungen. Im vorliegenden Fall stellt die Position des Benutzers die nicht direkt beobachtbare Größe dar. Messbar hingegen sind die Schritte des Benutzers, wenn auch – wie in vorigen Abschnitten etabliert – mit Unsicherheit hinsichtlich ihrer Richtung und Länge behaftet. Jedes Partikel repräsentiert eine Hypothese über den aktuellen Zustand, welcher wiederum in einem Zustandsmodell formalisiert ist. Zu festgelegten Zeitpunkten, im vorliegenden Fall bei jedem Schritt des Benutzers, finden Zustandsübergänge anhand der in einem Bewegungsmodell definierten Regeln statt.

Formal kann das Grundprinzip des Partikelfilters für die Positionsbestimmung durch die folgenden drei Schritte zusammengefasst werden (vgl. Hilsenbeck et al., 2014, S. 3):

$$x_t \sim p(x_t | x_{t-1}, u_t, \mathcal{G}) \quad (3.7)$$

$$w_t = w_{t-1} \cdot \mathcal{L}(z_t | x_t, \mathcal{G}) \quad (3.8)$$

$$\text{ziehe } x_t \propto w_t \quad (3.9)$$

Im ersten Schritt findet das *Update* des Partikelzustands x_t gemäß eines Bewegungsmodells statt, wobei der Navigationsgraph \mathcal{G} berücksichtigt wird. Anschließend wird das Gewicht w_t jedes Partikels anhand der Wahrscheinlichkeit des aktuellen Zustands bei einer Messung, d. h. der Übereinstimmung mit dem Graphmodell, aktualisiert (*Korrektur*-Schritt). Zuletzt werden beim *Resampling* proportional zur Verteilung der Gewichte neue Partikel aus der Verteilung gezogen (vgl. auch Harle, 2013, S. 1288 f.).

Partikelfiltergestützte Systeme lassen sich recht einfach um weitere Eingaben, z. B. in Form zusätzlicher Sensoren, erweitern und werden daher – wie bereits in Abschnitt 3.1.2 beschrieben – oft für die Implementierung hybrider Systeme beispielsweise aus PDR und WLAN verwendet. Im vorliegenden Fall wird das System später um HAR erweitert (siehe Abschnitt 5.4.1).

3.3.2 Implementierung

Aufbauend auf den Ausführungen über die grundsätzliche Funktionsweise des Partikelfilters folgt nun die Beschreibung der konkreten Implementierung des URwalking-Basis-Positionierungssystems. Dieses enthält als Kernstück den in Abschnitt 3.2.2 vorgestellten Algorithmus zum Traversieren des Navigationsgraphs. Die Partikelfilter-Implementierung orientiert sich an Thrun et al. (2006, S. 252)

und Hilsenbeck et al. (2014) und wird durch einige Anpassungen an die spezifischen Randbedingungen im URwalking-System erweitert. Im Folgenden werden zunächst die Komponenten der Partikelfilter-Lösung – namentlich das Zustandsmodell, die Initialisierung, das Update, die Gewichtung, das Resampling sowie die finale Positionsschätzung – im Einzelnen beschrieben.

Zustandsmodell

Das Zustandsmodell eines Partikels enthält alle Informationen für die Verortung auf dem Navigationsgraph. Hierfür sind, wie aus dem in Abschnitt 3.2.2 beschriebenen Algorithmus hervorgeht, folgende Merkmale nötig:

- Der aktuelle Knoten n
- Die von dort aus zurückgelegte Strecke d
- Die Verteilung der Kanten-Wahrscheinlichkeiten zu den angrenzenden Knoten e

Der resultierende Zustand zum Zeitpunkt t kann demnach durch den Vektor $\{n_t, d_t, e_t\}$ beschrieben werden. Hilsenbeck et al. (2014) setzen in ihrem ebenfalls graphbasierten Partikelfilter auf ein einfacheres Zustandsmodell, welches lediglich aus dem aktuellen Knoten und der Gehrichtung besteht. Das Verfahren setzt ein fein aufgelöstes Umgebungsmodell voraus, in dem die Länge aller Kanten in etwa der Schrittlänge entspricht. Die Kantenlängen im URwalking-Graph sind deutlich inhomogener und betragen in der Regel ein Vielfaches der Schrittlänge. Um den Fortschritt entlang solcher Kanten zu verfolgen, ist daher dieses komplexere Zustandsmodell nötig.

Initialisierung

Wie in Abschnitt 3.2.1 beschrieben, wird im Rahmen dieser Arbeit die Startposition in Form des Routenbeginns als gegeben betrachtet. Für die Initialisierung des Partikelfilters werden zunächst alle in einem kleinen Radius von 3 m um den Startpunkt liegenden Knoten ausgewählt. Auf diesen Knoten werden gleichverteilt Stichproben gezogen, welche die initiale Partikelmenge bilden. Durch Erweiterung der Partikelmenge auf nahegelegene Knoten wird dem Umstand Rechnung getragen, dass die Initialposition aufgrund der Diskretisierung des Raums durch das Graphmodell in der Regel nicht eindeutig mit der Benutzerposition übereinstimmt. Zudem wird dadurch sichergestellt, dass beispielsweise bei Auswahl einer Bürotür als Anfangspunkt der Route immer auch Partikel auf dem angrenzenden Korridor liegen, von dem aus eine Bewegung in beide Richtungen möglich ist. Abbildung 3.2 illustriert die Auswahl und Initialisierung der Startknoten.

Für jedes Partikel werden nun die angrenzenden Kanten bestimmt, anhand derer anschließend die diskrete Verteilung e_0 (vgl. Algorithmus 3.1) gleichverteilt initialisiert wird.

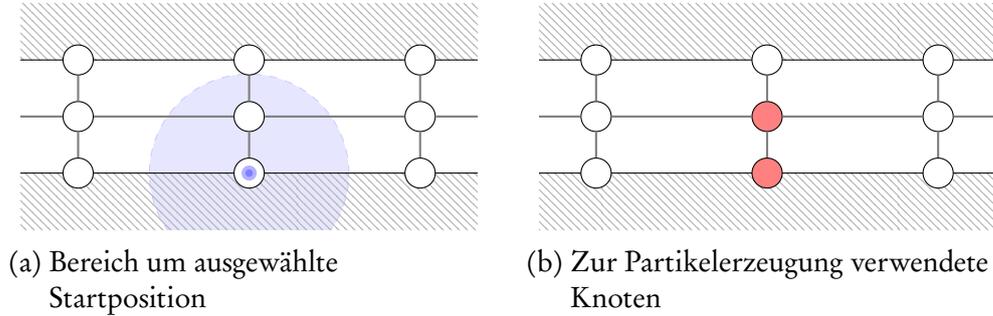


Abbildung 3.2: Initialisierung des Partikelfilters

Update

Nach jedem erkannten Schritt findet ein Update des Partikelfilters statt, bei dem der Zustand aller Partikel gemäß der folgenden Regel aktualisiert wird:

$$\{n_t, d_t, e_t\} \sim p(n_t, d_t, e_t | n_{t-1}, d_{t-1}, e_{t-1}, z_{\theta,t}, z_{l,t}, \mathcal{G}) \quad (3.10)$$

Die zum Zeitpunkt t vorgenommene Messung besteht aus dem Schrittwinkel $z_{\theta,t}$ und der Schrittlänge $z_{l,t}$. \mathcal{G} bezeichnet den Navigationsgraph.

Durch Anwendung der in Hilsenbeck et al. (2014, S. 5 f.) beschriebenen Prozedur kann die Update-Regel in ihre Einzelteile zerlegt werden. Die unsichere Schrittlängenschätzung ist durch

$$l_t \sim p(l_t | z_{l,t}) \sim \mathcal{N}(z_{l,t}, \sigma_l^2), \quad (3.11)$$

modelliert, mit σ_l^2 als der empirisch bestimmten Varianz. Dies führt zur neuen kumulativen Distanz

$$d_t \sim d_{t-1} + l_t. \quad (3.12)$$

Der erste Teil gilt analog auch für die Schrittrichtung:

$$\theta_t \sim p(\theta_t | z_{\theta,t}) \sim \mathcal{N}(z_{\theta,t}, \sigma_\theta^2) \quad (3.13)$$

Um die neue Kanten-Verteilung zu bestimmen, kommt folgende Regel zum Einsatz:

$$e_t^i \sim p(e_t^i | e_{t-1}^i, \theta_t, \mathcal{G}) \sim \mathcal{N}(\Delta\theta_t^i, \sigma_e^2) \cdot e_{t-1}^i \quad (3.14)$$

e_t^i bezeichnet hier die Wahrscheinlichkeit, sich auf der i -ten vom aktuellen Knoten wegführenden Kante zu befinden, $\Delta\theta_t^i$ die Winkel-Differenz zwischen der Schritt-richtung und der i -ten Kante.

Anschließend wird geprüft, ob durch den letzten Schritt ein neuer Knoten n_t erreicht wurde, d. h. ob die gegangene Distanz d_t die Länge l_E der Kante E mit der aktuell höchsten Wahrscheinlichkeit überschreitet (vgl. auch Y. Zhou et al., 2017, S. 15):

$$p(n_t|d_t, e_t) \sim \begin{cases} 1 & \text{falls } d_t \geq l_E \wedge E = \operatorname{argmax}_i(e_t) \\ 0 & \text{andernfalls} \end{cases} \quad (3.15)$$

Ist dies der Fall, wird der Knoten n_t auf den am Ende der Kante E befindlichen Knoten gesetzt. Die gegangene Distanz d_t wird zurückgesetzt, wobei etwaiger Überschuss von der vorherigen Kante angerechnet wird. Die Verteilung e_t wird unter Berücksichtigung der letzten Schritt-richtung neu initialisiert.

Gewichtung/Korrektur

Im Korrektur-Schritt werden die Stichproben gemäß ihrer Wahrscheinlichkeit in Bezug auf das Umgebungsmodell gewichtet. Partikel, deren Zustand nicht mit der zuvor erfolgten Messung vereinbar ist, erhalten auf diese Weise iterativ niedrigeren Einfluss. Im vorliegenden Fall erfolgt die Gewichtung anhand der Vereinbarkeit der gemessenen Schritte mit dem Navigationsgraph: Sind diese an der vermuteten Position gemäß des Umgebungsmodells nicht möglich, so muss diese Hypothese verworfen bzw. das Partikel niedriger gewichtet werden. Die formale Umsetzung sieht die Verteilung der Partikelgewichte w_t nach folgender Regel vor:

$$w_t = w_{t-1} \cdot p(z_{l,t}, z_{\theta,t}|n_t, \mathcal{G}) \sim w_{t-1} \cdot \max_i(e_t) \quad (3.16)$$

Die Gewichtsverteilung erfolgt demnach anhand der maximalen Wahrscheinlichkeit der Kanten aus e_t , wobei diese wiederum proportional zur Übereinstimmung von Schritt- und Kantenrichtung ist. Anschließend werden die Gewichte normalisiert (mit M als Partikelanzahl):

$$w_i = \frac{w_i}{\sum_{k=0}^M w_k} \quad (3.17)$$

Resampling

Als letzter Schritt erfolgt das Resampling, in dem gemäß der zuvor erfolgten Gewichtung neue Partikel mit Zurücklegen gezogen werden. Höher gewichtete Par-

tikel werden dabei unter Umständen mehrfach in die neue Partikelmenge mit aufgenommen, während bei Partikeln mit niedrigem Gewicht das Risiko steigt, nicht mehr enthalten zu sein. Nach dem *Survival-of-the-Fittest*-Prinzip nähert sich die Partikelverteilung somit der A-Posteriori-Verteilung an (Thrun et al., 2006, S. 99 f.).

Als konkretes Verfahren kommt das sog. *Low Variance Sampling* zum Einsatz, auch bekannt als *Stochastic Universal Sampling* (Baker, 1987). Das Verfahren garantiert eine zu den Partikelgewichten proportionale Verteilung der neuen Stichproben und weist eine geringe Komplexität von $O(M)$ auf (Thrun et al., 2006, S. 111). Um die Sampling-Varianz zu verringern, werden die Stichproben nicht unabhängig voneinander, sondern systematisch gezogen. Hierfür wird zunächst ein zufälliger Anfangswert u im Intervall $[0; \frac{1}{M}]$ bestimmt, von dem ausgehend durch $(M-1)$ -malige Addition von $\frac{1}{M}$ weitere äquidistante Werte im Intervall $[0; 1]$ erzeugt werden (vgl. Thrun et al., 2006, S. 109 ff.). Mithilfe dieser Werte werden nun Partikel selektiert, indem jedem Partikel proportional zu seinem Gewicht ein Intervall innerhalb von $[0; 1]$ zugewiesen wird. Abbildung 3.3 illustriert das beschriebene Verfahren anhand eines Beispiels mit zehn Partikeln.

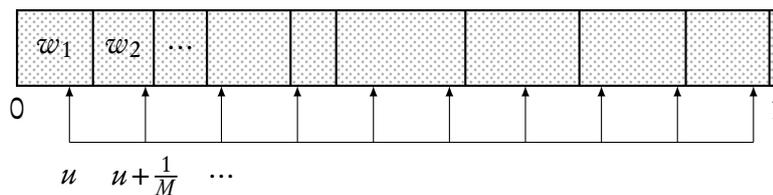


Abbildung 3.3: Low-Variance-Resampling-Verfahren (adaptiert von Thrun et al., 2006, S. 111)

Das Resampling der Partikelverteilung ist in der Regel nicht nach jedem einzelnen Schritt erforderlich, sondern kann nach Bedarf angestoßen werden. Ein verbreitetes Kriterium hierfür ist die nach folgender Formel berechnete *Effective Sample Size* (Kong, Liu & Wong, 1994):

$$N_{\text{eff}} = \frac{1}{\sum_{i=0}^M \omega_i^2} \quad (3.18)$$

Die Resampling-Prozedur wird immer dann durchlaufen, wenn N_{eff} unterhalb der erforderlichen Partikelanzahl bzw. eines definierten Schwellenwerts liegt. Welche Anzahl für diesen Anwendungszweck ausreichend bzw. angemessen ist, wird später in der Vorstudie in Abschnitt 3.4.3 evaluiert.

Positionsschätzung

Für die Integration in ein Navigationssystem ist es erforderlich, aus der Partikelverteilung eine konkrete Positionsangabe zu erzeugen, die dem Benutzer angezeigt oder die in der Navigationslogik weiterverarbeitet werden kann.

Enthält, wie bei F. Ebner et al. (2015) oder Fetzer et al. (2018), jedes Partikel unmittelbar seine Position, so kann der Erwartungswert dieser Verteilung als Positionsschätzung verwendet werden. Aufgrund des komplexeren Zustandsmodells muss im vorliegenden Fall zunächst ein Zwischenschritt erfolgen. Hierfür wird die Position jedes Partikels evaluiert, indem die vom Knoten n_t , dessen Koordinaten bekannt sind, zurückgelegte Strecke d_t auf den durch die wahrscheinlichste Kante beschriebenen Vektor angetragen wird. Anschließend wird der Mittelwert dieser Positionen unter Berücksichtigung der Partikelgewichte berechnet. Befinden sich Partikel in mehreren Etagen, so wird zunächst diejenige mit der höchsten Gewichtssumme bestimmt.

Aufgrund der Umrechnung in Koordinaten zur Bestimmung des Mittelwerts liegt die resultierende Position nicht notwendigerweise auf dem Graph. Als finale Positionsschätzung wird daher der nächstgelegene Punkt auf dem Graph bestimmt, um sicherzustellen, dass die Position an einem für den Benutzer zugänglichen Ort gelegen ist.

3.3.3 Anpassungen an fußgängerspezifisches Verhalten

Das bis hierher beschriebene Verfahren orientiert sich – mit Ausnahme des speziellen Umgebungsmodells – an den in der Literatur üblichen Vorgehensweisen, welche ihren Ursprung in der Robotik haben und demnach nicht als »nutzerzentriert« gelten können. Um dem Titel der Arbeit Genüge zu tun, werden im Folgenden einige Erweiterungen vorgestellt, durch die sich der Positionierungs-Algorithmus besser an seine NutzerInnen anpasst und die es ermöglichen, menschliches Navigationsverhalten abzubilden. Im Einzelnen sind dies die unbeschränkte Bewegung innerhalb offener Flächen im hybriden Graphmodell, die automatisierte Adaption der Schrittlänge während der Navigation sowie die Korrektur der Initialrichtung mithilfe von Routeninformationen. Während die grundlegenden Ideen und Verfahrensweisen bereits aus der Literatur bekannt sind, ist es ein Beitrag dieser Arbeit, sie im Kontext einer realistischen Navigationssituation und aufbauend auf dem existierenden Umgebungsmodell anzuwenden und zu evaluieren.

Bewegung in Freiflächen

Die erste Erweiterung des Verfahrens betrifft die durch Areanodes definierten Flächen, die aufgrund der dort herrschenden Bewegungsfreiheit nicht sinnvoll durch eine reine Graphstruktur abbildbar sind. Anstelle der starken Stabilisierungswirkung, die durch die Kanten des Graphs ausgeübt wird, können hier lediglich die äußeren Begrenzungen des Polygons zur Gewichtung der Partikel herangezogen werden, innerhalb davon ist komplett freie Bewegung erlaubt. Solange derartige Flächen in ihrer Ausdehnung begrenzt sind, bleibt die Akkumulation des durch unsichere Schrittrichtung und -länge verursachten Positionsfehlers in einem akzeptablen Rahmen.

Ferris et al. (2006) lösen Bewegung in einem hybriden Graph-/Polygon-Modell durch Verwendung separater Zustandsmodelle, je nachdem ob sich das Partikel auf einer Kante oder innerhalb einer Fläche befindet. Da diese Entscheidung im vorliegenden System auf Partikelebene stattfindet, ist die Bewegung in Freiflächen durch eine Erweiterung des Partikelzustands, ähnlich zum Verfahren von Y. Zhou et al. (2017), umgesetzt. Die diskrete Verteilung e_t enthält nunmehr nicht nur die an einen Knoten angrenzenden Kanten, sondern auch – sofern vorhanden – die mit dem Knoten verbundenen Flächen. Die mittels PDR geschätzte Roh-Position wird in Form von xy-Koordinaten gespeichert und bei jedem Schritt aktualisiert. Nach dem Positionsupdate findet zusätzlich ein Test statt, ob einer der Randknoten der Fläche (mit einer definierbaren Toleranz von standardmäßig 1 m) erreicht wurde. Ist dies der Fall, wird von dort aus das graphbasierte Verfahren wiederaufgenommen.

Bei der Gewichtung der Partikel gilt es, die oben erwähnte Problematik der fehlenden Stabilisierung durch den Graph zu lösen. Innerhalb von Flächen wird das Partikelgewicht nicht verändert, solange die Bewegung erlaubt ist, d. h. durch den letzten erfassten Schritt keine äußere Kante des Polygons überschritten wurde (mit Ausnahme der genannten Randknoten). Wird jedoch eine Begrenzung, meist eine Wand, durchschritten, so wird das Gewicht auf Null gesetzt, um das Partikel beim nächsten Resampling-Vorgang zu entfernen. In dieser Hinsicht entspricht der Algorithmus dem üblichen Vorgehen bei polygonbasierten Umgebungsmodellen (vgl. Beauregard et al., 2008, S. 143).

Schrittlängenschätzung

Die Problematik der Bestimmung der korrekten Schrittlänge wurde bereits in Abschnitt 3.1.1 thematisiert. Die in der Literatur verfolgten Ansätze sind aus unterschiedlichen Gründen nicht ideal für den vorliegenden Anwendungsfall geeignet: Im Falle eines in der Hand getragenen Geräts findet eine Entkopplung von der

Schrittbewegung statt, was zu einer Dämpfung oder – bei schwingender Gerätehaltung – Verstärkung der Beschleunigungskräfte führt. Verfahren wie etwa das von Weinberg (2002), die auf Basis der Beschleunigungs-Rohdaten die Schrittlänge zu schätzen versuchen, liefern daher unter Umständen stark schwankende oder unzuverlässige Schätzungen. Ebenso sind im Realbetrieb des Navigationssystems Körpergröße und Geschlecht der BenutzerInnen nicht bekannt, weswegen darauf aufbauende Lösungen nicht verwendet werden können.

Harle (2013, S. 1288) weist darauf hin, dass der durch ungenaue Schrittlängenschätzung verursachte Fehler gegenüber dem bei jedem Schritt möglichen Richtungsfehler in der Regel vernachlässigbar ist. Als einfachsten Ansatz zur Lösung des Problems nennt er die Annahme einer konstanten Schrittlänge, da sich FußgängerInnen üblicherweise mit relativ konstanter Geschwindigkeit bewegen, sofern sie nicht in Eile oder in Begleitung anderer Personen sind.

Im vorliegenden Fall wurde folgende Vorgehensweise gewählt, welche sich die Vorteile des Partikelfilter-Verfahrens zu eigen macht: Die Schrittlängenadaption findet anhand eines zusätzlichen Elements im Zustandsmodell, dem neu eingeführten Schrittlängen-Bias b_l , statt. Dieser stellt einen Skalierungsfaktor dar, mit dem bei jedem Schritt die ursprüngliche feste Länge multipliziert wird, wodurch jedes Partikel seine individuelle Schrittlänge bestimmt. Bei der Initialisierung werden die Bias-Werte gemäß dem von Rai, Chintalapudi, Padmanabhan und Sen (2012) beschriebenen Vorgehen im Intervall $[0, 75; 1, 25]$ gleichverteilt.

Geht der Benutzer nun beispielsweise geradeaus, so streut die Partikelverteilung nach einigen Iterationen entlang der Gehrichtung (siehe Abbildung 3.4). Bis zu diesem Zeitpunkt führt das Verfahren lediglich zu größerer Unsicherheit bei der Positionsschätzung. Der Vorteil kommt dann zum Tragen, wenn der Benutzer die Richtung ändert oder einen anderweitig identifizierbaren Ort passiert. In diesen Fällen werden die Partikel mit passendem Schrittlängenparameter aufgrund ihrer korrekten Positionsschätzung höher gewichtet (in Abbildung 3.4 durch die Partikelgröße dargestellt), wodurch das System implizit die Schrittlänge lernt. Ist diese Kalibrierung einmal erfolgt, kann die Schrittlänge eines Benutzers als relativ stabil angenommen werden (Harle, 2013, S. 1284).

Richtungskorrektur

Wie bereits beschrieben, wird für die Bestimmung der Schrittrichtung der GRV verwendet, welcher ohne das Magnetometer arbeitet und daher die Ausrichtung des Geräts um die globale z-Achse lediglich relativ zu einer arbiträren Startposition misst. Dies hat den Vorteil, dass Magnetfeldeinflüsse im Innenbereich keine störenden Auswirkungen auf die Richtungsschätzung haben, erfordert jedoch ei-

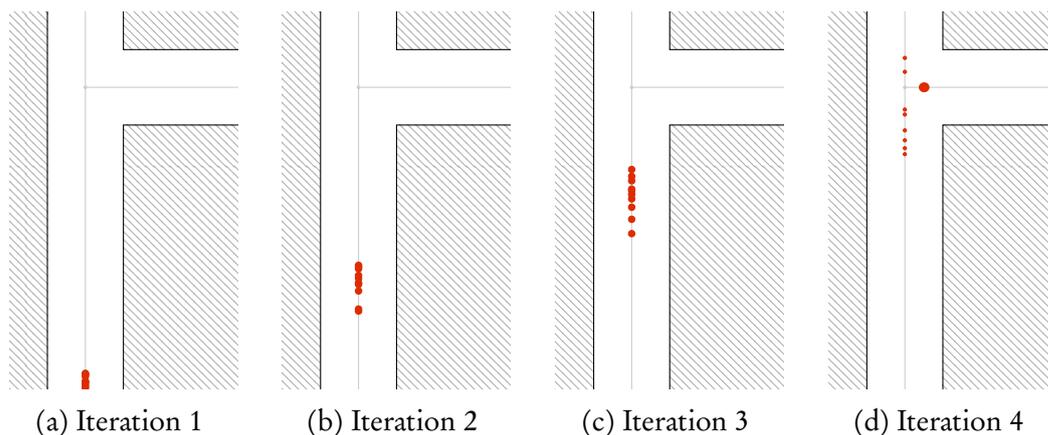


Abbildung 3.4: Beispiel für das Schrittlängen-Korrekturverfahren beim Abbiegen

ne zusätzliche Lösung für die Bestimmung der initialen Gehrichtung bzw. des Offsets, um den die GRV-Messwerte korrigiert werden müssen.

Auch hier erweist sich das Partikelfilter-Verfahren als vorteilhaft: Ähnlich wie zuvor bei der Schrittlänge wird nun auch für die Richtung ein Bias-Element b_θ in das Zustandsmodell aufgenommen, welches den Winkel repräsentiert, der zu den GRV-Richtungsangaben addiert werden muss, um die reale Gehrichtung zu erhalten. Während der Initialisierung wird dieser Wert systematisch von 0° bis 360° auf alle Partikel verteilt. Bei jedem erkannten Schritt erfolgt im Partikel-Update die Rotation des Schrittvektors um den Bias-Wert.

Im Laufe der ersten Resampling-Schritte kalibriert sich der Richtungs-Bias anhand der vom Navigationsgraph auferlegten Restriktionen hinsichtlich der begehbaren Wege von selbst, da inkorrekte Bias-Werte zu Schritt-Vektoren führen, die nicht mit dem Umgebungsmodell vereinbar sind. Startet die Navigation beispielsweise in einem Korridor mit zwei möglichen Gehrichtungen, so entsteht nach kurzer Zeit eine bimodale Verteilung der Bias-Werte. Für die weitere Disambiguierung kann – unter der Annahme, dass der Benutzer zu Beginn die Navigationsanweisungen befolgt – der Verlauf der geplanten Route herangezogen werden. Inwiefern dieses Verfahren unter realen Bedingungen erfolgreich ist, wird in Abschnitt 3.4.3 überprüft.

3.4 Studien zur Indoor-Positionierung

Zur Evaluation des beschriebenen Positionierungsverfahrens wurden mehrere Studien durchgeführt. In den folgenden Abschnitten werden zwei Vorstudien zur Verifizierung der grundlegenden Funktionalität und zur Erprobung des realitätsnahen Experiment-Setups vorgestellt. Eine große, abschließende Studie unter Ein-

beziehung von HAR findet sich später in Abschnitt 5.4.2.

3.4.1 Evaluationsmethodik

Vor der Beschreibung der beiden Studien wird in diesem Abschnitt die Methodik vorgestellt, die für alle weiteren Positionierungsexperimente in dieser Arbeit Anwendung findet. Im Einzelnen sind dies die an den Navigationskontext angepasste Evaluationsmetrik sowie Details zur Durchführung der Simulationsstudien.

3.4.1.1 Aufgabenorientierte Evaluation

Im URwalking-System sind – wie für Navigationssysteme im Allgemeinen typisch – Routen in mehrere Abschnitte unterteilt, welche mit einer Navigationsanweisung korrespondieren. Eine Besonderheit stellen die Landmarken dar, auf die in den Navigationsanweisungen verwiesen wird. Um als Benutzer auf diese Anweisung reagieren zu können, muss die referenzierte Landmarke vom aktuellen Standort aus sichtbar sein. Im Umkehrschluss bedeutet dies für das Positionierungssystem, dass die Bestimmung des korrekten Abschnitts zunächst ausreicht, um adäquate Navigationsassistenz in Form einer ausführbaren Navigationsanweisung zu leisten.

Aus diesem Grund kommt in der vorliegenden Arbeit eine Evaluationsmetrik zum Einsatz, die die Güte der Positionsbestimmung anhand der Trefferquote der durch die Segmentierung der Route definierten Areale misst. Auf diese Weise wird berücksichtigt, dass in einem Navigationskontext bestimmte Positionierungsfehler einen gravierenderen Einfluss als andere haben. Weicht die geschätzte Position beispielsweise um 5 m vom tatsächlichen Standort ab, so kann dies je nach Gebäudegeometrie einerseits dafür sorgen, dass die dadurch gegebene Navigationsanweisung eine für den Benutzer nicht sichtbare Landmarke referenziert und somit den Navigationsfortschritt gefährdet. In anderen Situationen, etwa in langen Korridoren oder allgemein in der Mitte von größeren Routenabschnitten, hat eine derartige Ungenauigkeit weniger oder keine Auswirkungen auf den Navigationsprozess.

Die Metrik ähnelt der vom ISO/IEC-18305-Standard genannten zonenbasierten Genauigkeit bzw. dem von Pulkkinen und Verwijnen (2015) vorgeschlagenen Verfahren. Anstelle von vordefinierten Zonen, die sich durch die Gebäudegeometrie ergeben, werden hier jedoch Bereiche, die von der aktuellen Route und den dort vorhandenen Landmarken abhängen, dynamisch generiert.

Die verwendete Evaluationsmetrik trägt daher einen Teil zum Ziel einer nutzerzentrierten Indoor-Positionierung bei, indem der Fokus auf den Situationen und

Zeitpunkten liegt, in denen eine genaue Positionsbestimmung besonders relevant für den Navigationserfolg ist.

Formal betrachtet gibt die Metrik, im Folgenden kurz Trefferquote genannt, die Genauigkeit an, mit der die geschätzten Positionen in die durch die Routenabschnitte definierte diskrete Verteilung klassifiziert werden (siehe Formel 3.19). Angewendet auf die Fußgängernavigation gibt sie somit einen Anhaltspunkt, welcher Prozentsatz der gegebenen Anweisungen (in Bezug auf die Gesamtzeit bzw. alle Positionsupdates) valide sind.

$$\text{TQ} = \frac{|\{p \in P \mid A_{\text{pred}}(p) = A_{\text{GT}}(p)\}|}{|P|} \quad (3.19)$$

3.4.1.2 Offline-Simulationsstudien

Um die Positionierungsgenauigkeit zu evaluieren, werden Offline-Simulationsstudien auf Basis von zuvor aufgezeichneten Sensordaten durchgeführt, um verschiedene Systemkonfigurationen reproduzierbar zu vergleichen.

Für die Simulationen wurde ein Wrapper-Programm entwickelt, welches den Positionierungsalgorithmus mit den zuvor erhobenen Logdaten versorgt sowie dessen Ausgaben und internen Zustand – die Positionsschätzung und Partikelverteilung – für die spätere Analyse aufzeichnet. Durch Parametrisierung anhand einer Konfigurationsdatei wird der Vergleich unterschiedlicher Konfigurationen des Algorithmus ermöglicht.

Der konkrete Ablauf während der Simulation ist wie folgt: Das Programm liest zu Beginn die erhobenen Sensordaten aus den im CSV-Format gespeicherten Logdateien ein und teilt dem Positionierungssystem die geplante Route mit, deren Startpunkt für die Initialisierung der Partikelverteilung dient. Anschließend werden die getätigten Schritte nacheinander durch Aufruf der korrespondierenden Funktion simuliert. Obwohl – wie für Offline-Studien typisch – schon zu Beginn des Experiments alle Daten verfügbar sind, wird im Rahmen dieser Arbeit daraus kein Vorteil gezogen, um die Vergleichbarkeit mit einem in Echtzeit laufenden System zu gewährleisten. Stattdessen werden alle Sensordaten sequenziell und unmittelbar zu einer Positionsschätzung verarbeitet. Das Verfahren greift u. a. beim Resampling auf einen Zufallszahlengenerator zurück. Um den Einfluss von Zufallseffekten zu minimieren, wird der Positionierungs-Algorithmus daher für jeden Testlauf mehrfach mit unterschiedlichen Initialwerten (*random seeds*) durchlaufen.

3.4.2 Verifizierung der grundlegenden Funktionsweise

In der ersten Vorstudie wurde die grundlegende Funktionsweise des Verfahrens anhand eines kontrollierten Experiments verifiziert. Die hauptsächlichen Fragen waren hierbei, ob der Positionsfehler in der gleichen Größenordnung wie bei vergleichbaren Systemen aus der Literatur bleibt und inwiefern die angepasste Evaluationsmethodik zu neuen Erkenntnissen führt.

3.4.2.1 Experimentaufbau und -vorbereitungen

Als Testareal für das Experiment diente das Philosophie-/Theologie-Gebäude (PT) der Universität Regensburg, welches sowohl enge Gänge als auch größere offene Flächen beinhaltet und somit unterschiedliche Bewegungsmöglichkeiten im Indoorbereich abdeckt. Zum Zeitpunkt der Studie existierte die in Abschnitt 2.4 beschriebene Navigations-App und somit die Anbindung an den URwalking-Routenserver noch nicht. Die 182 m lange Testroute, die durch vier Korridore im Erdgeschoss sowie mehrere Foyers und Durchgangstüren führte, wurde daher manuell festgelegt und in Abschnitte unterteilt. Hierfür identifizierten 19 Personen insgesamt 32 markante Objekte im Testareal und bewerteten diese in einem Verfahren, das sich an Kattenbeck (2016) orientierte, hinsichtlich ihrer Salienz. Die Objekte mit der höchsten Salienz – darunter beispielsweise eine Vitrine, ein Wandgemälde, eine Sitzbank und ein Schild zum Psychologie-Lehrstuhl – wurden gemeinsam mit anderen, architektonischen Merkmalen zur Unterteilung in 14 verschiedene Routenabschnitte verwendet (siehe Abbildung 3.5). Die Größe der Abschnitte variiert und ist durch die unterschiedliche Sichtbarkeit der Landmarken begründet.

3.4.2.2 Datenaufzeichnung

Anstelle der – wie oben erwähnt – noch nicht verfügbaren Navigations-App kam für die Aufzeichnung der benötigten Sensordaten von Schrittkenner und GRV eine eigens entwickelte Android-Applikation zum Einsatz. Zudem wurden mit der rückseitigen Kamera des Geräts Videoaufnahmen angefertigt, mit denen später der Experimentablauf nachvollzogen werden konnte. Während der Aufzeichnung war kein Positionierungsalgorithmus auf dem Gerät aktiv, um die VersuchsteilnehmerInnen nicht mit potentiell falschen Positionsschätzungen zu verwirren.

Die Benutzeroberfläche war einfach gehalten und zeigte neben der Kartendarstellung, auf der der aktuelle Abschnitt farblich hervorgehoben war, nur eine Schaltfläche an, die mit der Bezeichnung der momentan relevanten Landmarke beschriftet war. Das Erreichen der Landmarke und somit eines neuen Routenab-

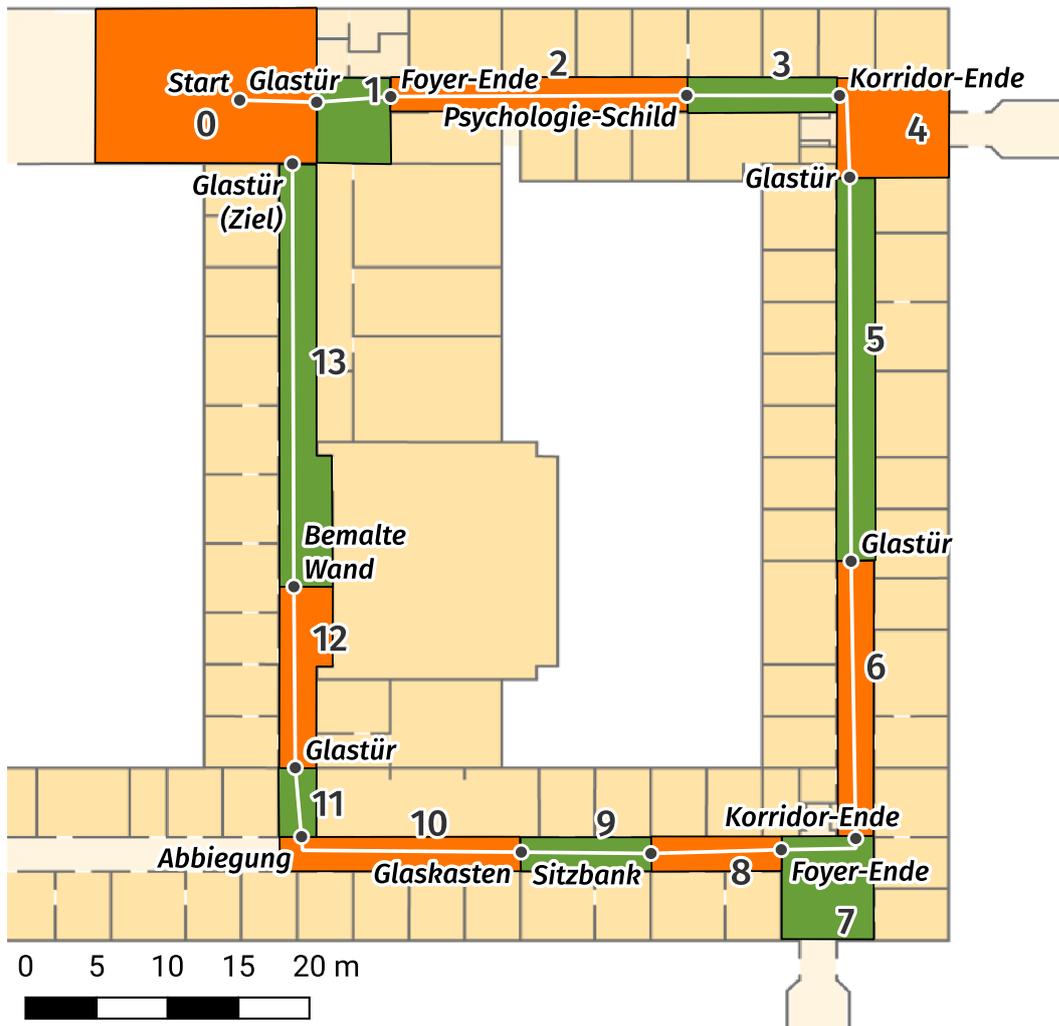


Abbildung 3.5: Das Testareal des ersten Experiments mit 14 durch Landmarken und architektonische Einheiten definierten Abschnitten

schnitts quittierten die VersuchsteilnehmerInnen selbstständig per Klick auf die Schaltfläche, woraufhin zum nächsten Abschnitt gewechselt wurde. Durch diesen Mechanismus wurden gleichzeitig Zeitstempel erhoben, die später als Ground Truth dienen.

Die Datenaufzeichnung fand im Laufe einer Woche im Sommer 2017 statt. Als Testpersonen standen sieben MitarbeiterInnen des Lehrstuhls für Informationswissenschaft im Alter von 25 bis 43 Jahren zur Verfügung, die mit dem Areal und den Landmarken vertraut waren und insgesamt 15 Läufe entlang der Testroute absolvierten. Die Testpersonen wurden in geringem Abstand vom Autor als Experimentleiter begleitet und gebeten, das Gerät während des Experiments in Gehrichtung vor sich zu halten. Ein LG Nexus 5X mit Android 7.1.2 als Be-

triebssystem diente als Testgerät. Dieses verfügt ebenso wie die später verwendeten Google Pixel über einen eingebauten Schrittkenner.

3.4.2.3 Analyse und Verifizierung der Sensordaten

Zur Verifizierung, ob die erhobenen Sensordaten repräsentative Bewegungsmuster abbilden, wurden einige statistische Kennwerte berechnet. Die mittlere Bewegungsgeschwindigkeit als Quotient von Routenlänge und Dauer eines Durchlaufs beträgt 1,30 m/s (SD=0,14 m/s) und liegt somit im Rahmen normaler Gehgeschwindigkeit (Öberg et al., 1993, S. 216). Bei der Kontrolle der Schrittkennung anhand der Videoaufzeichnungen zeigte sich, dass im Mittel 5,8 % der Schritte nicht vom integrierten Sensor erkannt wurden. Nach Korrektur der Schrittzahl betrug die mittlere errechnete Schrittlänge 0,73 m – etwas höher als die im System angenommene, voreingestellte Länge.

Da alle VersuchsteilnehmerInnen mit dem eng begrenzten Testareal gut vertraut waren, führte die selbstständige Annotation der Ground Truth zu ausreichend genauen Referenzpositionen mit einem Positionsfehler in der Größenordnung von ca. 1 m, was ebenfalls mithilfe der Videoaufzeichnungen bestätigt wurde.

3.4.2.4 Ergebnisse

Mit den so verifizierten Eingabe-Daten wurden nun jeweils zehn Durchläufe im Simulations-Tool durchgeführt. Die oben beschriebene Evaluationsmetrik gibt im Folgenden den Anteil der korrekt klassifizierten Routenabschnitte an. Zudem ermöglicht sie es, besonders problematische Bereiche auf der Route zu identifizieren.

Über alle Testläufe hinweg stimmen insgesamt 74,7 % aller geschätzten Positionen mit ihrem Ground-Truth-Abschnitt überein. Wie Abbildung 3.6 zeigt, erwiesen sich vor allem die Bereiche nach Abbiegungen als problematisch. In den Abschnitten 8, 9 und 11 (siehe Abbildung 3.5) wurden mit 43,9 %, 56,1 % respektive 12,9 % die niedrigsten Werte erreicht.

Bei der Betrachtung der einzelnen Testläufe zeigt sich in den meisten Fällen nur eine geringe Streuung über die zehn Einzelsimulationen hinweg (Abbildung 3.7). Mit Ausnahme der Testläufe 12 und – zu einem geringeren Grad – 8 bewegen sich alle innerhalb weniger Prozentpunkte um den Mittelwert. Eine nähere Betrachtung des Fehlerfalls in Testlauf 12 zeigt, dass dort vergleichsweise wenige, lange Schritte registriert wurden, wodurch die Notwendigkeit einer (in diesem Experiment noch nicht aktiven) Schrittlängenadaption unterstrichen wird.

Für den Vergleich mit anderen Arbeiten wurden auch metrische Fehlermaße berechnet. Unter Annahme einer konstanten Bewegung zwischen den bekannten

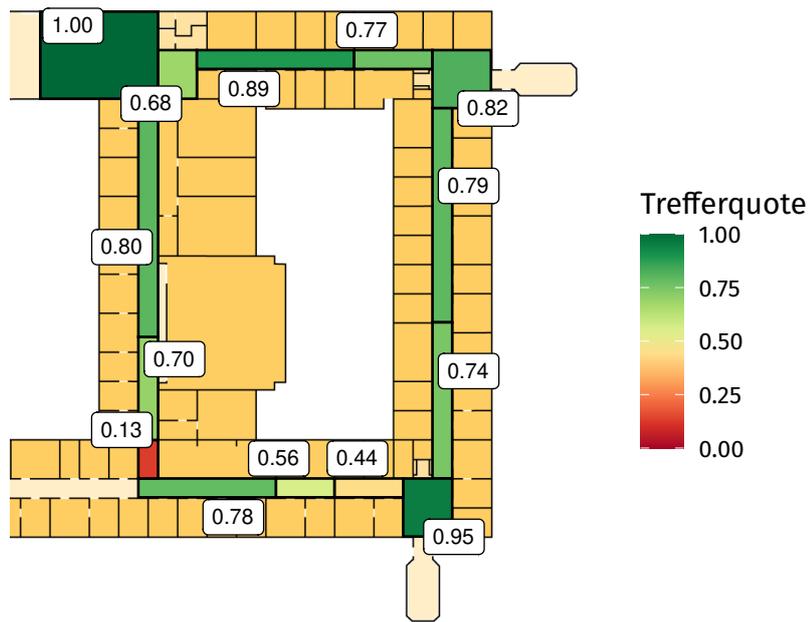


Abbildung 3.6: Trefferquote des Basis-Positionierungssystems

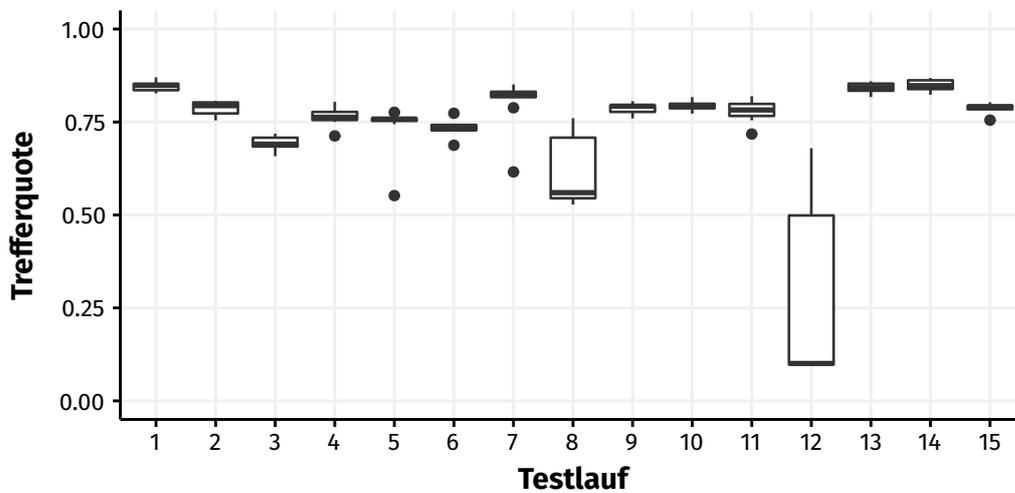


Abbildung 3.7: Streuung der Trefferquote in den 15 Testläufen

Checkpoints wurde hierfür die Ground Truth interpoliert, um zu jedem Zeitpunkt die Distanz zur geschätzten Position bestimmen zu können. Im Median beträgt der Positionsfehler über alle Testläufe hinweg 2,60 m, das 75%-Quantil liegt bei 4,37 m.

3.4.2.5 Detailbetrachtungen

Ausgehend von der Beobachtung, dass die Bereiche unmittelbar nach Abbiegungen die niedrigsten Trefferquoten aufweisen, wurde in einer Detailbetrachtung der Einfluss einer alternativen Modellierung des Navigationsgraphs untersucht.

Hierfür wurde die Abbiegung zwischen den Abschnitten 10 und 11 mit zusätzlichen schrägen Kanten verbunden (vgl. Abbildung 3.8), was zweierlei Vorteile mit sich bringt: Zum einen werden dadurch Pfade modelliert, in denen der Fußgänger die Kurve nicht im 90°-Winkel nimmt, sondern leicht anschneidet. Zum anderen wird eine bessere Kompensation unterschiedlicher Schrittlängen ermöglicht, da nun mehrere Kanten in den abzweigenden Korridor führen.

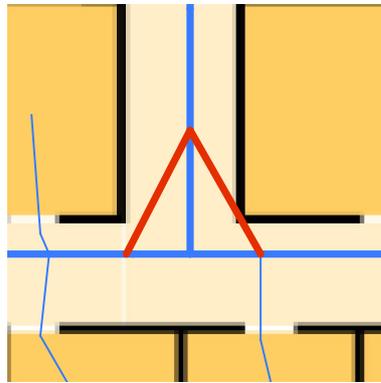


Abbildung 3.8: Zusätzlich eingefügte Kanten
(Detailansicht Abschnittswechsel 10/11)

In der Tat erhöht sich die Trefferquote in Abschnitt 11 nach der Modifikation von 12,9 % auf 35,9 %. Betrachtet man von diesem Abschnitt an beginnend den Rest der Route, so zeigt sich eine Verbesserung hinsichtlich der erreichten Trefferquote. Um dies auf statistische Signifikanz zu testen, wird ein Wilcoxon-Rangsummentest (Bortz & Schuster, 2010, S. 133 f.) durchgeführt, welcher einen signifikanten Unterschied ($W = 7086$, $p < 0,01$) bei moderater Effektstärke ($r = 0,32$) anzeigt.

3.4.2.6 Fazit

Abschließend lässt sich konstatieren, dass durch die erste Studie der Nachweis der grundlegenden Funktionalität erbracht ist. Während die Positionierung freilich nicht perfekt arbeitet, so ist doch beispielsweise an der zum Ende der Route hin zunehmenden Genauigkeit die Fähigkeit des Systems zur Selbstkorrektur nach fehlerhaft bestimmten Abschnitten zu erkennen. Der vergleichsweise berechnete metrische Positionsfehler befindet sich in einer für PDR-Systeme erwarteten und

für die Indoornavigation in den meisten Fällen ausreichenden Größenordnung.

Die Detailbetrachtung zeigt anhand eines anschaulichen Beispiels, dass es für die Positionsverfolgung von Vorteil ist, wenn der Navigationsgraph nicht nur die logischen Relationen zwischen Orten modelliert, sondern auch die Geometrie der Räumlichkeiten berücksichtigt. Auf diese Weise kann mit einfachen Mitteln der tatsächlich vom Fußgänger beschrittene Pfad besser erkannt werden. Diese Modellierungsform wird daher auch in den weiteren Studien angewendet.

3.4.3 Studie unter realistischen Bedingungen

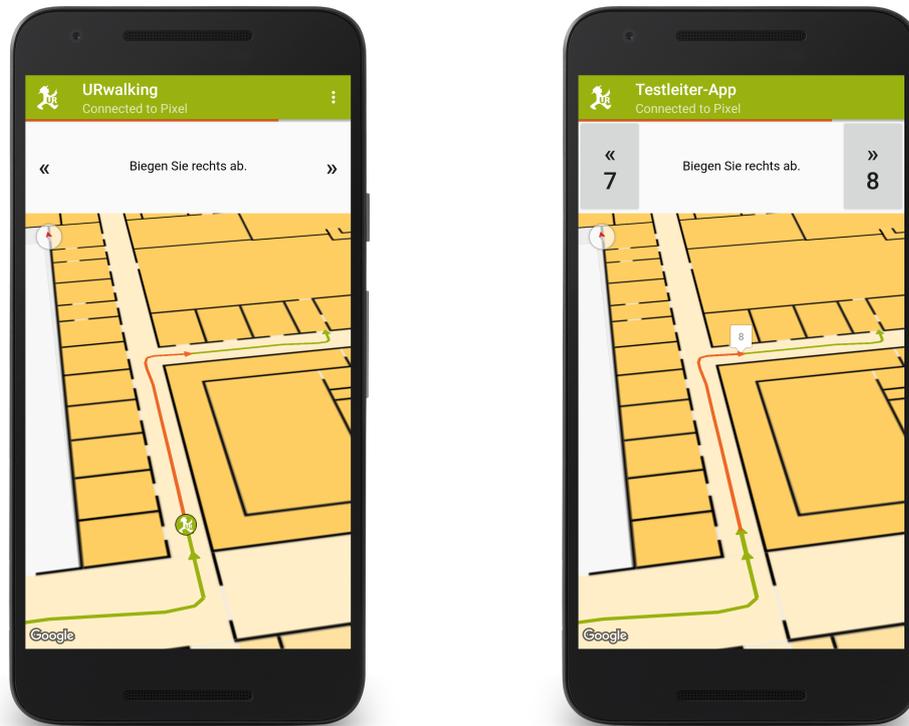
Aufbauend auf der ersten Vorstudie wurde ein zweites Experiment durchgeführt, bei dem der Fokus vor allem auf der Erzeugung möglichst realistischer Bedingungen lag, wie sie bei tatsächlicher Navigation mit einer mobilen App der Fall sind. Weiterhin wurden in dieser Studie die in Abschnitt 3.3.3 beschriebenen Anpassungen evaluiert, um die Grenzen rein PDR-basierter Positionierung zu erforschen.

3.4.3.1 Experimentaufbau

Anders als in der vorigen Studie kam nun der Prototyp einer vollwertigen Navigationsanwendung zum Einsatz, aus dem später die URwalking-App (siehe Abschnitt 2.4) hervorgehen sollte.

Die im ersten Experiment praktizierte Annotation der Ground Truth durch die VersuchsteilnehmerInnen ist zwar für den Zweck einer schnellen Datenerhebung vorteilhaft, erfordert jedoch eine gewisse Ortskenntnis seitens der Testperson. In der Literatur wird zur Lösung dieser Problematik oft auf aufgemalte oder aufgeklebte Markierungen am Boden zurückgegriffen, deren Passieren die Testperson per Druck auf einen Button o. Ä. bestätigen muss (siehe z. B. Torres-Sospedra et al., 2017). Dies zieht jedoch ebenfalls Aufmerksamkeit auf sich, lenkt vom eigentlichen Navigationstask ab und ist somit der angestrebten Realitätsnähe abträglich.

Ein separates Positionierungssystem zur Verifizierung der Ground-Truth-Daten, wie z. B. von Le Scornec et al. (2017) vorgeschlagen, stand für die Experimente nicht zur Verfügung. Stattdessen wurde in der zweiten Studie die Ground Truth durch einen ortskundigen Experimentleiter erhoben, der der Testperson in kurzem Abstand folgte. Die so festgelegte Position wurde zugleich auf das Smartphone der Testperson übertragen und dort zum Wechseln der Navigationsanweisungen verwendet. Konzeptuell ähnelt der Aufbau somit einem Wizard-of-Oz-Experiment – mit dem Unterschied, dass den VersuchsteilnehmerInnen die Präsenz und der Einfluss des Testleiters bewusst war.



(a) App der Testperson mit Navigationsanweisungen, aktuellem Routenabschnitt und Anzeige des Standorts

(b) App des Testleiters mit Buttons zum Wechseln der Navigationsanweisung

Abbildung 3.9: Experimentaufbau mit zwei per Bluetooth verbundenen Geräten

Für den Zweck der Fernsteuerung der Navigations-App wurde eine separate Android-App entwickelt, die sich per Bluetooth mit der Navigations-App verbindet. Die Navigations-App zeigte während des Experiments die geschätzte Position entlang der Route an, war aber so konfiguriert, dass nur durch Zutun des Experimentleiters der Abschnitt gewechselt werden konnte. Abbildung 3.9 illustriert den aus zwei Geräten bestehenden Experimentaufbau.

Drei verschiedene Testrouten, die nun direkt vom Routingalgorithmus der Campus-Navigation URwalking stammten, führten über mehrere Stockwerke des PT-Gebäudes der Universität Regensburg. Sie variierten in ihrer Länge zwischen 151 und 190 m und beinhalteten wiederum für öffentliche Gebäude typische architektonische Merkmale wie enge Korridore und offene Freiflächen, auf denen teils kleinen Hindernissen und FußgängerInnen ausgewichen werden musste. Entlang der Routen wurden mehrere Treppenhäuser und Durchgangstüren passiert. Ab-

bildung 3.10 illustriert die Routenverläufe. Route 1 beginnt an einem der Gebäudeeingänge und endet in einem Bereich im Erdgeschoss mit wenig Durchgangsverkehr. Die zweite Route schließt direkt daran an und führt über ein Treppenhaus zu einem Büro im zweiten Stockwerk. Von dort aus verläuft Route 3 wiederum über ein anderes Treppenhaus zurück ins Erdgeschoss und in die Nähe eines Ausgangs.

Die in den neuen Testrouten enthaltenen Stockwerkswechsel in Form von Treppenhäusern werden in diesem Experiment nicht berücksichtigt. Im Umgebungsmodell sind Treppenhäuser lediglich als vertikale Kanten modelliert, weswegen mittels PDR kein direktes Mapping der erkannten Bewegung auf den Graph möglich ist. Die Erkennung von Treppen und anderen ortsbezogenen Aktivitäten wird später in Kapitel 5 behandelt.

3.4.3.2 Datenaufzeichnung

Die Datenaufzeichnung erfolgte im Juni 2018 in einem Zeitraum von wenigen Tagen. Als VersuchsteilnehmerInnen wurden sechs Studierende der Informationswissenschaft rekrutiert, davon waren zwei weiblich. Testperson und Experimentleiter benutzten Google-Pixel-Geräte mit Android 8.1. Um für möglichst realistische Umstände zu sorgen, wurde den Testpersonen nicht vorgeschrieben, wie das Smartphone in der Hand zu halten ist. Dennoch wurde das Gerät mit Ausnahme von kurzen Passagen von allen TeilnehmerInnen vor dem Körper getragen.

Nach dem Absolvieren jeder Route wurden die VersuchsteilnehmerInnen gebeten, ihre Vertrautheit mit den einzelnen Routenabschnitten auf einer fünfstufigen endpunktbenannten Likert-Skala von »Gar nicht« bis »Sehr« zu bewerten. Die Frage lautete »Wie vertraut waren Sie vor dem Experiment mit diesem Abschnitt?«. Route 1 wurde im arithmetischen Mittel mit 2,80 Punkten ($SD = 1,75$; $Mdn = 2,5$) bewertet, Route 2 war mit 2,08 Punkten ($SD = 1,38$; $Mdn = 1$) erwartungsgemäß am wenigsten bekannt, und Route 3 erreicht 2,89 Punkte ($SD = 1,91$; $Mdn = 2$).

Die sechs Testläufe auf allen drei Routen ergaben insgesamt 55 min bzw. etwa 3 km an Navigations-Logdaten.

3.4.3.3 Ergebnisse

Wie im ersten Experiment bilden die erhobenen Sensordaten die Grundlage zur Durchführung von Simulationsstudien. Auf den komplexeren Routen und mit dem realistischeren Experimentaufbau werden nun die in Abschnitt 3.3.3 beschriebenen Anpassungen evaluiert, mithilfe derer sich das Indoor-Positionierungssystem an den Nutzer adaptiert. Im Einzelnen sind dies die Bewegung in Freiflächen, die Schrittlängenadaptation und die Korrektur der Initialrichtung, für die im Folgenden drei Hypothesen formuliert und getestet werden.

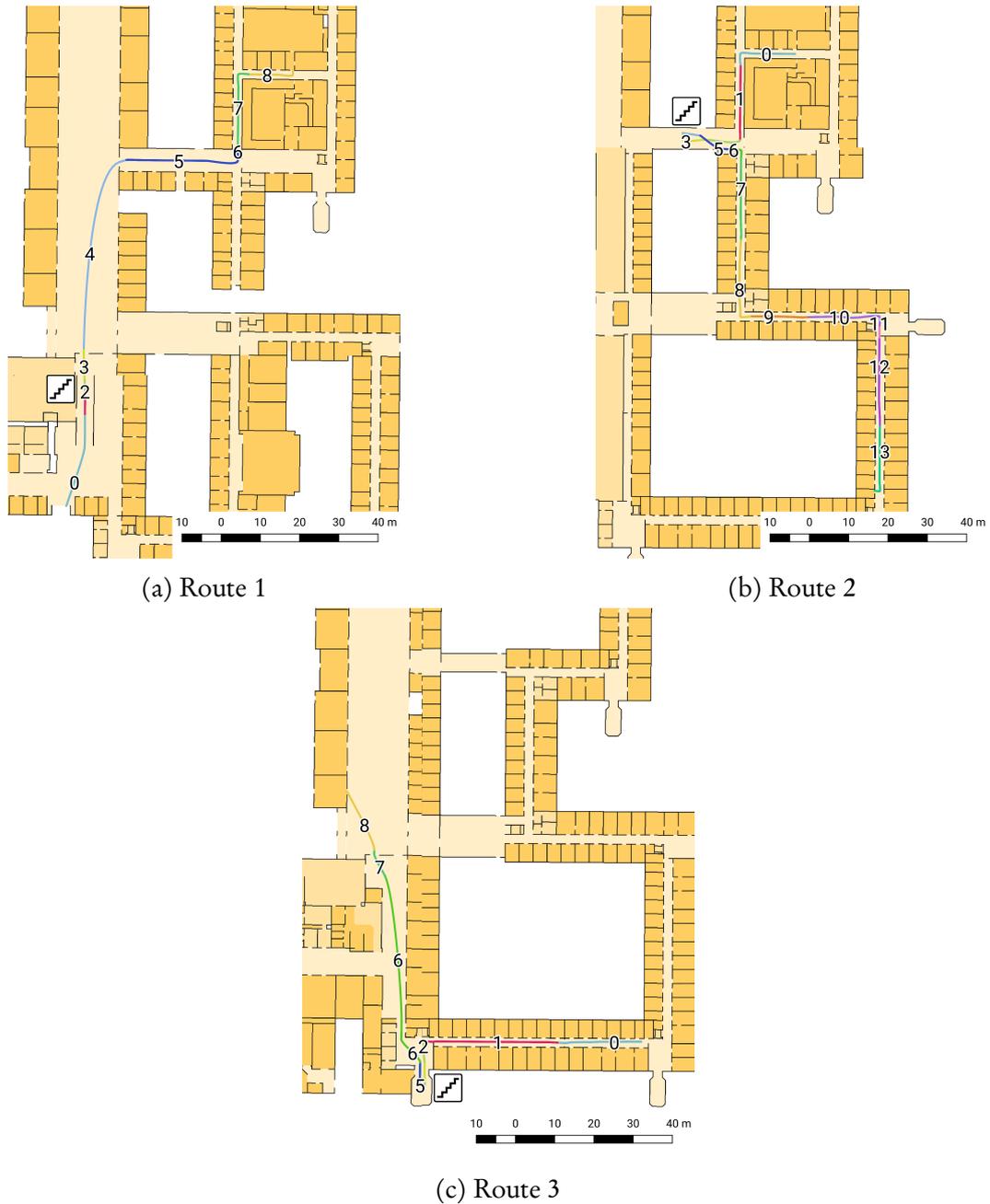


Abbildung 3.10: Die drei Testrouten im PT-Gebäude für Experiment 2. Treppensymbole kennzeichnen die Orte, an denen Stockwerkswechsel stattfinden.

Vor dem Test der Hypothesen wird der Einfluss der Partikelanzahl auf die Positionierungsgenauigkeit untersucht. Je mehr Partikel eingesetzt werden, desto besser kann der Zustandsraum abgedeckt werden. Bei Verwendung von zu wenigen Partikeln sinkt die Wahrscheinlichkeit, dass eine der durch sie repräsentierten Hypothesen über den tatsächlichen Zustand zutrifft. Im Verlauf mehrerer Simulati-

onsdurchläufe äußert sich dies durch starke Streuung der erreichten Genauigkeit. Für belastbare Aussagen muss zunächst die Partikelanzahl so gewählt werden, dass die Varianz über mehrere Durchläufe hinweg gering bleibt. Freilich kann die Anzahl nicht beliebig erhöht werden, da mit ihr auch der Bedarf nach Rechenleistung und Speicher ansteigt. Um die optimale Partikelanzahl zu bestimmen, wurden Simulationen mit 30, 100, 300, 1000 und 3000 Partikeln durchgeführt.

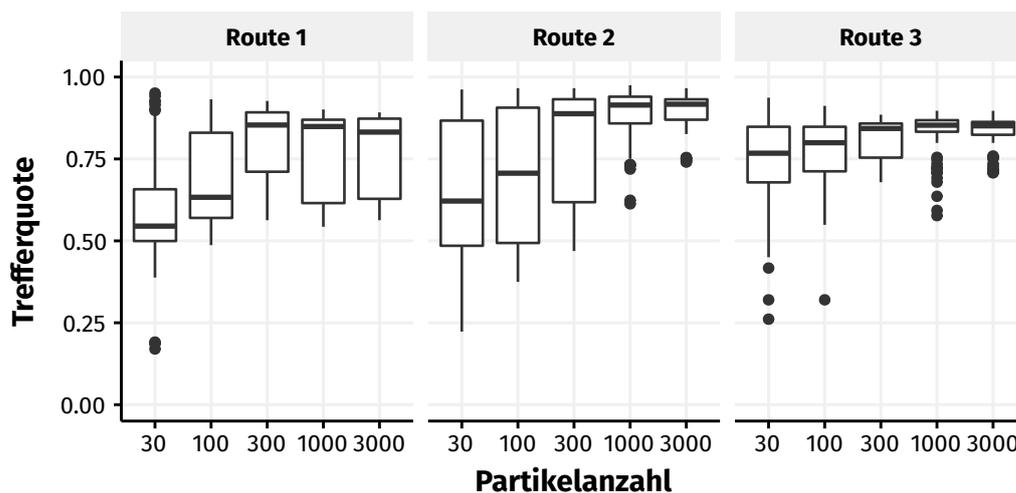


Abbildung 3.11: Auswirkung der Partikelanzahl auf die Positionierungsgenauigkeit

Abbildung 3.11 zeigt den Zusammenhang zwischen der erreichten Trefferquote und der Partikelanzahl für die drei Testrouten. Mit steigender Partikelanzahl erhöht sich – wie zu erwarten – die mittlere Genauigkeit von 65,9% (30 Partikel) auf 83,3% (3000 Partikel), während die Varianz von 3,6 auf 1,0 Prozentpunkte sinkt. Als Kompromiss zwischen Positionierungsgenauigkeit und benötigter Dauer bzw. Rechenleistung werden die weiteren Tests mit 1000 Partikeln durchgeführt – die Trefferquote liegt hier im Mittel bei 82,6%, die Varianz bei 1,2 Prozentpunkten. Ein Wilcoxon-Rangsummentest deutet auf keinen statistisch signifikanten Unterschied der beiden Konfigurationen mit 1000 und 3000 Partikeln hinsichtlich der erreichten Trefferquote hin ($W = 16168, p = 1$).

Bewegung in Freiflächen

Nach diesen Vorbereitungen werden nun die Anpassungen des Positionierungssystems an realistisches Benutzerverhalten evaluiert. Die erste Hypothese betrifft die ungehinderte Bewegung auf offenen Flächen, welche mithilfe von Arealodes umgesetzt ist:

- H_1 : Die Positionierungsgenauigkeit verbessert sich durch ungehinderte Bewegung auf offenen Flächen.
- H_0 : Die Positionierungsgenauigkeit ist bei ungehinderter Bewegung auf offenen Flächen nicht besser.

Für den Test der Hypothese werden drei Varianten erprobt:

- a) Das in Abschnitt 3.3.3 beschriebene hybride Modell aus Graphstruktur und Polygonen, welches komplett freie Bewegung in den so gekennzeichneten Flächen ermöglicht – mit dem Nachteil der fehlenden Stabilisierungsmöglichkeit durch den Graph.
- b) Eine Variante, in der innerhalb der gekennzeichneten Fläche ein vollständig vernetzter Graph erzeugt wird, in dem alle Randknoten bidirektional miteinander verbunden sind. Dies entspricht der Konfiguration, die für die Routenberechnung verwendet wird. Mithilfe der im Partikelzustand enthaltenen Verteilung aller aktuell möglichen Kanten sind auch hier Pfade abbildbar, die nicht auf direktem Weg zum Ausgangsknoten führen. Probleme entstehen jedoch insbesondere bei Flächen mit hohem Verzweigungsgrad, in denen sich die Winkel der ausgehenden Kanten nur geringfügig unterscheiden.
- c) Ein Modell, in dem die Fläche explizit mit Knoten und Kanten modelliert ist, um die dort begehbaren Wege möglichst umfassend abzubilden. Es entsteht somit eine Mesh- oder Gitterstruktur, wie sie z. B. bei F. Ebner, Deinzer und Lukas (2014) verwendet wird. In diesem Fall sind die Randknoten durch eine oder mehrere Kanten miteinander verbunden. Aufgrund der damit vorgenommenen Diskretisierung des Raums können nicht alle gegangenen Pfade realitätsgetreu abgebildet werden. Ein wichtiger Vorteil gegenüber dem vollständig vernetzten Graph ist der niedrigere Verzweigungsgrad an den einzelnen Knoten.

Abbildung 3.12 illustriert die drei Modellierungs-Varianten anhand des in Route 1 enthaltenen Foyers im Erdgeschoss des PT-Gebäudes. In der mittleren Darstellung ist zu beachten, dass hier aus Gründen der Übersichtlichkeit nur diejenigen Kanten abgebildet sind, die vom Startknoten am unteren Rand ausgehen. In der erzeugten Graphstruktur sind auch alle anderen Randknoten der Freifläche auf direktem Weg miteinander verbunden.

Die mittlere Trefferquote über alle Routen hinweg beträgt für die Polygon-Variante 81,9 %, für die Kanten-Variante 56,4 % und die Gitter-Variante 74,4 %. Es liegt dabei – wie durch Shapiro-Wilk-Tests überprüft – keine Normalverteilung

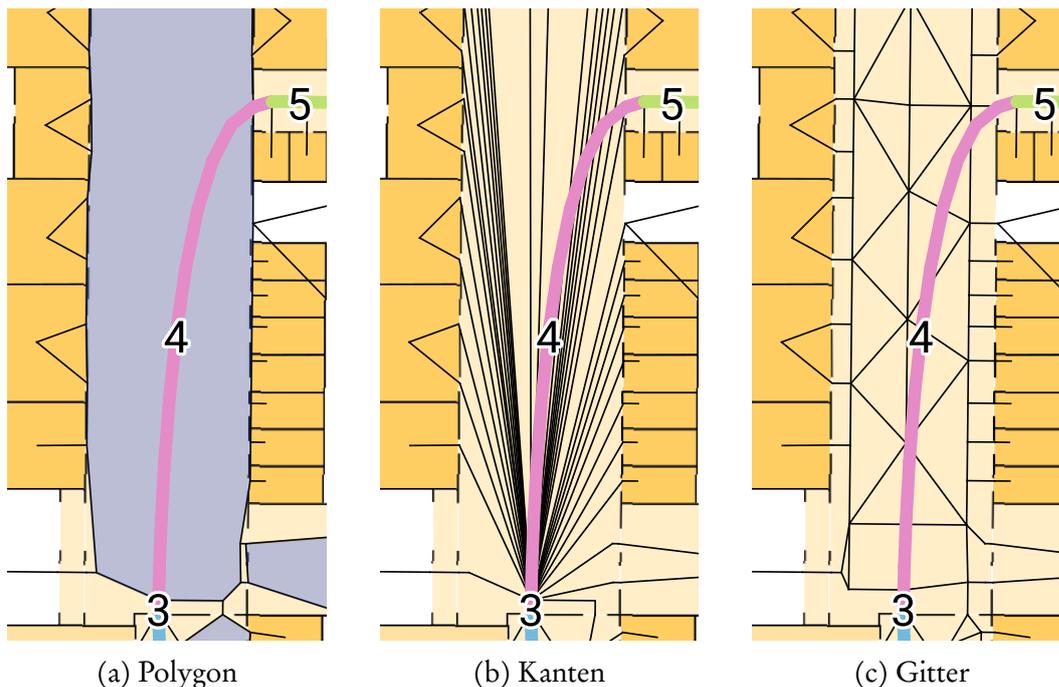


Abbildung 3.12: Beispiele für die drei Modellierungs-Varianten im PT-Foyer

vor; ebenso muss die Annahme der Varianzhomogenität aufgrund eines signifikanten Levene-Tests abgelehnt werden.¹ Um die Unterschiede zwischen den Konfigurationen auf statistische Signifikanz zu testen, kommt deshalb ein Kruskal-Wallis-Rangsummentest zum Einsatz, welcher ohne die Annahme einer parametrischen Verteilung auskommt (Kruskal & Wallis, 1952). Dieser weist – über alle Routen hinweg betrachtet – auf statistisch signifikante Unterschiede hin ($\chi^2(2) = 109; p < 0,01$), bei einer großen Effektstärke (η_H^2) von 0,20 (Tomczak & Tomczak, 2014, S. 24). Ein post-hoc durchgeführter Dunnett-Tukey-Kramer-Test (Dunnett, 1980) ist auf einem α -Niveau von 0,05 paarweise zwischen allen Konfigurationen signifikant.

Bei Betrachtung der einzelnen Routen zeigen sich erwartungsgemäß Unterschiede: Da in Route 2 keine größeren Freiflächen enthalten sind, ist hier kein statistisch signifikanter Unterschied zwischen den Konfigurationen vorhanden ($\chi^2(2) = 0,06; p = 1$), während bei den restlichen Routen der Einfluss der unterschiedlichen Modellierung deutlich erkennbar ist (siehe Abbildung 3.13).

Zusammenfassend zeigt sich, dass die in diesem Experiment gegangenen Pfade durch das hybride Umgebungsmodell, bestehend aus Graphstruktur und Polygonen, am besten repräsentiert werden konnten. Gegenüber den alternativen Mo-

¹Dies gilt analog für alle folgenden Hypothesentests in diesem Abschnitt und wird im Weiteren nicht gesondert erwähnt.

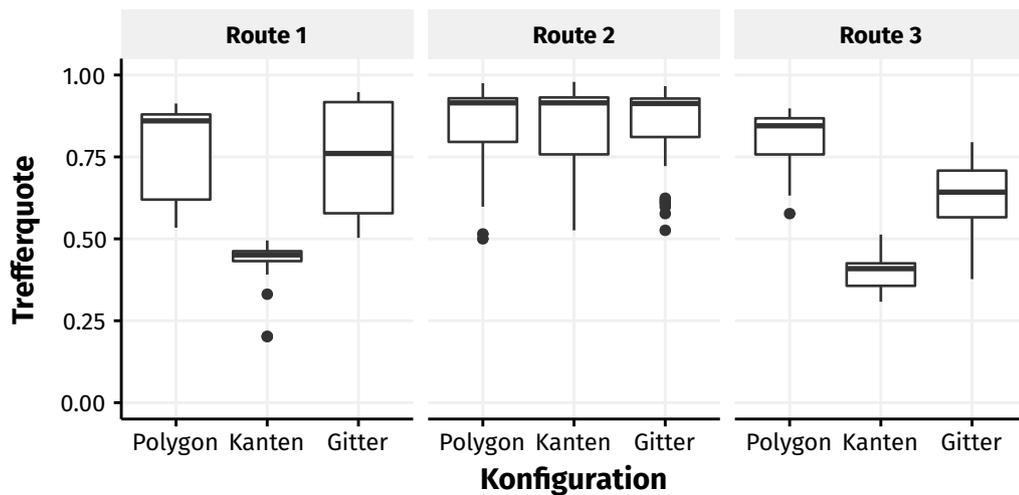


Abbildung 3.13: Auswirkung der Modellierung von Freiflächen auf die Positionierungsgenauigkeit

dellierungsformen ergibt sich eine signifikante Verbesserung der Positionierungsgenauigkeit, sodass die Null-Hypothese verworfen und die Alternativhypothese angenommen werden kann.

Schrittlängenadaption

Die zweite Hypothese adressiert den Mechanismus der Schrittlängen-Variation zur Adaption an unterschiedliche NutzerInnen.

- H_1 : Die Positionierungsgenauigkeit verbessert sich durch Adaption an die Schrittlänge.
- H_0 : Die Positionierungsgenauigkeit ist bei Adaption an die Schrittlänge nicht besser.

Um die Hypothese zu testen, werden drei verschiedene Konfigurationen erstellt: In der ersten bleibt die Schrittlänge für alle Partikel konstant auf dem voreingestellten Wert. Die restlichen beiden Konfigurationen geben den Partikeln initial einen zufälligen Skalierungsfaktor für die Schrittlänge mit, welcher entweder geringe ($\pm 10\%$) oder hohe ($\pm 25\%$) Abweichungen erlaubt. Ersteres entspricht dem von Rai et al. (2012, S. 301) verwendeten Wert, Letzteres stellt das obere Ende der von Öberg et al. (1993) beobachteten Spannweite dar.

Aus den in Abbildung 3.14 sichtbaren Ergebnissen geht hervor, dass sich die erreichte Genauigkeit nur geringfügig zwischen den Konfigurationen unterscheidet. Ein durchgeführter Kruskal-Wallis-Rangsummentest weist demnach – über alle Routen hinweg – nicht auf statistisch signifikante Unterschiede hin ($\chi^2(2) =$

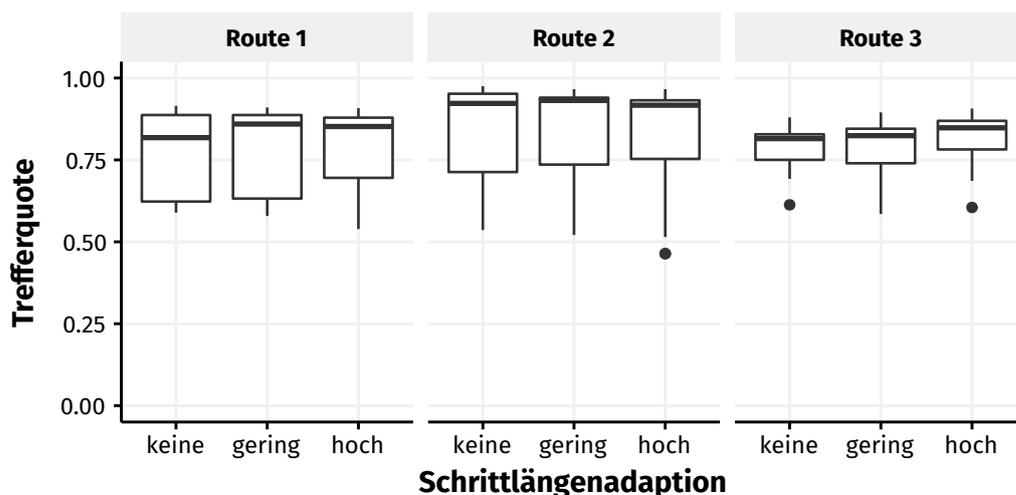


Abbildung 3.14: Auswirkung der Schrittlängenadaption auf die Positionierungsgenauigkeit

2; $p = 0,4$). Bei individueller Betrachtung der drei Testrouten zeigt sich nur bei Route 3 ein signifikantes Ergebnis ($\chi^2(2) = 12; p < 0,01$), jedoch mit geringer Effektstärke ($\eta_H^2 = 0,06$). Ein post-hoc durchgeführter Dunnett-Tukey-Kramer-Test ist dort auf einem α -Niveau von 0,05 jeweils zwischen keiner und hoher bzw. geringer und hoher Schrittlängenadaption signifikant, die Differenz der Mittelwerte beträgt jeweils 2,5 Prozentpunkte.

Die Schrittlängenadaption entfaltet in den meisten Fällen nicht den gewünschten Effekt bzw. ist in diesen Testrouten für eine adäquate Positionsverfolgung nicht nötig. Die obige Nullhypothese kann anhand der Ergebnisse also nicht verworfen werden. In einer späteren Studie (Abschnitt 5.4.2) wird der Einfluss der Schrittlänge noch einmal auf einer längeren Teststrecke evaluiert.

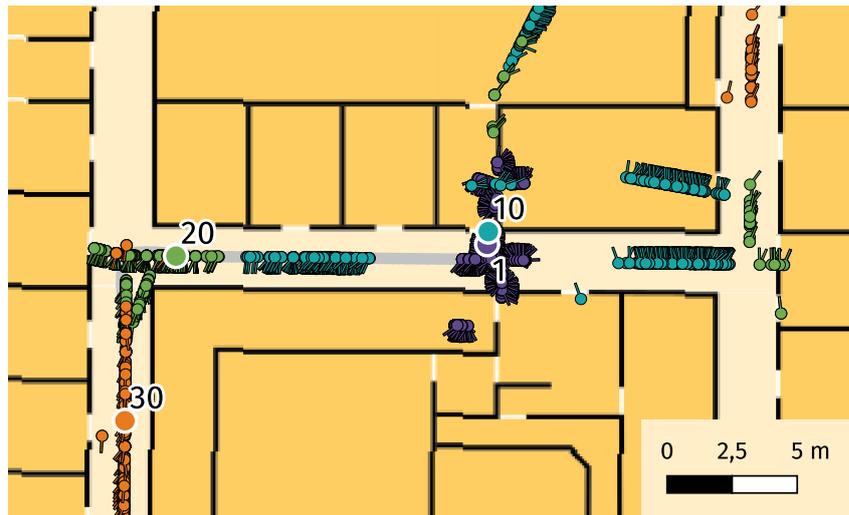
Korrektur der Initialrichtung

Die dritte Hypothese schließlich widmet sich der Richtungskorrektur anhand der aufgrund des Routenverlaufs erwarteten initialen Gehrchtung.

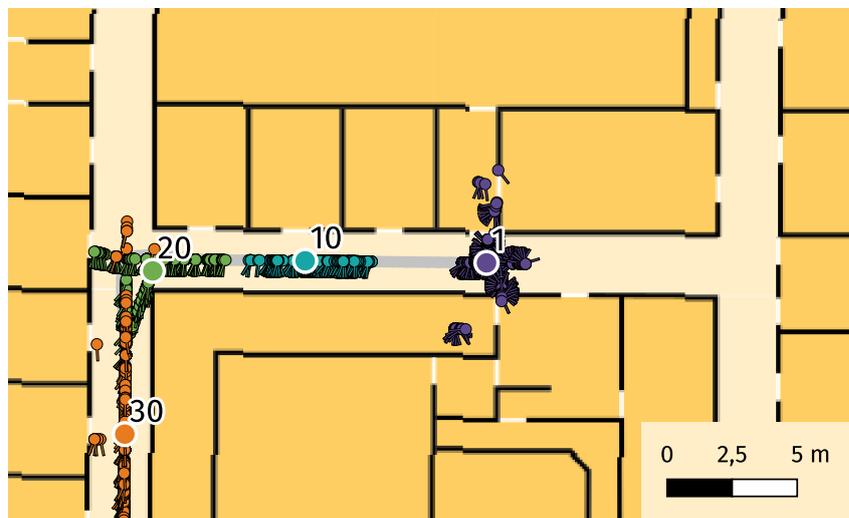
- H_1 : Die Positionierungsgenauigkeit verbessert sich durch die Korrektur der Initialrichtung.
- H_0 : Die Positionierungsgenauigkeit ist durch die Korrektur der Initialrichtung nicht besser.

Hierfür werden wiederum drei Systemvarianten miteinander verglichen: In der ersten Konfiguration findet keine Richtungskorrektur statt. Bei den ersten Schritten bewegen sich die Partikel zunächst gemäß des während der Initialisierung ver-

gebenen gleichverteilten Richtungsbias in alle Richtungen, bis aufgrund des Umgebungsmodells allmählich der korrekte Bias-Wert erlernt bzw. gefiltert wird. Die restlichen beiden Konfigurationen verkürzen diesen Prozess, indem zu Beginn (10 bzw. 20 Schritte oder ca. 5 respektive 10 m) angenommen wird, dass sich der Benutzer entlang der Route bewegt. Der Bias kann somit unabhängig vom Umgebungsmodell geschätzt werden.



(a) ohne Richtungskorrektur



(b) mit Richtungskorrektur

Abbildung 3.15: Verlauf von Partikelverteilung und geschätzter Position für einen ausgewählten Simulationsdurchlauf

Das resultierende Verhalten lässt sich anhand einer Detailstudie veranschaulichen: Abbildung 3.15 zeigt den Beginn eines Simulationsdurchlaufs auf Route 2 (graue Linie), jeweils ohne und mit Richtungskorrektur (in diesem Beispiel 20

Schritte). In beiden Fällen sind die Partikelverteilungen nach 1, 10, 20 und 30 Schritten sowie die dazugehörigen geschätzten Positionen abgebildet. Der größte Unterschied ist in diesem Beispiel bei Schritt 10 erkennbar: In der Variante ohne Richtungskorrektur liegt die geschätzte Position hier noch nahe der Ausgangsposition, da sich die Partikel initial in alle Richtungen bewegen, während sich im anderen Fall alle Partikel entlang der Route orientieren. Schon bei Schritt 20 jedoch sind die geschätzten Positionen beider Varianten nahezu identisch, da viele der sich nach Osten (im Bild rechts) bewegenden Partikel aufgrund der Gebäudegeometrie auf Hindernisse stoßen. Da sich beide Positionen bei Schritt 10 im Bereich des ersten Routenabschnitts befinden, hat diese vorübergehende Ungenauigkeit keine Auswirkungen auf die Trefferquote. Dies lässt sich durch die Tatsache rechtfertigen, dass die Navigationsanweisung (»*Biegen Sie links ab.*«) an beiden Positionen sinnvoll ausführbar ist.

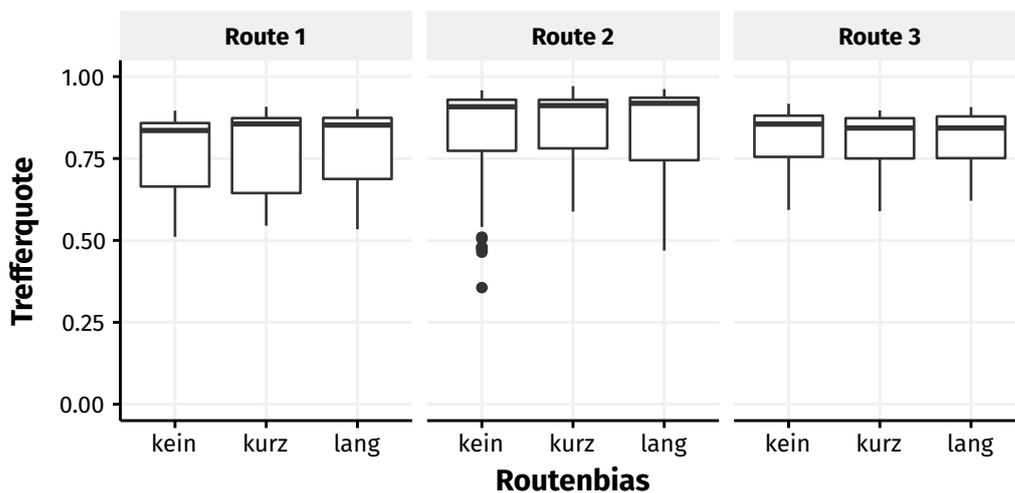


Abbildung 3.16: Auswirkung der Richtungskorrektur auf die Positionierungsgenauigkeit

Im allgemeinen Fall, d. h. über alle Routen und Simulationsdurchläufe hinweg betrachtet, sind die Unterschiede hinsichtlich der erreichten Trefferquote nicht signifikant ($\chi^2(2) = 0,3; p = 0,9$). Gleiches gilt für die Routen im Einzelnen (siehe Abbildung 3.16). Zusammenfassend kann die Nullhypothese hier also nicht verworfen werden. Dies bedeutet im Umkehrschluss, dass auf den drei Testrouten auch ohne zusätzliche Annahmen über das Benutzerverhalten – d. h., dass diese anfangs der Route folgen – die Positionsverfolgung zuverlässig möglich ist. Die Ergebnisse zeigen, dass an Startpositionen mit wenigen Freiheitsgraden (z. B. Korridoren) die Bestimmung der Initialrichtung, z. B. anhand des Magnetkompasses, nicht zwingend nötig ist. Die Disambiguierung verschiedener möglicher Trajek-

torien kann dort durch die vom Umgebungsmodell auferlegten Beschränkungen erfolgen.

3.4.3.4 Fazit

Die zweite Studie diente in erster Linie zur Etablierung des realistischen Experimentaufbaus mit der Trennung von Testleiter- und Testpersonen-Gerät. Diese Methodik hat sich bewährt und wird später auch in der abschließenden Evaluationsstudie (Abschnitt 5.4.2) angewendet.

Aufgrund der neuen Testrouten und des veränderten Experimentaufbaus sind die erreichten Genauigkeitswerte nicht unmittelbar mit denen aus der ersten Studie vergleichbar, liegen jedoch wiederum in einer ähnlichen Größenordnung.

Die Ergebnisse zeigen, dass auch mit reinen infrastrukturfreien Verfahren eine stabile Positionierung im Navigationsgraph möglich ist – zumindest für die Länge der Testrouten. Um Stabilität auf längeren Strecken zu gewährleisten, sind zusätzliche Informationen nötig. Eine Möglichkeit dafür ist die in Kapitel 5 behandelte Erkennung von ortsgebundenen Aktivitäten. Zunächst widmet sich jedoch das folgende Kapitel dem Navigationsverhalten unter realen Bedingungen, um mehr Informationen über die Anforderungen an ein Indoor-Positionierungssystem zu gewinnen.

Kapitel 4

Positionierungsrelevantes Navigationsverhalten

In der Literaturübersicht im vorigen Kapitel wurden einige Arbeiten vorgestellt, die Indoor-Positionen mit hoher Genauigkeit bestimmen und verfolgen können. In den meisten Fällen wird dies in kontrollierten Experimenten nachgewiesen (vgl. Bousdar Ahmed et al., 2020) oder unter Wettbewerbsbedingungen evaluiert (z. B. Potorti et al., 2020, 2021; Torres-Sospedra et al., 2018). Diese Methoden eignen sich jedoch nicht dazu, tatsächliches Benutzerverhalten in einer realen Navigationssituation zu erforschen, was wiederum wichtig für das Verständnis der Anforderungen an ein Indoor-Positionierungssystem ist, welches in einer Endnutzer-Anwendung zum Einsatz kommen soll.

Aus diesem Grund geht dieses Kapitel der Frage nach, wie sich Personen während der Navigation mit einer Fußgängernavigations-App verhalten, mit speziellem Fokus auf denjenigen Aspekten, die die Indoor-Positionierung beeinflussen. Als Untersuchungsgegenstand dient die bereits in Abschnitt 2.4 vorgestellte Campus-Navigations-App URwalking für die Android-Plattform.

Methodisch findet an dieser Stelle ein Wechsel hin zu naturalistischen Logstudien statt. Wie Abschnitt 4.2.2 zeigen wird, sind derartige Studien üblicherweise auf grobe Nutzungsanalysen beschränkt, da Forschenden in der Regel kein Einblick in das Innenleben von Drittanbieter-Applikationen wie beispielsweise Google Maps gewährt wird. Da die Navigations-App in der vorliegenden Arbeit jedoch selbst entwickelt wurde und beliebig angepasst werden kann, sind derartige Analysen in diesem Fall möglich.

Im weiteren Verlauf dieses Kapitels werden zunächst die Leitfragen der naturalistischen Studie (Abschnitt 4.1) sowie vorhandene Literatur zur Smartphone-Nutzung im Allgemeinen und zum Navigationsverhalten im Speziellen (Abschnitt 4.2) vorgestellt. In Abschnitt 4.3 folgt die Beschreibung des mithilfe der Cam-

pus-Navigations-App erhobenen Datensatzes, inklusive der dabei unternommenen Vorverarbeitungsschritte. In der anschließenden Analyse wird die grundsätzliche Verwendung der App untersucht (Abschnitt 4.4), um daraufhin mithilfe der gewonnenen Erkenntnisse den Datensatz zu filtern und das für die Positionsbestimmung relevante Benutzerverhalten während tatsächlicher Navigationssitzungen zu studieren (Abschnitt 4.5).

4.1 Leitfragen der naturalistischen Studie

Das grundsätzliche Ziel der Studie ist – wie bereits beschrieben –, Informationen über das Fußgänger-Navigationsverhalten zu sammeln, die für die Umsetzung eines Indoor-Positionierungssystems von Relevanz sind. Auf diese Weise können die Szenarien und Anwendungsfälle besser eingeschätzt werden und informierte Entscheidungen über die Schwerpunkte während der Entwicklung und Abstimmung des Systems gefällt werden. Es folgen die Fragestellungen, an denen sich die Datenanalyse orientiert:

Wie sind Navigationssitzungen räumlich und zeitlich charakterisiert?

Die erste Frage betrifft die grundlegenden Charakteristiken von Navigationssitzungen. Insbesondere sind hier die Sitzungsdauer, d. h. die Zeit, die zwischen dem Öffnen und Schließen der App vergeht, von Interesse. Wissen um die durchschnittliche Sitzungsdauer ermöglicht es, die Anforderungen an ein Positionierungssystem, insbesondere dessen Fähigkeit zur längerfristigen Positionsverfolgung, besser einzuschätzen. Gleichmaßen ist in diesem Zusammenhang die Länge der geplanten Routen relevant. Weiterhin erlaubt die räumliche und zeitliche Verteilung der Sitzungen Rückschlüsse auf die Anwenderschaft der App.

Wie häufig und wie weit wird tatsächlich navigiert?

Navigation ist nach Montello (2005) eine Kombination aus *Wayfinding*, dem Planen der Route oder der nächsten Schritte, und *Locomotion*, der tatsächlichen Fortbewegung. Es stellt sich dabei die Frage, ob und wie häufig die App lediglich zur Planung verwendet wird; in diesen Fällen ist keine Positionierung nötig. In diesem Zusammenhang ist ebenfalls von Interesse, ob NutzerInnen im Falle tatsächlicher Navigationsvorgänge typischerweise mithilfe der App bis zum Ziel gelangen oder den Vorgang vorher abbrechen.

Wie häufig treten Pausen und Unterbrechungen auf?

Im Verlauf einzelner Navigationssitzungen sind – bedingt durch *Wayfinding*- und *Locomotion*-Phasen – Pausen zu erwarten, d. h. Zeiträume, in denen der Benutzer stehen bleibt und sich am selben Ort aufhält. In kontrollierten Indoor-Positionierungsstudien spielt dieses Phänomen jedoch nur selten eine Rolle (vgl. z. B. Xiao, Wen, Markham & Trigoni, 2015), im Gegenteil wird oft Wert auf möglichst konstante Bewegungsgeschwindigkeit während des kompletten Experiments gelegt (vgl. z. B. Wu, Chen, Gu, Zheng & Shang, 2019). Für ein PDR-basiertes Positionierungssystem ist insbesondere relevant, ob während derartiger Pausen fälschlicherweise Schritte erkannt werden, beispielsweise durch ruckartige Bewegungen des Smartphones. So zeigten beispielsweise L. Yang, Li und Rizos (2014), dass vermehrte Geh-Pausen die Genauigkeit der Schritterkennung reduzieren.

Ebenso wie die kurzen Geh-Pausen, welche durch *Wayfinding* erklärt werden können, sind auch Unterbrechungen von Interesse, in denen die Navigations-App nicht mehr aktiv benutzt wird, sondern nach dem Start eines Navigationsvorgangs im Hintergrund weiterläuft. Dies ist der Fall, wenn das Gerät entweder anderweitig verwendet oder aber vorübergehend in den Standby-Modus geschickt wird. Im Hinblick auf die Indoor-Positionierung stellt sich in diesem Zusammenhang die Frage, ob während derartiger Unterbrechungen die Position weiter verfolgt werden soll, damit beim erneuten Öffnen der App eine möglichst genaue Positionsschätzung vorliegt, oder aber, ob – auch vor dem Hintergrund von Energieeffizienz und Akkulaufzeit – die Positionierung pausiert werden sollte.

Die Arten und Ursachen der Unterbrechungen können mit dem erhobenen Datensatz nicht ergründet werden, da sich dieser auf die Interaktionen mit der URwalking-App beschränkt. Die Analyse der Logdaten wird daher hauptsächlich der Frage nachgehen, wie häufig und wie lange Pausen und Unterbrechungen auftreten.

Wie wird das Gerät getragen?

Die Art und Weise, wie das Gerät getragen wird, hat Einfluss auf die Fähigkeit eines Positionierungssystems, die Nutzerposition zu verfolgen (Brajdic & Harle, 2013). In vielen Studien zur Indoor-Positionierung bleibt das Gerät konstant vor dem Körper, auch um während des Experiments Annotationen von Ground-Truth-Positionen vornehmen zu können (vgl. z. B. Torres-Sospedra et al., 2017). Diese Annahme erlaubt es, die Gehrichtung unmittelbar aus der Ausrichtung des Geräts abzuleiten. In der Studie soll daher untersucht werden, ob diese Trageart in der Tat die vorherrschende ist und wie oft sich das Gerät in anderen Positionen und Ausrichtungen befindet.

4.2 Literaturübersicht

Im Folgenden wird zunächst anhand der Literatur und ausgewählter naturalistischer Studien zum Smartphone-Nutzungsverhalten die in diesem Kapitel verwendete Methodik motiviert. Anschließend folgt ein Überblick über Erkenntnisse zum Fußgänger-Navigationsverhalten, insbesondere im Hinblick auf die für die Positionsbestimmung relevanten Aspekte.

4.2.1 Naturalistische Studien und Logdatenanalyse

Naturalistische Log-Studien zielen darauf ab, das Verhalten von NutzerInnen unter realistischen Bedingungen anstelle im Rahmen eines kontrollierten Experiments zu erforschen. Hierfür werden vom Untersuchungsgegenstand – im vorliegenden Fall die Fußgängernavigations-App – Benutzeraktionen und weitere relevante Daten erhoben, verarbeitet und analysiert. Wie Dumais et al. (2014, S. 351 f.) darlegen, bieten derartige Studien mehrere Vor- und Nachteile gegenüber Labor- oder Feldstudien: Positiv ist, dass die TeilnehmerInnen das zu untersuchende System wie gewohnt und ohne direkte Beobachtung oder Eingriff durch einen Experimentleiter nutzen können. Da die Datenerhebung durch ein technisches System realisiert ist, können in kurzer Zeit große Mengen an Daten gesammelt werden, wodurch auch geringe Effekte, beispielsweise beim A/B-Test zweier Systeme, statistisch messbar werden. Im Gegensatz zu Interviewstudien erfolgt die Datenerhebung objektiv und ist nicht auf das Gedächtnis oder das aktive Mitwirken der TeilnehmerInnen angewiesen. Jansen, Taksa und Spink (2009, S. 10) nennen als zusätzliche Vorteile die Ortsunabhängigkeit und die potenziell unbegrenzte Dauer der Studie, da keine explizite Rekrutierung von Teilnehmern nötig ist.

Hauptnachteil von Log-Studien ist nach Dumais et al. (2014, S. 352) die fehlende Möglichkeit, Informationen über die Motivation, die Ziele oder die Zufriedenheit der BenutzerInnen zu erfassen. Auch der Nutzungskontext ist in der Regel nur unvollständig erhebbar. Jansen et al. (2009, S. 10 f.) weisen darauf hin, dass auch in naturalistischen Studien das Nutzerverhalten durch die bloße Tatsache der Datenerfassung (bzw. das Wissen darum) verändert wird. Weitere von Jansen et al. genannte Nachteile sind die begrenzte Aussagekraft von Logdaten hinsichtlich nachträglich formulierter Forschungsfragen oder die oft schwache Verbindung zwischen Logdaten und den zu messenden Konzepten. Dies kommt jedoch vor allem beim Arbeiten mit existierenden Datensätzen zum Tragen, während in der vorliegenden Arbeit die Datenerhebung unmittelbar auf die eingangs genannten Fragestellungen abgestimmt ist.

Dumais et al. (2014, S. 369 ff.) sprechen auch die ethischen Fragestellungen an,

die bei der Konzeption und Durchführung von naturalistischen Log-Studien beachtet werden müssen. Trotz Anonymisierung oder Pseudonymisierung können detaillierte Logdaten unter Umständen Rückschlüsse auf einzelne Personen zulassen, als Beispiel wird u. a. der bekannt gewordene Fall eines Suchmaschinen-Datensatzes ins Feld geführt. Kiukkonen, Blom, Dousse, Gatica-Perez und Laurila (2010) beispielsweise treffen bei der *Lausanne Data Collection Campaign* – einer groß angelegten Studie, bei der fast 170 Personen über einen Zeitraum von einem Jahr ihre Smartphone-Nutzungsdaten bereitstellten – einige Vorkehrungen zum Schutz der Privatsphäre der TeilnehmerInnen. Darunter fallen u. a. das Unkenntlichmachen eindeutig identifizierbarer Daten wie Namen und Telefonnummern sowie eine reduzierte GPS-Genauigkeit an sensiblen Orten. Kiukkonen et al. führen weiterhin aus, dass die Problematik allein durch technische Maßnahmen nicht zu lösen ist und dass letztendlich eine Abwägung zwischen dem Schutz der Privatsphäre und der Möglichkeit der präzisen Beantwortung von Forschungsfragen nötig ist. Einige der beschriebenen Maßnahmen kommen auch in der vorliegenden Arbeit zum Einsatz (siehe Abschnitt 4.3).

Naturalistische Methoden wurden bereits in vielen Studien zur Untersuchung des Smartphone-Nutzungsverhaltens angewendet, wobei die Erfassung und Analyse der Nutzungsdaten meist zwangsläufig auf einer hohen Abstraktionsebene stattfindet. Zwar können Nutzungszeiten und sonstige Kontextfaktoren wie z. B. der Standort untersucht werden, nicht aber das Verhalten innerhalb einzelner Apps.

Aufbauend auf dem Datensatz von Kiukkonen et al. wurden u. a. die automatisierte Erkennung von besuchten Orten (Do & Gatica-Perez, 2013b; Malmi, Do & Gatica-Perez, 2012; Montoliu, Blom & Gatica-Perez, 2013), die Modellierung von menschlicher Mobilität (Do, Dousse, Miettinen & Gatica-Perez, 2015; Do & Gatica-Perez, 2014) sowie durch Annäherungsdaten erkennbare Interaktionsmuster (Do & Gatica-Perez, 2013a) erforscht.

Mehrere Studien liefern Erkenntnisse zur typischen Sitzungsdauer von Smartphone-Apps. Bei Böhmer, Hecht, Schöning, Krüger und Bauer (2011) wird diese als 71,6 s angegeben; an der Studie nahmen mehr als 4000 Android-BenutzerInnen über einen Zeitraum von etwa 5 Monaten teil. Carrascal und Church (2015) errechnen eine mittlere Sitzungsdauer von 90 s, welche sehr linkssteil verteilt ist ($Mdn = 17$ s). In der Studie von Brown, McGregor und McMillan (2014), bei der in einem Zeitraum von 100 Tagen von 15 TeilnehmerInnen Bildschirmaufzeichnungen mit einer Gesamtlänge von über 70 h gesammelt wurden, beträgt die Sitzungsdauer im Median 38 s. Die Autoren identifizierten vor allem sog. *Checking Behaviour*, d. h. das Überprüfen auf neue Nachrichten oder Social-Media-Updates, als Erklärung für die kurzen Sitzungen. Ein weiterer Beleg für kurze App-Sitzungen findet sich z. B. bei Yan, Chu, Ganesan, Kansal und Liu (2012, S. 114), wonach 80 % der Apps

weniger als zwei Minuten lang benutzt werden.

Die bisher beschriebenen Studien betreffen das Smartphone-Nutzungsverhalten im Allgemeinen und beschränken sich, wie eingangs erwähnt, aus technischen Gründen meist auf grob messbare Charakteristiken wie die Sitzungslänge. Im folgenden Abschnitt werden Studien vorgestellt, die sich speziell mit dem Fußgänger-Navigationsverhalten befassen.

4.2.2 Studien zum menschlichen Navigations- und Bewegungsverhalten

Forschung zu menschlichem Navigations- und Bewegungsverhalten betrifft u. a. Aspekte der Psychologie und Physiologie. Eine erschöpfende Literaturübersicht geht weit über die in der vorliegenden Arbeit verfolgten Ziele hinaus, weswegen der Fokus im Folgenden auf den für die Positionsbestimmung relevanten Aspekten liegt, welche eingangs als Leitfragen formuliert wurden.

Einige der im vorigen Abschnitt genannten Arbeiten schlüsseln die App-Nutzung nach Kategorien auf. Bei Böhmer et al. (2011) sind Sitzungen in der Kategorie *Travel*, in der u. a. Navigations-Apps wie Google Maps und Waze einsortiert sind, mit 44,7 s kürzer als der Durchschnitt (71,6 s), während die Kategorie *Travel & Local* bei Carrascal und Church (2015) mit durchschnittlich 111,6 s eine etwas längere Sitzungsdauer aufweist. Google-Maps-Sitzungen sind bei Savino et al. (2020) im Mittel 65 s lang ($Mdn = 44$ s). Es zeigt sich also, dass je nach Studie die Nutzungsdauer stark schwankt, wobei sicherlich auch die Art der Einordnung bzw. Klassifizierung von Navigations-Apps eine Rolle spielt.

Ebenfalls stark abhängig von der konkret untersuchten Umgebung ist die Routenlänge. Für den Regensburger Campus wurden von Ullmann (2020, S. 111 f.) in einer Nutzerbefragung 221 Routen erhoben, die typische Wege auf dem Campus repräsentieren. Die Pfade sind im Durchschnitt 352 m lang ($SD = 115$ m), mit einer Spannweite von 108 bis 805 m. Vergleichsweise vom Routing-Algorithmus berechnete kürzeste Pfade zwischen den angegebenen Start- und Endpunkten haben eine durchschnittliche Länge von 304 m ($SD = 107$ m).

Aus physiologischen Studien sind umfangreiche Daten zu menschlichem Gehverhalten bekannt. Bohannon und Williams Andrews (2011) tragen in einer Meta-studie Erkenntnisse über die typische Geh-Geschwindigkeit für verschiedene Altersgruppen zusammen. Bei gesunden Männern liegt diese im Mittel zwischen 96,8 und 143,4 cm/s, bei Frauen zwischen 94,3 und 139,0 cm/s. Dies deckt sich größtenteils mit den Beobachtungen von Öberg et al. (1993), welche zusätzlich Informationen zu Schrittlänge und -frequenz liefern. Bei normaler Gangart – d. h. weder explizit schnell noch langsam gehend – bewegt sich die Schrittfrequenz bei

Männern je nach Alter zwischen 1,91 und 2,14 Schritten/s, bei Frauen zwischen 1,97 und 2,16 Schritten/s. Die Schrittlänge, ebenfalls berichtet von Öberg et al. (1993, S. 222), beträgt zwischen 61,5 und 66,0 cm (Männer) bzw. 53,5 und 59,7 cm (Frauen). Bei schneller Gangart steigen diese Werte auf über 70 cm, bei langsamem Gehen fallen sie teilweise auf unter 50 cm.

Die Thematik von Pausen während der Navigation wird beispielsweise von Robinson, Jones, Eslambolchilar, Murray-Smith und Lindborg (2010) erwähnt, welche in ihrem Navigationsexperiment auf einem Universitätscampus gelegentlich kurze stationäre Phasen beobachten, vor allem an größeren Kreuzungen. Ansonsten war die Geschwindigkeit während des Experiments weitgehend konstant. Rehrl, Häusler, Leitinger und Bell (2014) zeichneten in einer In-Situ-Studie (N=48) im Außenbereich u. a. die Dauer und die Ursachen von Geh-Pausen auf. Einige der gewonnenen Informationen – beispielsweise der Einfluss des Straßenverkehrs – sind nicht unmittelbar auf die in dieser Arbeit betrachtete Umgebung übertragbar. Erwähnenswert ist jedoch, dass die Dauer und Anzahl der Pausen stark von der verwendeten Assistenzstrategie abhängt. Insbesondere im Fall der in der Studie untersuchten Augmented-Reality-Lösung führen Sensor-Ungenauigkeiten und Orientierungsprobleme oft zu Pausen, während diese bei Benutzung digitaler Karten weit weniger häufig auftreten.

Zum aktuellen Zeitpunkt sind nur wenige naturalistische Untersuchungen mit Fokus auf smartphonebasierter (Indoor-)Fußgängernavigation bekannt, u. a. aufgrund eines Mangels an verfügbaren Datensätzen. Als Gründe dafür führen Savino et al. (2020) in ihrer Studie zur Verwendung von mobilen Kartenanwendungen Bedenken hinsichtlich der Privatsphäre sowie – im Falle kommerzieller Produkte – der wirtschaftlichen Konkurrenz an. Die Autoren arbeiten mithilfe einer Wrapper-Applikation um die Google-Maps-Webseite, welche es ermöglicht, das Verhalten innerhalb der (Web-)App zu untersuchen. So wird beispielsweise als häufiges Muster die Abfolge von Kartenmanipulation, Suche, Auswahl eines Ortes und anschließender Abfrage von Navigationsanweisungen identifiziert. Eine weitere naturalistische Studie wurde von Ruggiero, Charitha, Xiang und Lucia (2018) durchgeführt, welche mithilfe von Kameraaufnahmen die verschiedenen Wege untersuchten, die Personen in einem Foyer im Innenbereich zurücklegten. Aus den fast 300 000 erhobenen Trajektorien wurden anschließend mathematische Modelle für die Bewegung in derartigen Freiflächen abgeleitet, aus denen hervorgeht, dass FußgängerInnen meist leicht gebogene Pfade beschreiten. Aufgrund der fehlenden Experimentsituation bescheinigen die Autoren der Studie eine hohe ökologische Validität; Nachteil ist jedoch die Begrenzung auf eine einzelne, vergleichsweise kleine Fläche.

Zusammenfassend lässt sich konstatieren, dass den für die Indoor-Positionie-

rung relevanten, eingangs in Form der Leitfragen formulierten Aspekten menschlichen Navigationsverhaltens bisher in der Literatur wenig Aufmerksamkeit zuteil wurde. Im weiteren Verlauf des Kapitels wird die aus diesem Grund durchgeführte naturalistische Logstudie beschrieben, beginnend im nächsten Abschnitt mit einer Übersicht über den verwendeten Datensatz.

4.3 Der Navigations-Log-Datensatz

Der naturalistische Navigations-Log-Datensatz wurde mithilfe der für die Android-Plattform verfügbaren Campus-Navigations-App URwalking (siehe Abschnitt 2.4) erhoben. Die aktive Nutzerakquise für URwalking im Allgemeinen erfolgte hauptsächlich in Form von auf dem Campus verteilten Schildern (mit QR-Codes zur Positionsbestimmung) sowie durch Verweise auf der Universitäts-Homepage. Die URwalking-Android-App ist durch Verlinkung von der URwalking-Webseite sowie im Google-Play-Store als eines der ersten Ergebnisse unter dem Suchbegriff »Uni Regensburg« verfügbar. Auf weitere, explizite Werbemaßnahmen wurde verzichtet. Die Auswertung der Statistik im Google-Play-Store zeigt, dass mehr als 90 % der Installationen auf die dortige Suchfunktion zurückzuführen sind, und nur wenige die App per externem Link, z. B. via URwalking-Webseite, finden. Dies lässt die Vermutung zu, dass die Bekanntheit der App vor allem auf Mundpropaganda zurückzuführen ist.

4.3.1 Erhebung der Daten

BenutzerInnen der URwalking-App werden zu Beginn der ersten Navigationssitzung gefragt, ob sie ihre Nutzungsdaten anonymisiert für wissenschaftliche Zwecke zur Verfügung stellen möchten. Die Zustimmung kann jederzeit in den Einstellungen der App widerrufen werden. Es werden keine personenbezogenen Daten wie Name, Alter oder Geschlecht erhoben.

Eine Sitzung beginnt mit dem Start der Navigation oder dem Öffnen der Karte und läuft solange, bis die App explizit geschlossen wird. Nach Beendigung einer Sitzung werden die dazugehörigen Daten zu ZIP-Archiven komprimiert und im Hintergrund an den URwalking-Server gesendet. Erfolgreich hochgeladene Archive werden anschließend vom Gerät gelöscht. Wurde in die Datenübertragung eingewilligt, so werden während einer Sitzung folgende Daten in jeweils einzelnen CSV-Dateien aufgezeichnet:

Positionsdaten

Die erhobenen Positionsdaten bestehen einerseits aus dem vom Android-System über den Fused-Location-Provider bereitgestellten Standort. Die Daten umfassen einen Zeitstempel, Längen- und Breitengrad (WGS84-Koordinaten), die Höhe über Normalnull in Metern sowie die geschätzte Genauigkeit der Position (siehe Listing 4.1). Weiterhin wird die vom im vorigen Kapitel beschriebenen Indoor-Positionierungssystem berechnete Position aufgezeichnet. Diese wird ebenfalls in Form von GPS-Koordinaten sowie zusätzlich im intern verwendeten Gauß-Krüger-Format samt Stockwerks-ID und geschätzter Genauigkeit abgespeichert.

```
timestamp;lat;lon;alt;acc
1615018090000;49.000858;12.093956;419.660006;7.962000
1615018471954;49.000120;12.094051;419.760012;27.125000
1615018476002;49.000828;12.093875;419.760012;17.041000
1615018477304;49.000836;12.093905;419.760012;14.225000
1615018479000;49.000804;12.093952;419.760012;13.104000
...
```

Listing 4.1: Gekürztes Beispiel einer Positions-Logdatei

Sensordaten

Neben den Positionsdaten werden Messwerte aller für weitere Analysen relevanten Sensoren erhoben. Um nachträglich die Positionsbestimmung zu verifizieren und ggf. mit aktualisierten Verfahren zu verfeinern, werden die dafür nötigen Daten von Schrittkenner und GRV erfasst. Weiterhin werden die für spätere HAR-Analysen verwendeten Daten von Gyroskop, Accelerometer, Magnetfeldsensor und Barometer gespeichert.

Die Daten aller Sensoren werden jeweils in Form der in Abschnitt 2.3.2 beschriebenen Dimensionen und – sofern vorhanden – mitsamt der gemeldeten Genauigkeit gespeichert. Jedes Sensor-Event enthält einen Zeitstempel, welcher jedoch geräte- und versionsabhängig unterschiedliche Bedeutung hat: Teils werden damit Nanosekunden seit Start des Geräts, teils seit Beginn der Unix-Zeit am 01.01.1970 kodiert; in wieder anderen Fällen wird eine fortlaufende Ganzzahl hochgezählt. Aus diesem Grund wird neben dem vom Sensor gemeldeten Zeitstempel auch die aktuelle Systemzeit als Unix-Zeitstempel in Millisekunden erfasst, wodurch der zuverlässige Abgleich von Daten unterschiedlicher Sensoren sowie von Interaktionsdaten gewährleistet ist. Ein Beispiel für eine Logdatei des Accelerometers ist in Listing 4.2 zu sehen.

```
timestamp;x;y;z;accuracy;sensorTimestamp
1615018468004;0.37589;8.95192;4.26407;3;1054904272635797
1615018468012;0.30167;8.93037;4.36224;3;1054904280631402
1615018468020;0.20590;8.87531;4.36224;3;1054904288565973
1615018468028;0.08619;8.85615;4.39336;3;1054904296531061
1615018468036;0.02634;8.82024;4.40533;3;1054904304496149
...
```

Listing 4.2: Gekürztes Beispiel einer Logdatei des Beschleunigungssensors

Interaktionsdaten

Die erhobenen Interaktionsdaten umfassen grundlegende Interaktionen mit dem User-Interface, z.B. das Weiterklicken der Routenanweisungen oder das Zoomen und Verschieben der Karte, welche in dieser Studie jedoch nicht ausgewertet werden. Zu den Interaktionsdaten zählen ebenso Informationen über das Öffnen und Schließen bzw. Pausieren der App, welche für die Bestimmung der Sitzungslänge und der Unterbrechungen herangezogen werden (siehe Listing 4.3). Technisch umgesetzt ist das Aufzeichnen des Öffnens und Schließens der App durch Überschreiben der `onResume()`- und `onPause()`-Funktionen in der Navigations-*Activity*. Auch das Sperren des Bildschirms wird intern äquivalent zum Pausieren der App gehandhabt.

```
timestamp;type
1615018465378;INTERACT_RESUME_APP
1615018466880;INTERACT_START_NAVIGATION
1615018471949;INTERACT_SCROLL
...
1615018478711;INTERACT_ZOOM
1615018480771;INTERACT_PAUSE_APP
1615018543906;INTERACT_RESUME_APP
...
```

Listing 4.3: Gekürztes Beispiel einer Interaktions-Logdatei

Metadaten

Neben den Positions-, Sensor- und Interaktionsdaten werden einige weitere Metadaten erfasst. Beim ersten Start der App wird eine eindeutige installationsspezifische ID generiert, anhand derer die Logdaten ein- und desselben Geräts identifiziert werden können. Somit ist nach einem Gerätewechsel oder einer Neu-Installation der App keine Verfolgung des Benutzers möglich. Im Folgenden wird angenommen, dass BenutzerInnen nur sehr selten das Gerät wechseln oder die

App neu installieren, und dass Geräte nicht von mehreren Personen im Wechsel verwendet werden, sodass die erzeugte ID als Proxy für eine einzelne Person dient. Als eindeutige Sitzungs-ID wird in Kombination mit der Installations-ID der Unix-Zeitstempel des Sitzungsbeginns verwendet. Weiterhin werden etwaige Sucheingaben sowie die geplante Route erfasst, Letztere im vom Pfadplanungs-Server gelieferten JSON-Format. Schließlich werden für statistische Zwecke die Android-Version, Gerätehersteller und -typ, sowie Informationen über verbaute Sensoren erhoben.

4.3.2 Vorverarbeitung

Vor der Analyse der naturalistischen Logdaten waren einige Vorverarbeitungsschritte und Maßnahmen zur Säuberung des Datensatzes notwendig.

Zunächst wurden durch Abgleich der Versionsnummer der App – erkennbar an der in Nicht-Release-Versionen enthaltenen Revision des Versionskontrollsystems – alle Sitzungen ausgeschlossen, die beim Entwickeln und Testen entstanden und hochgeladen worden waren. Während des Untersuchungszeitraums wurde der Logging-Mechanismus auch für kontrollierte Experimente im Rahmen von Abschlussarbeiten verwendet. Die betroffenen Sitzungen waren durch spezielle Tags gekennzeichnet und wurden vom Datensatz entfernt.

Nach diesen Filtervorgängen verbleiben 13541 Sitzungen von 1677 BenutzerInnen, wovon 1845 Sitzungen auf den reinen Kartenmodus entfallen, bei dem ohne Auswahl einer Route Campus- und Gebäudepläne betrachtet werden können. Da sich die nachfolgenden Analysen auf tatsächliche Navigationssituationen konzentrieren, wurden diese Sitzungen ebenfalls nicht weiter berücksichtigt.

Die restlichen Sitzungen wurden zum Zwecke performanterer weiterer Analysen in eine PostgreSQL-Datenbank importiert. Je Sensor- bzw. Datentyp wurde eine Tabelle angelegt, in welche die Inhalte aller korrespondierenden CSV-Dateien eingelesen wurden. Bei dieser Gelegenheit fanden weitere Vorverarbeitungsschritte statt: Koordinaten, die sich außerhalb des Campus befinden, wurden aus Datenschutzgründen auf zwei Nachkommastellen, d. h. auf Größenordnung von ca. 1 km, gerundet. Aus den als Quaternion vorliegenden Orientierungsangaben des GRV wurden unter Anwendung folgender Formeln einfacher interpretierbare Richtungsvektoren berechnet (Shepperd, 1978, S. 224):

$$\text{GRV}_x = 2(q_x q_y - q_w q_z) \quad (4.1)$$

$$\text{GRV}_y = q_w^2 - q_x^2 + q_y^2 - q_z^2 \quad (4.2)$$

$$\text{GRV}_z = 2(q_y q_z + q_w q_x) \quad (4.3)$$

Nach der Filterung des Datensatzes verbleiben 11 696 Sitzungen von 1603 NutzerInnen aus einem Zeitraum von ca. 30 Monaten (August 2018 – Februar 2021). Der zeitliche Umfang beträgt über 700 h, aneinandergereiht sind alle abgerufenen Routen 3183 km lang.

Nicht in allen Sitzungen sind Daten aller Sensoren vorhanden. Vor allem in sehr kurzen Sitzungen ist beispielsweise oft keine GPS-Position bekannt, in anderen Fällen sind Sensoren schlichtweg nicht im Gerät verbaut. Die folgenden Analysen beschränken sich daher – falls nötig – auf die Untermenge des Datensatzes, in denen auf den spezifischen Sensor tatsächlich zugegriffen werden konnte.

4.4 Verwendung der Navigations-App URwalking

Vor der Analyse des Navigationsverhaltens ist zunächst von Interesse, wie die App im Allgemeinen verwendet wird. Die Fragestellungen an den Datensatz zielen darauf ab, möglichst viele für die Bereitstellung optimaler Assistenz relevante Informationen zu extrahieren. Die meisten Analysen weisen daher einen unmittelbaren Bezug zu Design-Entscheidungen des Indoor-Positionierungssystems auf.

Im Einzelnen werden in den folgenden Abschnitten räumliche und zeitliche Muster sowie Charakteristika von Navigationssitzungen untersucht. Ein Überblick über die verwendeten Geräte und die dort verbauten Sensoren rundet die Analyse ab.

4.4.1 Räumliche und zeitliche Muster

Um die Frage zu beantworten, wann und wo die App verwendet wird, werden zunächst grundlegende Nutzungsstatistiken ausgewertet. Dies umfasst die zeitliche Verteilung von Navigationssitzungen im Jahres- und Wochenverlauf sowie die Charakteristiken der geplanten Routen.

Nutzung im Zeitverlauf

Der Datensatz erstreckt sich, wie oben bereits erwähnt, über mehr als zwei Jahre. Die Nutzungsintensität innerhalb dieses Zeitraums variiert deutlich: Jeweils zu Semesterbeginn ist ein markanter Anstieg der Navigationssitzungen zu verzeichnen, nach dem sich das Niveau in den darauf folgenden Wochen wieder verringert. In den vorlesungsfreien Zeiten ist das Aufkommen erwartungsgemäß am geringsten.

Abbildung 4.1 zeigt – man beachte die logarithmische y-Achse – die Anzahl der wöchentlichen Sitzungen im Verlauf der Studie, mit sichtbaren Spitzen zu Beginn

jedes Semesters im Herbst und Frühjahr. Eine Ausnahme bildet das Sommersemester 2020, bei dem anhand ausbleibender NutzerInnen die Auswirkungen der Covid-19-Pandemie gut zu erkennen sind.

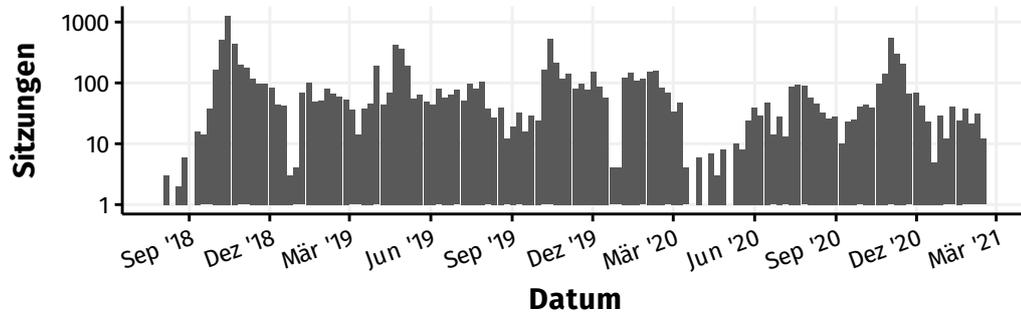


Abbildung 4.1: Wöchentliche Sitzungen im Verlauf der Studie

Die Nutzungsmuster entsprechen den vorausgehenden Annahmen bezüglich der Zielgruppe der App. Studierende oder Angestellte an der Universität stehen vor allem zu Semesterbeginn vor der Problematik, bisher unbekannte Räumlichkeiten auffinden zu müssen. Über den gesamten Zeitraum bis zur pandemiebedingten Einstellung der Präsenzveranstaltungen im März 2020 hinweg ist der Nutzungstrend relativ stabil, was sich durch die begrenzte Zielgruppe und den Verzicht auf besondere Werbemaßnahmen erklären lässt.

Auch die Analyse der Nutzungszeiten im Tages- und Wochenverlauf (Abbildung 4.2) führt zu einem erwarteten Bild. Ein Großteil der Sitzungen (78,1%) findet wochentags zwischen 8 und 18 Uhr statt und fällt somit auf die Hauptveranstaltungszeiten der Universität.

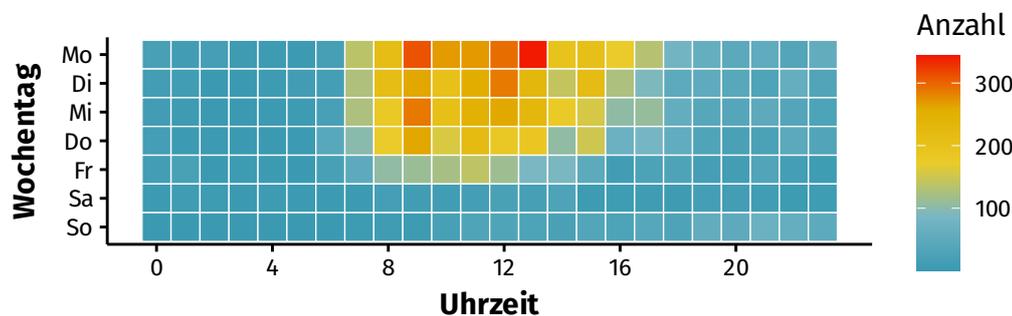
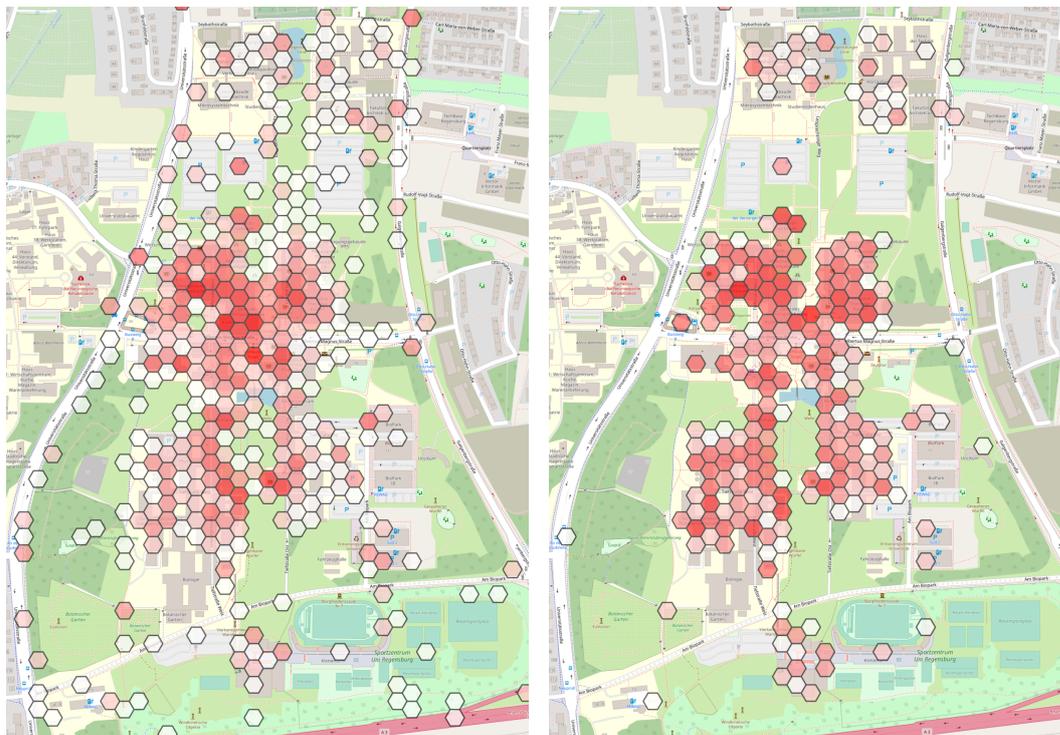


Abbildung 4.2: Verteilung der Sitzungen im Tages- und Wochenverlauf

Start- und Zielorte

Neben den zeitlichen sind auch räumliche Muster zu erkennen. Sowohl Start- als auch Zielorte der abgefragten Routen sind über nahezu den kompletten Campus verteilt (Abbildung 4.3). Die Startpunkte liegen dabei auch in den Randbereichen, Wegen und Grünflächen, welche nicht als Zielpunkte in Frage kommen.



(a) Startpunkte

(b) Zielpunkte

Abbildung 4.3: Verteilung der Start- und Zielpunkte der Routen auf dem Campus. Kartenmaterial von OpenStreetMap (Haklay & Weber, 2008).

Gebäude, Bereiche und Stockwerke

Eine Charakteristik der Navigation auf einem Campus wie an der Universität Regensburg ist der Wechsel zwischen verschiedenen Gebäuden. Es ist zu vermuten, dass gerade dann erhöhter Bedarf nach Navigationsassistenz herrscht, wenn das Ziel in fremden oder selten besuchten Gebäuden liegt. Für die Positionierung relevant sind insbesondere die Übergänge zwischen Innen- und Außenbereichen (oder umgekehrt): Im Außenbereich kann die Genauigkeit der Positionierung mithilfe von satellitengestützten Systemen verbessert werden, um etwaige im Innenbereich angehäufte Fehler zu reduzieren. Aufgrund der üblicherweise überschaubaren Anzahl von Gebäudeein- und -ausgängen stellen diese außerdem eine gute Möglich-

keit dar, die Positionsschätzung auf eine eindeutige Position zu korrigieren. Dies ist insbesondere dann möglich, wenn der Türdurchgang mittels Aktivitätserkennung detektiert wird. Ähnliches gilt für das Wechseln von Stockwerken (siehe für beide Aspekte Kapitel 5).

Um die Wichtigkeit derartiger Komponenten einzuschätzen, wurde der Verlauf der geplanten Routen analysiert. Jedes Gebäude, das in einer separaten Datei modelliert wurde, gilt dabei als ein eigener Bereich, potenziell bestehend aus mehreren Stockwerken. Das komplette Außengelände des Campus bildet ebenfalls einen separaten Bereich. Die Analyse zeigt, dass sich Routen im Median über drei Bereiche erstrecken, der Maximalwert liegt bei neun. 18,2 % der Routen verlaufen innerhalb eines einzigen Bereichs, während 22,5 % zwei Bereiche miteinander verbinden. Am häufigsten führen Routen durch drei Bereiche (30,3 %), aber auch vier (16,2 %) oder gar fünf und mehr (12,6 %) Bereiche sind keine Seltenheit (siehe Abbildung 4.4).

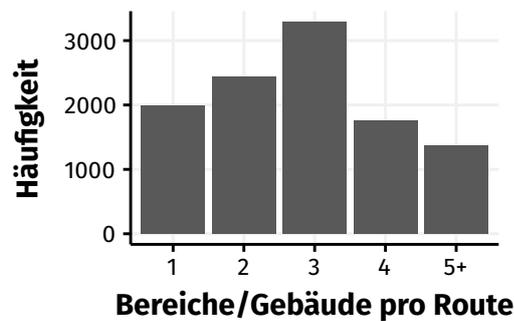
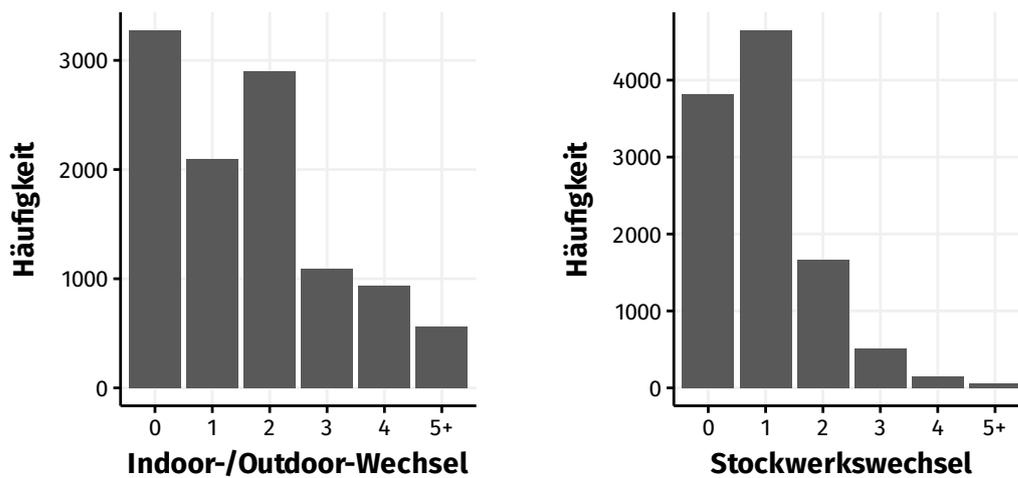


Abbildung 4.4: Anzahl unterschiedlicher Bereiche/Gebäude pro Route

Jede Route enthält im Median zwei Wechsel zwischen Indoor- und Outdoorbereichen. Dies entspricht dem häufigen Fall, bei dem die Route von einem Gebäude(-komplex) über eine Außenfläche zu einem anderen Gebäude führt. In 30,2 % aller Routen ist kein derartiger Übergang nötig, der größte Teil davon entfällt auf reine Indoor-Routen, die sich über ein oder mehrere baulich verbundene Gebäude erstrecken. Daraus folgt auch, dass in etwa 7 von 10 Routen ein Outdoor-Anteil sowie mindestens ein Indoor-/Outdoor-Wechsel enthalten sind (siehe Abbildung 4.5a).

Die meisten Routen kommen mit wenigen Stockwerksübergängen aus. In vielen Fällen ist keiner (35,2 %) oder nur ein einziger (42,9 %) enthalten (siehe Abbildung 4.5b).

Im Mittel enthält jede Route 7,3 Durchgangstüren ($SD = 4,6$; $Mdn = 7$; $Max = 27$), was dem Passieren einer Tür ca. alle 53 m entspricht. Deutlich seltener sind im Vergleich dazu Indoor-/Outdoor-Wechsel (alle 167 m) und Stockwerksübergänge



(a) Anzahl der Wechsel zwischen Indoor- und Outdoorbereichen pro Route

(b) Anzahl der Stockwerks-Wechsel pro Route

Abbildung 4.5: Indoor-/Outdoor-Wechsel und Stockwerkswechsel pro Route

(alle 306 m). Für das Positionierungssystem bedeutet dies, dass Türdurchgänge, sofern sie zuverlässig erkannt werden können, eine gute Möglichkeit sind, die Routen in kürzere, mit dem PDR-System leichter zu bewältigende Abschnitte aufzuteilen.

4.4.2 Charakteristika von Navigationssitzungen

Nach dem Überblick über den gesamten Datensatz folgt nun eine Analyse der Charakteristika von Navigationssitzungen, insbesondere deren Dauer und der Länge der geplanten Route, sowie der zurückgelegten Strecke. Mithilfe dieser Informationen können die Anforderungen an das Positionierungssystem besser eingeschätzt werden.

Sitzungsdauer

Das Messen der genauen Sitzungsdauer erwies sich als unerwartet schwierig. Ein Grund dafür ist die Tatsache, dass die App während des Navigationsvorgangs im Hintergrund weiterläuft, falls der Benutzer das Gerät nebenbei anderweitig verwendet. Sofern die App nicht explizit geschlossen wird, beendet sie das Android-Betriebssystem nach einer Weile selbst, wobei die Kriterien dafür je nach Gerätehersteller und Android-Version variieren. In solchen Fällen besteht oft keine Möglichkeit, das Ende der Sitzung explizit aus den Logdaten abzulesen. Dazu kommen in Einzelfällen unplausible Zeitstempel, welche möglicherweise durch das Umstel-

len der lokalen Uhrzeit oder das Wechseln der Zeitzone verursacht wurden.

Für die Analysen der Sitzungsdauer wurden daher nur solche Sitzungen herangezogen, für die entweder explizite Anfangs- und Endmarkierungen in Form von Interaktions-Logdaten vorhanden oder bei denen anhand von durchgehend aufgezeichneten Sensordaten Rückschlüsse auf Anfang und Ende möglich sind. Abbildung 4.6 zeigt die Verteilung der Dauer der verbliebenen 7917 Sitzungen. Mehr als 50 % der Sitzungen sind weniger als eine Minute lang, der Median liegt bei 55 s. Mit steigender Sitzungsdauer nimmt die Anzahl der Sitzungen ab, es ergibt sich also eine klassische Long-Tail-Verteilung.

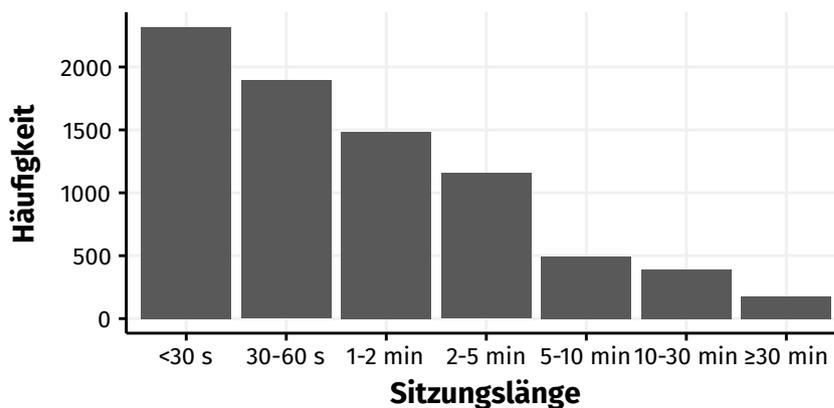
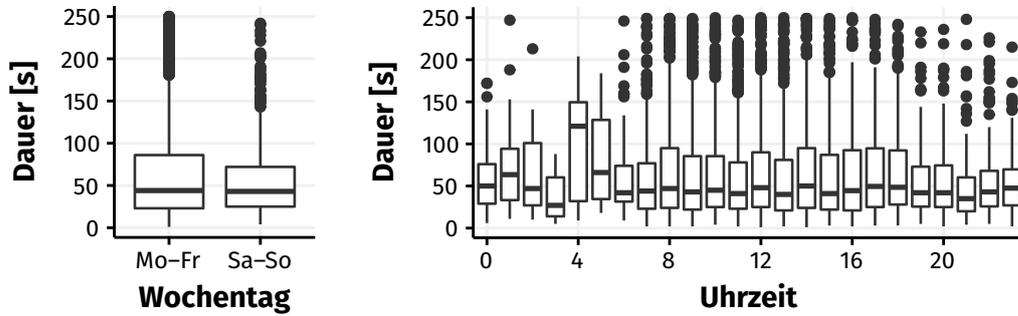


Abbildung 4.6: Verteilung der Sitzungslängen

Die typische Sitzungsdauer verändert sich wochentags- und tageszeitabhängig. Sitzungen am Wochenende sind im Durchschnitt kürzer ($Mdn = 47$ s) als unter der Woche ($Mdn = 56$ s). Ein durchgeführter Wilcoxon-Rangsummentest weist bei geringer Effektstärke ($r = 0,05$) auf einen signifikanten Unterschied hin ($W = 2285924$, $p < 0,01$). Analog dazu sind Sitzungen tagsüber während der Vorlesungszeiten länger ($Mdn = 58$ s) als abends oder nachts ($Mdn = 47$ s). Auch dieser Unterschied ist bei geringer Effektstärke ($r = 0,08$) signifikant ($W = 4285950$, $p < 0,01$). Beide Beobachtungen sind in Abbildung 4.7 zusammengefasst und bewegen sich im aus der Literatur bekannten Rahmen.

Routenlänge

Die mittlere Länge der vom Server berechneten Routen beträgt 302,3 m ($Mdn = 242,3$ m; $SD = 226,3$ m) und entspricht somit fast exakt dem von Ullmann (2020, S. 112) berichteten Wert. Dies ist ein Hinweis dafür, dass es sich beim erhobenen Datensatz um eine für den Untersuchungsbereich repräsentative Stichprobe handelt.



(a) Sitzungsdauer im Wochenverlauf

(b) Sitzungsdauer im Tagesverlauf

Abbildung 4.7: Einfluss von Wochentag und Tageszeit auf die Sitzungsdauer

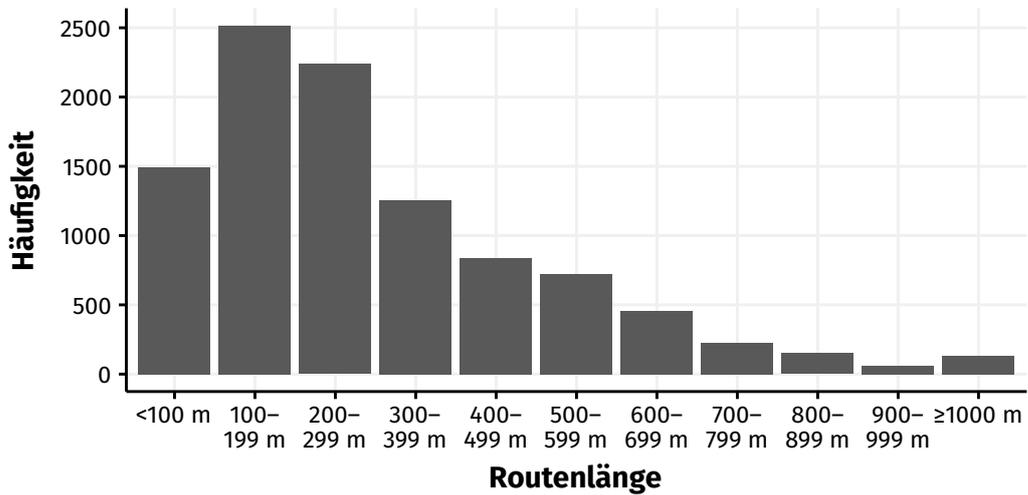


Abbildung 4.8: Verteilung der Routenlängen

Abbildung 4.8 illustriert die Verteilung der Routenlängen. Knapp die Hälfte der Routen sind zwischen 100 und 300 m lang, woraufhin die Häufigkeit schnell abnimmt. Routen mit einer Länge von mehr als 600 m machen nur etwa 10 % aller Anfragen aus. Mit diesem Wissen kann – auch unter Einbeziehung der im vorigen Abschnitt beschriebenen räumlichen Unterteilungen – die typische Distanz abgeschätzt werden, die ein Indoor-Positionierungssystem unter realistischen Bedingungen zurücklegen muss.

Schrittzahl

Eine weitere Möglichkeit zur Charakterisierung einer Sitzung ist die Anzahl der Schritte, die vom Schritterkenner registriert wurden. Anhand der Route ist bekannt, welche Strecke zurückgelegt werden muss, um ans Ziel zu gelangen – unter der Annahme, man befindet sich am Startort. Mithilfe eines Richtwerts von ca. 0,65 m als mittlerer Schrittlänge lässt sich damit auch die nötige Schrittzahl schätzen.

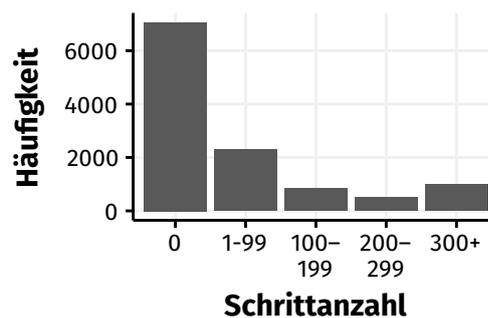


Abbildung 4.9: Vom Schritterkenner registrierte Schritte pro Sitzung

Die Analyse der Schrittdaten zeigt, dass im Großteil der Fälle keine oder nur sehr wenig Bewegung stattfindet, lediglich in etwa einem Fünftel der Sitzungen wurden 100 oder mehr Schritte erfasst (Abbildung 4.9). Die Tatsache, dass keine Schritte registriert wurden, ist nicht auf den fehlenden Sensor oder eine nicht funktionierende Rückfall-Implementierung der Schritterkennung zurückzuführen; in den meisten Fällen sind die betroffenen Geräte mit einem Schritterkenner ausgestattet.

In 8,7 % der Sitzungen wurde eine Distanz mindestens in Länge der zugehörigen Route zurückgelegt (bei einer angenommenen mittleren Schrittlänge von 0,65 m), was jedoch nicht zwangsweise mit dem Erreichen des Ziels gleichzusetzen ist. In 16,6 % der Sitzungen wurde wenigstens die Hälfte der Strecke absolviert, in 22,6 % mindestens ein Viertel. Diese Beobachtungen – insbesondere die hohe

Anzahl von Sitzungen ohne jegliche Schritte – deuten darauf hin, dass in einem großen Teil der App-Nutzung die tatsächliche Navigation entlang der geplanten Route keine Rolle spielt.

Startposition

In diesem Zusammenhang stellt sich die Frage, wie häufig sich BenutzerInnen bei der Initiierung eines Navigationsvorgangs auf dem Campus befinden. Zur Beantwortung der Frage können die Logdaten des Fused-Location-Providers herangezogen werden. Die aufgrund der GPS-Komponente im Innenbereich niedrige Genauigkeit ist für diese Unterscheidung kein Nachteil, da sich der Randbereich des Campus durchwegs im Freien befindet, wo eine ausreichend exakte Positionsbestimmung möglich ist.

Die Analyse der Daten zeigt, dass die Startposition des Benutzers nur in 6 von 10 Fällen auf dem Campus liegt. Hierbei wurden diejenigen Sitzungen nicht berücksichtigt, in denen keine Information über die Nutzerposition vorliegt.

Zu unterscheiden ist die Startposition vom Anfangspunkt der Route. Zwar bietet die Navigations-App auf dem Campus die Möglichkeit, von der aktuellen Position aus Routen zu berechnen, der Anfangspunkt kann jedoch auch frei gewählt werden. Im Durchschnitt ist der Anfangspunkt der Route ca. 95 m entfernt ($Mdn = 45$ m), lediglich 18,3 % der Sitzungen beginnen in weniger als 10 m Entfernung. Obwohl diese Werte aufgrund der angesprochenen Ungenauigkeit der GPS-Position mit Unsicherheit behaftet sind, zeigt sich doch deutlich, dass BenutzerInnen der App häufig nach Routen suchen, auf denen sie nicht unmittelbar die Navigation starten können.

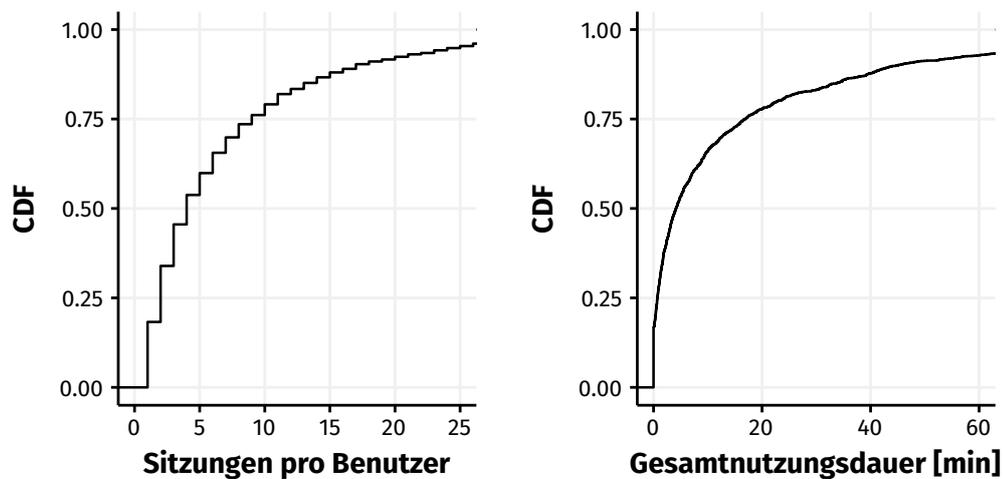
So ist es auch keine Überraschung, dass Sitzungen, die auf dem Campus beginnen, signifikant länger dauern ($Mdn = 81$ s) als andere ($Mdn = 45$ s), was durch einen Wilcoxon-Rangsummentest bestätigt wird ($W = 5082131; p < 0,01; r = 0,21$).

4.4.3 Nutzercharakteristik

Die bisherigen Analysen zielten darauf ab, aus dem Datensatz als Ganzes und über die gesamte Nutzerschaft hinweg Erkenntnisse über das Nutzungsverhalten zu gewinnen. In diesem Abschnitt werden die Daten im Hinblick auf charakteristische Eigenschaften der BenutzerInnen ausgewertet.

Zunächst ist von Interesse, inwiefern die App regelmäßig verwendet wird. Im Median wurden von jeder Person vier Sitzungen registriert ($M = 7,3; SD = 9,4$), die maximale Anzahl beträgt 125. 18,3 % der BenutzerInnen starten lediglich eine einzige Sitzung, 23,9 % mindestens 10, und 5,2 % sind Vielnutzer mit mehr als 25

Sitzungen. Es ergibt sich also eine klassische Long-Tail-Verteilung, was auch an der steil ansteigenden kumulativen Dichtefunktion (Abbildung 4.10a) erkennbar ist. Ähnlich verhält es sich mit der Gesamtnutzungsdauer pro Benutzer, welche im Median bei 6,7 min liegt. Etwa ein Viertel der BenutzerInnen verwendet die App weniger als 2 min lang, bei 8,7% akkumuliert sich die Dauer zu mehr als einer Stunde (siehe Abbildung 4.10b). Insgesamt zeigt sich also ein sehr diverses Bild der Nutzerschaft, mit einigen wenigen intensiven NutzerInnen und einem großen Anteil an sporadischen oder seltenen Sitzungen.



(a) Anzahl registrierter Sitzungen pro Benutzer (nicht abgebildet: 74 BenutzerInnen mit mehr als 25 Sitzungen)

(b) Gesamtnutzungsdauer pro Benutzer (nicht abgebildet: 116 BenutzerInnen mit mehr als 60 min Dauer)

Abbildung 4.10: Nutzungsstatistiken pro Benutzer

Der vorige Abschnitt zeigte, dass in vielen Sitzungen keine oder nur wenig Bewegung stattfindet. Wiederholt man diese Analyse in Hinblick auf die einzelnen BenutzerInnen, so wird deutlich, dass von knapp einem Drittel der Nutzerschaft überhaupt keine Schritte getätigt werden, während eine kleine Gruppe eine kumulative Schrittzahl von mehreren Tausend erreicht. Abbildung 4.11 illustriert diese erneute Long-Tail-Verteilung. Im Median werden pro Nutzer über alle Sitzungen hinweg 151 Schritte registriert ($M = 620,1$; $SD = 1303,5$).

Auch die an den BenutzerInnen orientierte Betrachtung der Daten führt also zu weiteren Hinweisen, dass mithilfe der App häufig Routen geplant werden, ohne sie aber zur Navigation zum Ziel zu verwenden.

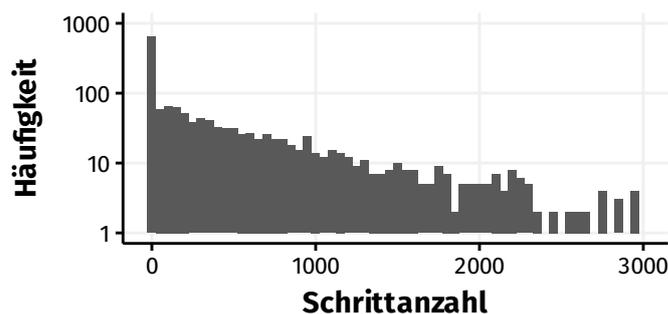


Abbildung 4.11: Kumulative Verteilung der pro Benutzer getätigten Schritte
(nicht abgebildet: 62 BenutzerInnen mit mehr als 3000 Schritten)

4.4.4 Geräte und Sensoren

Im Android-Ökosystem gilt es, unterschiedliche Hersteller, Geräte und Formfaktoren zu unterstützen, um breite Einsetzbarkeit einer Positionierungslösung zu gewährleisten. Im Folgenden ist daher die Geräte-Verteilung sowie die Sensor-Ausstattung und deren berichtete Genauigkeit von Interesse.

Gerätehersteller und Modelle

Auf der von Google bereitgestellten Google-Play-Kompatibilitätsliste (Google, 2021b) befinden sich – mit Stand April 2021 – 21906 Gerätetypen von 2120 Herstellern bzw. Marken, teils mit mehreren Untervarianten. Unter der Nutzerschaft der URwalking-App kommen davon 272 Gerätetypen von 32 unterschiedlichen Marken zum Einsatz. Der Großteil der Sitzungen (knapp 80 %) entfällt auf die beiden Hersteller Samsung und Huawei. Die Android-Hauptversion der Geräte erstreckt sich von 6.0 aus dem Jahr 2015 bis zu der bei Abschluss der Datenaufzeichnung aktuellen Version 11.

Verfügbare Sensoren

Die Vielzahl von Gerätetypen bringt die Problematik unterschiedlicher Sensor-Ausstattungen mit sich. Abbildung 4.12 zeigt, mit welcher Häufigkeit die unterschiedlichen Sensortypen in den im Datensatz enthaltenen Geräten verfügbar sind.

Während die meisten Sensoren, beginnend beim Accelerometer und endend beim GRV, in mehr als 90 % der Geräte verbaut sind, ist der für die PDR-basierte Indoor-Positionierung elementare Schrittkenner deutlich seltener verfügbar. Dadurch wird die Notwendigkeit einer auf dem Accelerometer basierenden Rückfalllösung unterstrichen.

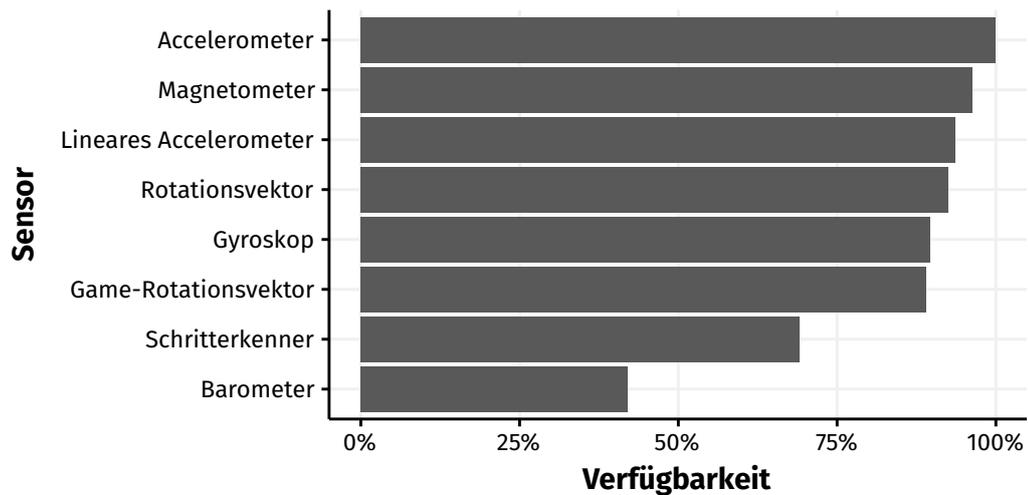


Abbildung 4.12: Verfügbarkeit ausgewählter Sensoren auf den Geräten im Datensatz

Unter den betrachteten Sensoren noch seltener vorhanden – in weniger als der Hälfte aller Geräte – ist das Barometer, welches in der Indoor-Positionierung ebenfalls eine wichtige Rolle bei der Erkennung von Stockwerksübergängen einnimmt.

Genauigkeit der Sensoren

Wie in Abschnitt 2.3.2 beschrieben, geben Android-Sensoren neben den eigentlichen Messwerten auch Auskunft über Änderungen in der Genauigkeit der Daten. Der erhobene Datensatz bietet die Möglichkeit, die berichtete Genauigkeit der Sensoren in einer breiten Stichprobe zu untersuchen.

Im Falle von Messungen des Fused-Location-Providers wird die Genauigkeit der geschätzten Position als Standardabweichung in Metern angegeben. Demnach besteht eine 68-prozentige Wahrscheinlichkeit, dass sich die tatsächliche Position innerhalb eines Kreises um die geschätzte Position mit einem Radius in Höhe der angegebenen Genauigkeit befindet (Google Developers, 2021b).

Die berichtete Genauigkeit aller Positionen innerhalb der Grenzen des Campus beträgt im Median 15,97 m ($M = 40,43$ m; $SD = 163,87$ m). Der Wertebereich erstreckt sich dabei von einem Minimum von 1 m bis zu 5734 m, wodurch das Ausmaß der potentiellen Ungenauigkeit der Positionsbestimmung deutlich wird. Ein Vergleich der Genauigkeit zwischen Positionen, die sich innerhalb und außerhalb von Gebäuden befinden, ist aufgrund der hohen Varianz nur bedingt aussagekräftig. Unter der Annahme, dass die geschätzten Positionen im Mittel um den tatsächlichen Standort streuen, zeigt sich erwartungsgemäß im Außenbereich eine höhere Genauigkeit von 10,0 m (Median) im Vergleich zu 17,4 m innerhalb von

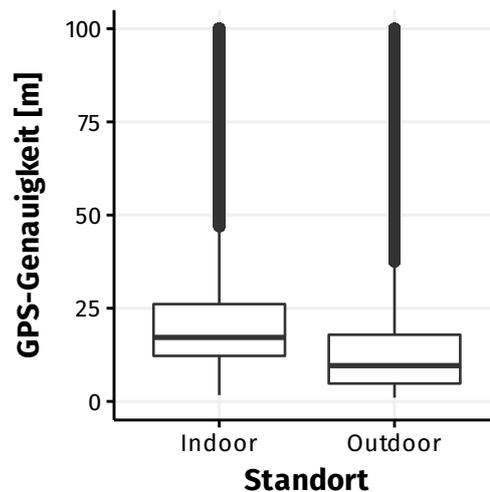


Abbildung 4.13: Vergleich der GPS-Genauigkeit innerhalb und außerhalb von Gebäuden (Werte > 100 m nicht abgebildet).

Gebäuden (siehe Abbildung 4.13).

Auch die meisten anderen Android-Sensoren liefern neben den eigentlichen Messwerten eine Einschätzung über die Verlässlichkeit derselben mit. Die berichtete Genauigkeit ausgewählter Sensortypen über den kompletten Datensatz hinweg ist in Abbildung 4.14 dargestellt. Erfreulicherweise ist in allen Fällen die hohe Genauigkeit mit Abstand am häufigsten vertreten – man beachte die stark logarithmische y-Achse. Die meisten Abweichungen davon sind beim Magnetfeldsensor beobachtbar, was den zuvor gemachten Erfahrungen hinsichtlich seiner begrenzten Zuverlässigkeit innerhalb von Gebäuden entspricht.

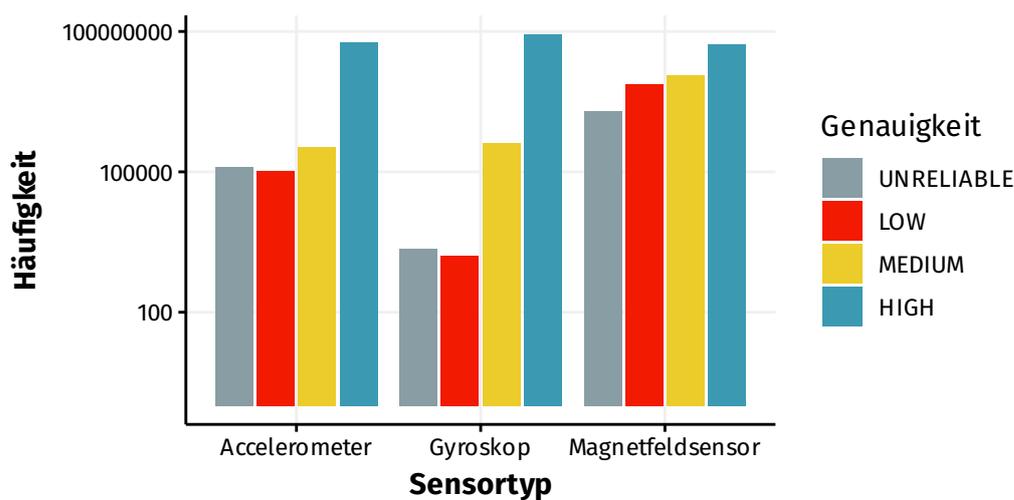


Abbildung 4.14: Berichtete Genauigkeit von ausgewählten Sensoren

Bei genauerer Analyse zeigt sich, dass das Accelerometer bei 56,4 %, das Gyroskop bei 83,2 % und der Magnetfeldsensor bei 33,2 % aller Geräte während des gesamten Erhebungszeitraums konstant hohe Genauigkeit berichten. Etwas besser verhält es sich im Hinblick auf einzelne Sitzungen, mit 64,4 % (Accelerometer), 94,2 % (Gyroskop) respektive 58,8 % (Magnetfeldsensor) ausnahmslos hoher Genauigkeit.

4.5 Benutzerverhalten während der Navigation

Die bisherigen Ergebnisse haben gezeigt, dass die App in vielen Fällen nicht für tatsächliche Navigation verwendet wird. Dies lässt sich am besten daran festmachen, dass viele BenutzerInnen keine oder nur wenige Schritte zurücklegen oder sich zum Zeitpunkt der App-Nutzung überhaupt nicht auf dem Campus befinden. Vielmehr zeichnet die Auswertung der Daten das Bild, dass häufig Routen im Voraus geplant werden, sei es von zu Hause aus oder in einem anderem Bereich auf dem Campus.

Im Rahmen dieser Arbeit ist jedoch das Nutzerverhalten während tatsächlicher Navigationsvorgänge von Interesse – nur dort wird schließlich auch ein funktionierendes Positionierungssystem benötigt. Aus diesem Grund erfolgt vor den weiteren Analysen des Datensatzes ein weiterer Filterungsschritt mithilfe einer Heuristik, welche im folgenden Abschnitt beschrieben wird.

Anschließend folgen Analysen zu unterschiedlichen Aspekten, um das für die Positionsbestimmung relevante Navigationsverhalten unter realen Bedingungen zu ergründen. Den Anfang bildet das Gehverhalten, inklusive der unterwegs eingelegten Pausen, gefolgt von Unterbrechungen in Form von Zeiträumen, in denen die Navigations-App im Hintergrund lief. Danach wird die Geräteausrichtung untersucht, um Zeiträume aktiver Nutzung zu bestimmen.

4.5.1 Filterheuristik für tatsächliche Navigationssitzungen

Um die Sitzungen zu filtern, werden sie zunächst anhand der im vorigen Abschnitt erarbeiteten Merkmale geclustert. Da es sich sowohl um metrische (z. B. Sitzungsdauer, Routenlänge, Schrittzahl) als auch um kategorische (Startposition auf dem Campus?) Merkmale handelt, muss eine angepasste Distanzmetrik verwendet werden. In solchen Fällen bietet sich die Gower-Distanz an, welche die Ähnlichkeit zwischen zwei Elementen anhand von n merkmalsbezogenen Teildistanzen

definiert (Gower, 1971):

$$D_{Gower}(x, y) = \frac{1}{n} \sum_{k=1}^n d_k(x, y) \quad (4.4)$$

Im Falle kategorischer Daten beträgt die Distanz bei Übereinstimmung 0, ansonsten 1. Für metrisch skalierte Merkmale kommt die Formel

$$d_k(x, y) = \frac{|x - y|}{R_k} \quad (4.5)$$

zum Einsatz, wobei R_k die Spannweite zwischen Minimum und Maximum beschreibt (vgl. Kuhn & Johnson, 2019, S. 200 f.).

Die auf dieser Metrik basierende Distanzmatrix bildet anschließend die Grundlage für das Clustering. Fehlende Werte, welche beispielsweise bei der Sitzungsdauer vorkommen, werden durch Imputation des Medians ersetzt. Als Clusteringverfahren kommt das k-Medoids-Verfahren zur Anwendung, welches im Gegensatz zum verbreiteten k-Means auch mit selbst definierten Distanzmetriken umzugehen weiß (Hastie, Tibshirani & Friedman, 2009, S. 515 ff.). Die optimale Clusteranzahl – in diesem Fall drei – wurde mithilfe des Silhouettenkoeffizienten (Kaufman & Rousseeuw, 2005, S. 87 f.) bestimmt.

Abbildung 4.15 zeigt die Güte des Clusterings visuell anhand des t-SNE-Verfahrens (*t-Distributed Stochastic Neighbor Embedding*), welches eine dimensionsreduzierte Repräsentation der Daten erzeugt (van der Maaten & Hinton, 2008). Bei Betrachtung der Clusterinhalte wurde deutlich, dass sich die für die folgenden Analysen relevanten Sitzungen in Cluster 1 befinden. Um den Datensatz auf tatsächliche Navigationssitzungen zu beschränken, werden zusätzlich die folgenden Kriterien angewendet:

- Die Sitzung beginnt auf dem Campus, d. h. die erste vom Fused-Location-Provider registrierte Position befindet sich innerhalb des durch die Grenzen des Campus vorgegebenen Polygons.
- Der Startpunkt der Route ist nicht weiter als 50 m vom Standort des Benutzers entfernt. Durch die Wahl des vergleichsweise hohen Grenzwerts wird der bereits angesprochenen Ungenauigkeit der Positionsbestimmung im Innenbereich Rechnung getragen.
- Die Dauer der Sitzung beträgt mindestens 30 s und maximal 60 min. Die obere Grenze erlaubt es einem Fußgänger, alle Ziele auf dem Campus zu erreichen, selbst wenn unterwegs Pausen eingelegt werden.
- Die Route ist mindestens 50 m lang, wodurch ein Mindestmaß an Navigati-

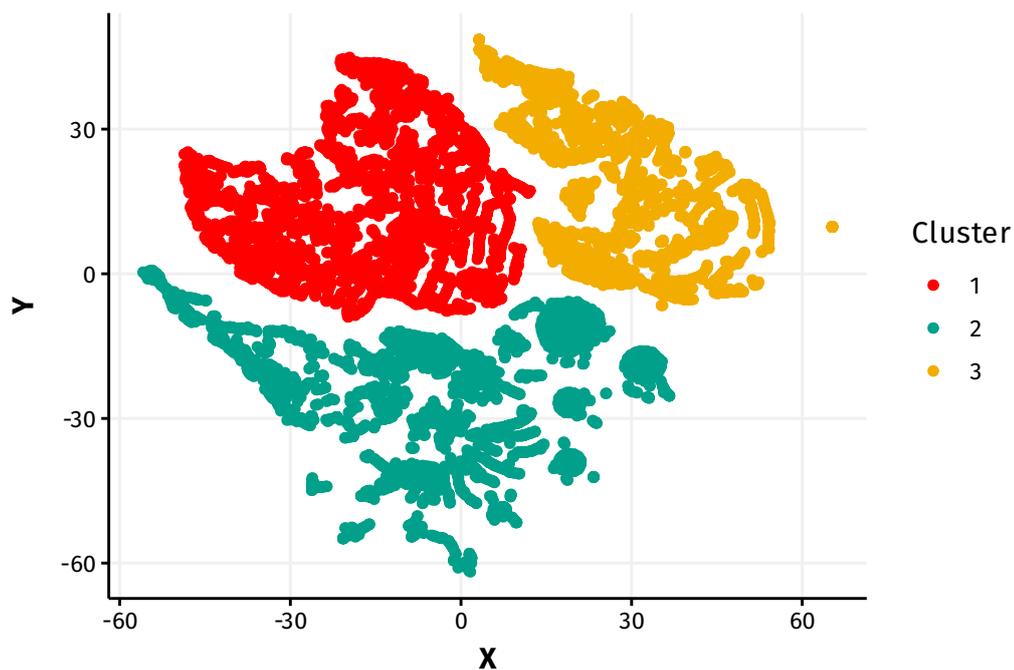


Abbildung 4.15: Darstellung des Clustering-Ergebnisses mithilfe des t-SNE-Verfahrens

onsaufwand gefordert ist.

- Die Routendistanz wird – gemäß der Anzahl der registrierten Schritte – mindestens zur Hälfte absolviert. Dadurch ist sichergestellt, dass wenigstens ein Versuch der tatsächlichen Navigation unternommen wird.

Nach Anwendung der Kriterien bleiben 600 Sitzungen von 351 verschiedenen BenutzerInnen übrig, welche sich alle im Zeitraum zwischen 7 und 21 Uhr befinden. Abgesehen von 13 Ausnahmen fallen zudem alle Sitzungen auf einen Arbeitstag (Mo–Fr). Alle folgenden Analysen stützen sich auf diesen gefilterten Datensatz. Die Routen in diesen Sitzungen addieren sich zu einer geplanten Strecke von 119,8 km und decken, wie aus Abbildung 4.16 hervorgeht, nahezu den gesamten Campus ab. In vielen Routen enthalten, in der Abbildung in gelb gefärbt, sind vor allem Areale im zentralen Bereich sowie auf den Nord-Süd-Achsen des Campus.

4.5.2 Gehverhalten

Da das in dieser Arbeit entwickelte Indoor-Positionierungssystem auf der Schritt-erkennung basiert, ist zunächst das Gehverhalten der BenutzerInnen von Interesse, weswegen im Folgenden die Schrittfrequenz und Pausen während des Gehens analysiert werden.

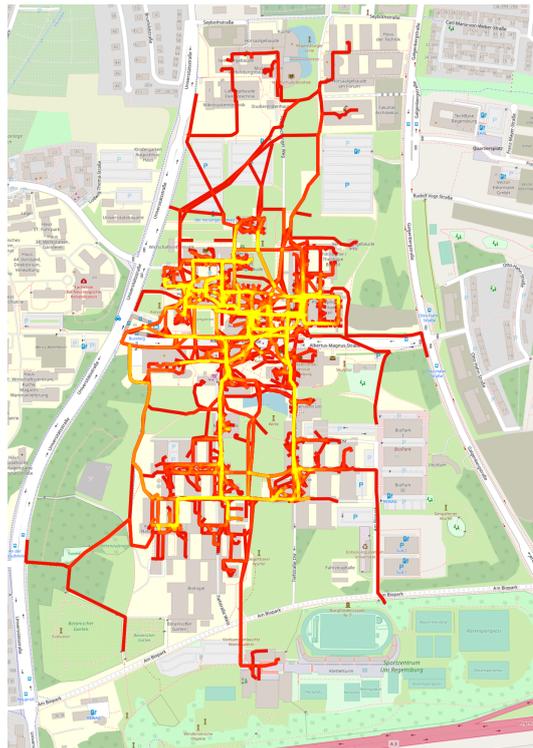


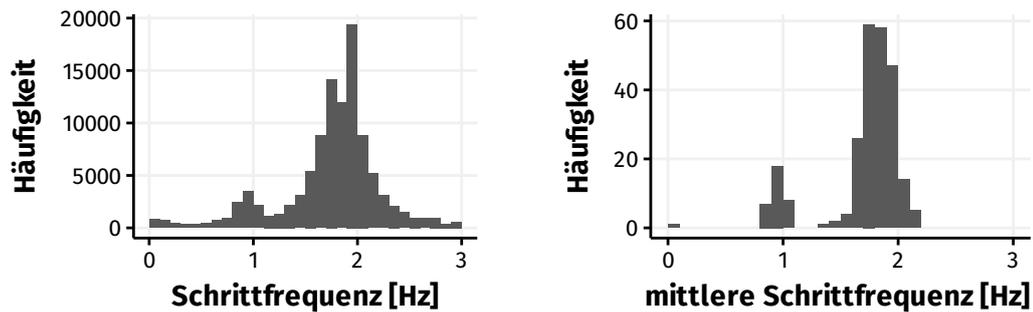
Abbildung 4.16: Heatmap des Verlaufs aller Routen im gefilterten Datensatz.
Kartenmaterial von OpenStreetMap (Haklay & Weber, 2008)

Schrittfrequenz

Im vorigen Kapitel wurde bereits die Schrittfrequenz als ein charakteristisches Merkmal angesprochen, welches in einem Positionierungssystem mithilfe zusätzlicher Informationen zur Schrittlängenschätzung dienen kann.

Der zeitliche Abstand zwischen zwei Schritten beträgt über den Datensatz hinweg – insgesamt wurden 106 038 Schritte registriert – im arithmetischen Mittel 998 ms, der Median liegt bei lediglich 541 ms. Aus dieser Diskrepanz folgt, dass es Ausreißer nach oben, d. h. längere Pausen zwischen zwei Schritten, geben muss. Die Schrittfrequenz streut erwartungsgemäß um einen Wert von knapp 2 Hz. Bei Analyse der Verteilung der Schrittfrequenz (Abbildung 4.17a) ist ein zweites lokales Maximum bei ca. der halben Frequenz zu erkennen, welches auf die Auslassung bzw. Nicht-Erkennung jedes zweiten Schritts unter gewissen Umständen hindeutet.

Ein ähnliches Bild zeigt sich bei Betrachtung der mittleren Schrittfrequenz pro Benutzer (Abbildung 4.17b). Bei einer klar abgegrenzten Benutzergruppe beträgt die Frequenz konsistent nur in etwa die Hälfte derjenigen der restlichen Nutzer. Die Erklärung ist durch Abgleich mit den Gerätedaten zu finden: Für die halbierte Schrittfrequenz sind ausschließlich Sensoren des Herstellers Qualcomm verant-



(a) Verteilung der Schrittfrequenz
(Schritte einzeln)

(b) Verteilung der Schrittfrequenz
(Median) pro Benutzer

Abbildung 4.17: Schrittfrequenz-Verteilung

wortlich, verbaut u. a. in Geräten der Marke OnePlus. Da von diesen Sensoren keinerlei Datenpunkte im erwarteten Bereich von ca. 2 Hz vorliegen, ist davon auszugehen, dass hier entgegen der Spezifikation nicht einzelne Schritte, sondern Doppelschritte (engl. *strides*) gemessen wurden. Da sich dies jedoch mangels Dokumentation nicht zweifelsfrei verifizieren lässt, werden die betreffenden Daten (7,6 % der Gesamtschritte) in den nachfolgenden Analysen nicht berücksichtigt. Dieses Beispiel illustriert anschaulich die Herausforderungen, die bei der Arbeit mit Daten aus nur teilweise kontrollierbaren Quellen sowie bei der Entwicklung eines für den breiten Einsatz gedachten Positionierungssystems auftreten. Ohne Wissen um die Eigenheiten der Sensoren würde in diesem Fall nur jeder zweite Schritt gezählt, was bei PDR-basierten Systemen unweigerlich zu stark verfälschten Positionsschätzungen führt.

Insgesamt befindet sich die Schrittfrequenz am unteren Ende des aus der Literatur erwartbaren Rahmens, was auf eine leicht verlangsamte Gehgeschwindigkeit hindeutet (Öberg et al., 1993, S. 219).

Pausen

Gemäß dem Modell von Montello (2005) sind während der Navigation Planungs- und Orientierungsphasen (*Wayfinding*), während derer der Benutzer möglicherweise stillsteht, sowie Fortbewegungsphasen (*Locomotion*) zu erwarten. Ein längeres Stehenbleiben deutet darauf hin, dass unter Umständen die Anweisungen nicht verstanden wurden und zusätzliche Assistenz nötig ist. Aus diesem Grund sollte eine in den Navigationsprozess integrierte Positionierungslösung Informationen über derartige Pausen erfassen und zur Verfügung stellen. Durch die Analyse von Bewegungsdaten des Schritterkenners können die beiden Phasen grob klassifiziert werden.

Als Pause wird im Folgenden das Ausbleiben von Schritten in einem Zeitraum von mindestens 2 s bezeichnet – ein Wert, der auch von Brändle, Bauer und Seer (2006) als untere Grenze für die Erkennung von stehenbleibenden FußgängerInnen verwendet wurde. Die Wahl dieses Grenzwerts verhindert ebenso, dass sporadisch nicht erkannte Schritte sofort als Pausen interpretiert werden: Bei einer Schrittfrequenz von ca. 2 Hz wären dafür mehrere aufeinanderfolgende Fehler nötig.

Der gefilterte Datensatz enthält 2245 derartige Pausen mit einer rechtsschief verteilten Dauer ($M = 16,5$ s; $Mdn = 5,9$ s). Um der stark ungleich verteilten Dauer Rechnung zu tragen, werden in der folgenden Analyse Pausen unterschiedlicher Länge separat betrachtet. Dabei wird das arithmetische Mittel für die Aufteilung herangezogen, wodurch sich zwei Gruppen ergeben: Kurze Pausen stellen vorübergehendes Zögern dar, in dem sich der Benutzer möglicherweise neu orientiert und dann die Navigation fortführt. Längere Stops deuten auf eine tiefer gehende Unsicherheit hin, bei der zusätzliche Assistenzangebote hilfreich gewesen wären.

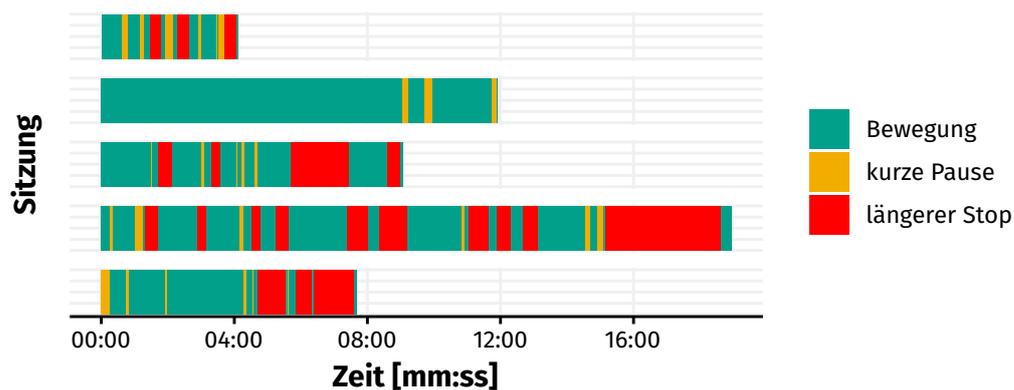


Abbildung 4.18: Beispiele für durch Gehpausen unterbrochene Navigationssitzungen

Abbildung 4.18 vermittelt einen Eindruck über die zeitliche Verteilung von Pausen und Bewegungsphasen anhand einiger beispielhaft ausgewählter Sitzungen, welche alle am gleichen Startort – der zentralen Omnibushaltestelle – beginnen.

Im ganzen Datensatz treten kurze Pausen in 83,1%, längere Stops in etwas mehr als der Hälfte (50,7%) der Sitzungen auf. Je Sitzung werden im Mittel 5,0 kurze Pausen ($Mdn = 3$) und 1,2 längere Stops registriert ($Mdn = 1$). Über alle Navigationssitzungen hinweg findet bei ca. jedem 55. Schritt eine kurze Pause statt, bei jedem 222. ein längerer Stop.

Mithilfe der Pausendefinition lässt sich auch grob der Anteil der Navigations-

zeit bestimmen, den die BenutzerInnen in Bewegung verbracht haben. Über alle Sitzungen hinweg beträgt dieser 59,7 %. Die restliche Zeit teilt sich auf kurze (11,4 %) und lange Pausen (28,9 %) auf.

Es lässt sich also konstatieren, dass die Fortbewegung während der Navigation immer wieder von Pausen unterbrochen wird – ein Aspekt, der bei der Positionierung bedacht werden muss.

4.5.3 Unterbrechungen in der App-Nutzung

Wie in Abschnitt 2.4 beschrieben, ist die URwalking-App in der Lage, die Positionsbestimmung auch im Hintergrund aufrechtzuerhalten. Wird die App während eines Navigationsvorgangs pausiert und später wieder geöffnet, so wird der Navigationsvorgang fortgesetzt. Durch Aufzeichnung des Vorder- bzw. Hintergrundstatus der App ergibt sich die Möglichkeit, neben den Geh-Pausen eine weitere Form von Unterbrechungen während der Navigation zu untersuchen.

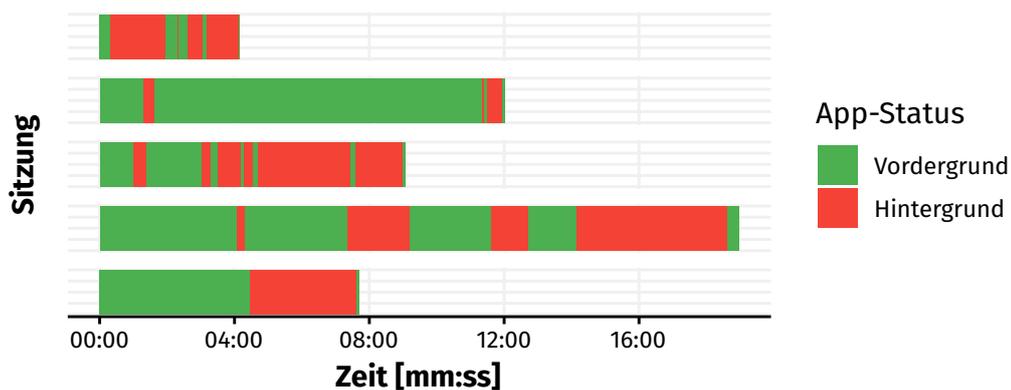


Abbildung 4.19: Vorder- und Hintergrundstatus der Navigations-App in beispielhaft ausgewählten Sitzungen

Abbildung 4.19 zeigt für dieselben exemplarischen Sitzungen wie im vorigen Abschnitt die Zeiträume an, in denen die Navigations-App im Vorder- und Hintergrund läuft. Unterbrechungen, während derer die App im Hintergrund läuft, treten in etwa zwei Dritteln (67,4 %) der Navigationssitzungen auf, im Mittel 2,3-mal pro Sitzung ($Mdn = 2$). Die Dauer der Unterbrechungen ist rechtsschief verteilt und beträgt im Mittel 124 s ($Mdn = 32$ s). Pro Sitzung wird durchschnittlich 28,3 % im Hintergrund verbracht, wobei der Anteil im Zeitverlauf bis kurz vor Ende der Sitzungen nahezu kontinuierlich zunimmt (siehe Abbildung 4.20).

Da keine Informationen darüber vorliegen, ob bzw. welche anderen Apps während der Hintergrund-Perioden verwendet werden, kann diese Studie keinen Aufschluss über die Ursachen der Unterbrechungen liefern. Wie eingangs in diesem

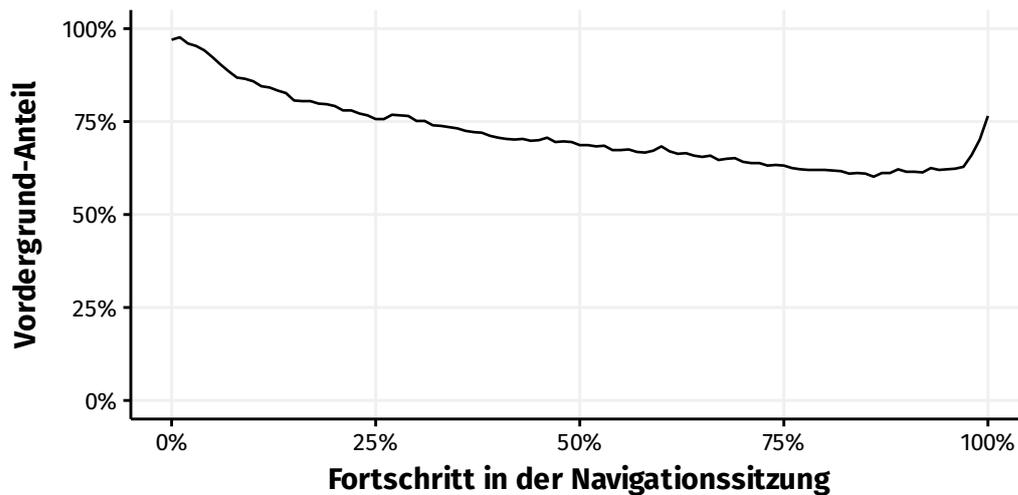


Abbildung 4.20: Verlauf des Vorder-/Hintergrundstatus im Zeitverlauf der Navigationssitzungen (auf die Sitzungslänge normiert)

Kapitel erwähnt, liegen jedoch einige Erklärungen nahe: So ist es beispielsweise denkbar, dass die Navigation durch andere Apps wie beispielsweise eingehende Textnachrichten oder Anrufe unterbrochen wird. Möglicherweise wird zum aktuellen Zeitpunkt auch schlichtweg keine Assistenz benötigt, um zum nächsten Zwischenziel zu gelangen, weswegen das Gerät in den Standby-Modus geschickt wird. Ungeachtet der Beweggründe ist jedoch klar, dass die Aufmerksamkeit des Benutzers nicht dem Navigationssystem gewidmet ist.

4.5.4 Geräteposition

Die Assistenzstrategie der App basiert darauf, dass BenutzerInnen jederzeit in der Lage sind, auf geänderte Navigationsanweisungen zu reagieren. Im vorigen Abschnitt wurde bereits der Unterschied zwischen Vorder- und Hintergrundstatus herausgearbeitet. Doch eine im Vordergrund aktive App ist alleine nicht hinreichend dafür, dass aktualisierte Anweisungen tatsächlich gesehen werden. Dies hängt auch von der Art und Weise ab, wie das Gerät getragen wird.

In der in Abschnitt 3.4.3 vorgestellten Studie trugen Testpersonen das Gerät meist geradeaus vor ihrem Körper, den Bildschirm leicht nach oben geneigt, ohne explizit dazu aufgefordert worden zu sein. Um zu überprüfen, ob dies auch für die BenutzerInnen der Navigations-App gilt, wird die Geräteausrichtung anhand des GRV untersucht.

Abbildung 4.21 zeigt die Verteilung der Geräteausrichtung in z-Richtung, von $+90^\circ$ (direkt nach oben) bis -90° (direkt nach unten). Daraus wird ersichtlich,

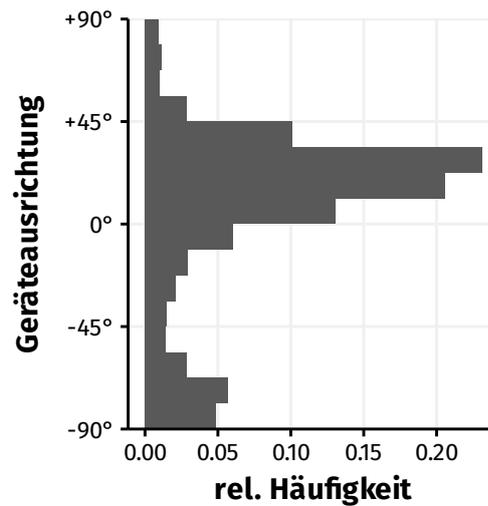


Abbildung 4.21: Verteilung der Ausrichtung des Geräts nach oben/unten

dass das Gerät tatsächlich den Großteil der Zeit horizontal bzw. leicht nach oben geneigt gehalten wird, was dem in der Literatur als »texting mode« (Klein, Solaz & Ohayon, 2018) beschriebenen Verhalten entspricht. Dies ist im Hinblick auf die Verwendung zur Richtungsbestimmung im Indoor-Positionierungssystem erfreulich. Gleichzeitig deutet jedoch ein zweites lokales Maximum am unteren Ende der Skala auf Zeiträume hin, in denen das Gerät nach unten zeigt, beispielsweise beim Tragen in einer (Hosen-)Tasche.

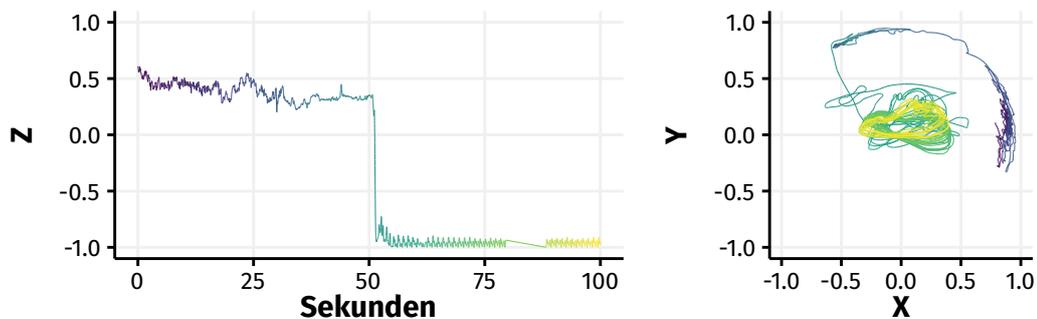


Abbildung 4.22: Beispielsitzung mit in der Tasche getragenen Gerät

Durch Visualisierung der dreidimensionalen Rohdaten der einzelnen Sitzungen lassen sich filigrane Bewegungsmuster anschaulich illustrieren und genauer analysieren. In den Abbildungen 4.22 und 4.23 ist der Verlauf der Geräteausrichtung zu Beginn zweier ausgewählter Navigationssitzungen zu sehen. Links ist jeweils die z-Komponente des normierten Orientierungsvektors im Zeitverlauf angetragen, rechts die Projektion auf die xy-Ebene. Anhand des Farbverlaufs ist auch rechts

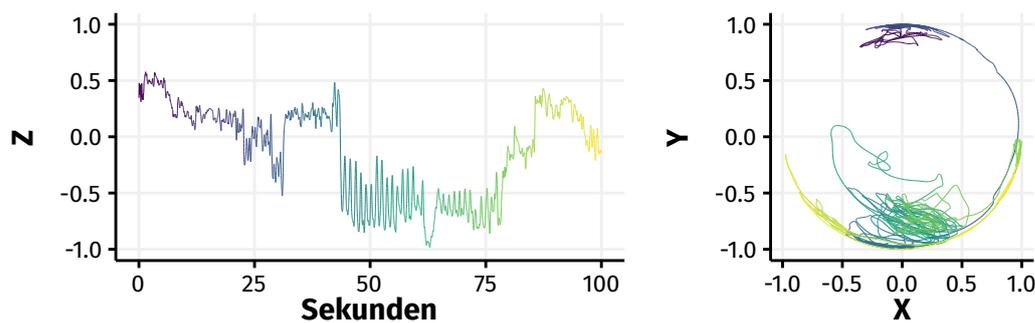


Abbildung 4.23: Beispielsitzung mit in der Hand schwingendem Gerät

die zeitliche Komponente und somit die beim Gehen entstehende Periodizität erkennbar.

In beiden Fällen befindet sich das Gerät zu Beginn der Sitzung in der üblichen, leicht nach oben geneigten Nutzungsposition. Nach einiger Zeit wird es in die Tasche gesteckt (Abbildung 4.22) bzw. in der Hand schwingend getragen (Abbildung 4.23), wodurch sich beim Gehen unterschiedliche Bewegungsmuster – insbesondere auf der z-Achse – ergeben. Dass diese Muster durch die genannten Tragearten zustande kommen, wurde experimentell überprüft, nicht immer sind diese jedoch so eindeutig wie in den gezeigten Beispielen zuzuordnen. Die automatisierte Erkennung dieser Aktivitäten und Durchführung weiterer Analysen wird daher im nächsten Kapitel in Abschnitt 5.3 wieder aufgegriffen.

4.6 Fazit der Logdatenauswertung

Die Logdatenanalyse der Campus-Navigations-App hat wertvolle Erkenntnisse über das Benutzungs- und Navigationsverhalten gebracht. Während manche Aspekte wie etwa die zeitliche Verteilung den Erwartungen entsprechen, so überrascht insbesondere der hohe Anteil an Sitzungen, die nicht zu einem Navigationsvorgang führen. Bei den tatsächlichen Navigationssitzungen wird deutlich, dass Pausen und Unterbrechungen einen erheblichen Teil der Nutzungsdauer ausmachen.

Um neben den stationären und Bewegungsphasen weitere Aktivitäten und Gerätepositionen während der Navigation zu erkennen und für die Indoor-Positionierung zu nutzen, wird im folgenden Kapitel eine ML-basierte HAR-Lösung entwickelt.

Kapitel 5

Aktivitätserkennung in der Fußgängernavigation

Die naturalistische Log-Studie im vorigen Kapitel hat Aufschluss über das Verhalten von Personen während der Fußgängernavigation gegeben. Nun wird dieses Wissen angewendet, um eine HAR-Lösung zu entwickeln, zu evaluieren und anschließend in das in Kapitel 3 beschriebene Basis-Positionierungssystem zu integrieren. Das Ziel dabei ist, eine längerfristig stabile Positionsverfolgung nicht durch Nutzung externer Infrastruktur, sondern durch gezielte Erkennung von menschlichen Aktivitäten während der Fußgängernavigation zu erreichen.

Bereits Brush et al. (2010) haben gezeigt, dass Navigationsanweisungen, die nicht nur auf der Nutzerposition beruhen, sondern auch die momentane Aktivität miteinbeziehen, bessere Assistenz leisten können. Auch die vom URwalking-Pfadplanungsserver gelieferten Navigationsanweisungen enthalten oft mehr oder weniger explizit die vom Benutzer im Verlauf der geplanten Route auszuführenden Aktivitäten. Typischerweise finden sich dort Anweisungen wie »*Biegen Sie links ab*«, »*Gehen Sie am Wegpunkt XY vorbei*« oder »*Gehen Sie durch die Tür*«.

Manche Bewegungsabläufe, beispielsweise das Treppensteigen oder das Öffnen und Schließen von Türen, sind an bestimmte Orte gebunden. Wenn es gelingt, diese Aktivitäten zuverlässig zu erkennen, so können dadurch Rückschlüsse auf die Nutzerposition gezogen werden. Voraussetzung hierfür ist ein detailliertes Umgebungsmodell, in dem die Orte verzeichnet sind, an denen die korrespondierenden Aktivitäten möglich sind. Das Konzept derartiger durch die Sensorik erkennbarer Landmarken wurde bereits von H. Wang et al. (2012) und Abdelnasser et al. (2016) beschrieben – nicht zu verwechseln mit dem Landmarkenbegriff von Lynch (1960). Im Folgenden wird daher im Kontext von HAR der Begriff *Aktivitätsort* verwendet.

Ein großer Vorteil dieses Vorgehens gegenüber anderen Methoden, die eine ab-

solute Positionsbestimmung innerhalb von Gebäuden ermöglichen (z. B. WLAN-Ortung), ist die fehlende Abhängigkeit von zusätzlicher Infrastruktur. Aufgrund der Universalität der grundlegenden Bewegungsmuster kann ein auf HAR basierendes System ohne aufwändige Kalibrierung oder Vermessung prinzipiell auch an anderen Orten eingesetzt werden. Da jedoch in der Regel lediglich Klassen von Aktivitäten erkannt werden können, nicht aber beispielsweise das Passieren von bestimmten Türen oder das Hinaufsteigen einer eindeutig identifizierbaren Treppe, muss die gewonnene Ortsinformation disambiguiert werden. Hierfür ist weiterhin eine grobe Positionsverfolgung vonnöten, welche z. B. mithilfe von PDR stattfinden kann.

Ein Beitrag dieser Arbeit ist der Fokus auf Aktivitäten, die während der Fußgängernavigation, insbesondere innerhalb von Gebäuden, relevant sind. Im Vergleich zu den in der HAR-Literatur vorherrschenden, groben Klassen wie etwa *Gehen*, *Stehen* oder *Sitzen* werden daher deutlich granularere Aktivitäten definiert. Freilich ist insbesondere die Unterscheidung *Gehen/Stehen* auch in diesem Kontext von Interesse, doch durch feinere Untergliederung lässt sich die Positionsbestimmung einerseits und die Navigationsassistentz andererseits noch präziser gestalten, beispielsweise durch gezielte Bezugnahme auf die vorzunehmenden Aktivitäten.

Bei der smartphonebasierten Fußgängernavigation besteht die Schwierigkeit, dass die Geräteposition im Bezug auf den Körper nicht fixiert ist, sondern sich das Gerät – wie schon in Abschnitt 4.5.4 gezeigt – beispielsweise in der Hand oder einer Tasche befinden kann. Die Bestimmung der Art und Weise, wie das Smartphone vom Benutzer getragen wird, ist daher eng verwandt mit der Aktivitätserkennung. Für ein unter realen Bedingungen einsetzbares System muss sichergestellt sein, dass jede der Aktivitäten unabhängig von der Trageart erkannt werden kann.

Der Rest dieses Kapitels ist wie folgt untergliedert: Abschnitt 5.1 gibt einen Überblick über das Forschungsfeld HAR, zunächst allgemein und später in Bezug auf die Fußgängernavigation und Indoor-Positionierung. Außerdem werden hier auch existierende HAR-Datensätze vorgestellt und die Notwendigkeit der Erhebung eines eigenen Datensatzes begründet. In Abschnitt 5.2 erfolgt die Beschreibung des in dieser Arbeit verfolgten HAR-Ansatzes, bestehend aus den betrachteten Aktivitäten, dem erhobenen Datensatz, der ML-Methodik sowie deren Evaluation. Das HAR-System wird anschließend in Abschnitt 5.3 dazu verwendet, Teilaspekte der naturalistischen Studie aus dem vorherigen Kapitel zu vertiefen. Schließlich wird in Abschnitt 5.4 die HAR-Lösung in das existierende Indoor-Positionierungssystem integriert und gemeinsam in einer abschließenden Studie evaluiert.

5.1 Das Forschungsfeld Human Activity Recognition

HAR wird in unterschiedlichsten Anwendungsgebieten eingesetzt, etwa zum Erkennen von Alltagsaktivitäten (*Activities of Daily Living*, ADL), beim Fitness-Tracking, im Gesundheitswesen, für die Heimautomatisierung und zum Erkennen von Zubereitungsschritten beim Kochen (Ramasamy Ramamurthy & Roy, 2018, S. 3).

Der folgende Abschnitt widmet sich zunächst den unterschiedlichen methodischen Herangehensweisen für die Aktivitätserkennung, bevor anschließend smartphonebasierte und im Bereich der Fußgängernavigation eingesetzte Ansätze vorgestellt werden.

5.1.1 Methodik

Wie bereits in Abschnitt 2.3.2 bei der Beschreibung der einzelnen Sensoren angemerkt, werden im Rahmen dieser Arbeit diverse Smartphone-Sensoren für die Aktivitätserkennung verwendet. Da die meisten dieser Sensoren wiederum Messwerte entlang mehrerer Achsen liefern, erhält man eine multivariate Zeitreihe, innerhalb derer es gilt, anhand von Mustern und Korrelationen die zugrundeliegende Aktivität zu erkennen. Um in der Praxis kontinuierliche und zeitnahe Aktivitätsinformationen zu erhalten, werden die Daten üblicherweise in kurze, möglicherweise überlappende Zeitfenster aufgeteilt. Die Länge der Zeitfenster ist abhängig vom Anwendungsfall; die von Lara und Labrador (2012, S. 1198) zusammengetragenen Beispiele aus der Literatur bewegen sich zwischen 0,08 s und 30 s.

Methodisch lässt sich grob zwischen zwei Ansätzen unterscheiden: Traditionelle ML- bzw. Mustererkennungsverfahren arbeiten mit im Voraus festgelegten Features, während neuere Deep-Learning-basierte Verfahren diesen Schritt automatisieren (Sousa Lima, Souto, El-Khatib, Jalali & Gama, 2019, S. 5). Beispiele für beide Ansätze werden im Folgenden separat beschrieben.

Traditionelle Verfahren

Traditionelle Verfahren umfassen beispielsweise Support-Vektor-Maschinen, Naive-Bayes-Klassifikatoren, Entscheidungsbäume oder Hidden-Markov-Modelle (Lara & Labrador, 2012, S. 1199; Sousa Lima et al., 2019, S. 2). Als Input für derartige Verfahren dienen in der Regel nicht die Sensor-Rohdaten, sondern daraus extrahierte Features, für die je nach Ausprägung eine Umwandlung der Messwerte von der Zeit- in die Frequenzdomäne nötig wird. Die Auswahl der Features (das Fea-

ture-Engineering) findet entweder datengetrieben oder intellektuell unter Berücksichtigung von Domänenwissen statt (Kuhn & Johnson, 2019, S. 13).

Als Beispiele für in der Literatur verwendete zeitbasierte Features für Beschleunigungssignale führen Lara und Labrador (2012, S. 1197) u. a. das arithmetische Mittel, die Standardabweichung und Varianz, den Interquartilsabstand und die mittlere absolute Abweichung auf. Gerade bei periodischen Aktivitäten bietet sich die Anwendung von Fourier-Transformationen oder diskreten Kosinustransformationen für frequenzbasierte Features an (Z. He & Jin, 2009). Verfahren wie die Hauptkomponentenanalyse erlauben die Auswahl von Features mit der höchsten Trennschärfe (Z. He & Jin, 2009).

Deep-Learning-Verfahren

Dank steigender Rechenleistung auch und insbesondere auf mobilen Geräten haben sich Deep-Learning-Verfahren in den letzten Jahren immer mehr zum De-Facto-Standard für HAR-Verfahren entwickelt (Sousa Lima et al., 2019, S. 2). Bei der Entwicklung derartiger Systeme verlagert sich der Aufwand weg vom Feature-Engineering und hin zur Optimierung der verwendeten Modell-Architektur (J. Wang, Chen, Hao, Peng & Hu, 2019, S. 1). Die Forschung verteilt sich vor allem auf verschiedene Varianten von künstlichen neuronalen Netzen, darunter *Convolutional Neural Networks* (CNNs) und *Recurrent Neural Networks* (RNNs).

CNNs verarbeiten mithilfe von Faltungs- und Poolingoperationen mehrdimensionale Input-Daten und werden u. a. auch für die Bild- und Spracherkennung eingesetzt (LeCun, Bengio & Hinton, 2015). Für die Aktivitätserkennung eignen sie sich, so J. Wang et al. (2019, S. 4), aufgrund ihrer Fähigkeit zur Berücksichtigung lokaler Abhängigkeiten und zur Skalierung, im vorliegenden Anwendungsfall beispielsweise relevant für die Anpassung an unterschiedliche Schrittfrequenzen.

Beispiele für den Einsatz von CNNs für HAR finden sich bei Zeng et al. (2014), Ravi, Wong, Lo und Yang (2016) und Lee, Yoon und Cho (2017). Basierend auf der *Inception*-Architektur (Szegedy et al., 2015), in der durch geschickte Anordnung von Faltungsschichten ein neuer Bestwert für die Bildklassifikation in der ImageNet Large-Scale Visual Recognition Challenge 2014 erreicht wurde, stellen Ismail Fawaz et al. (2020) *InceptionTime* als eine speziell für die Zeitreihenklassifikation entwickelte Variante vor.

In tiefen neuronalen Netzen tritt in der Praxis – u. a. aufgrund der begrenzten Genauigkeit von Fließkommazahlen – das Problem verschwindender Gradienten auf, wodurch der Einfluss von Informationen aus früheren Schichten abnimmt; im Extremfall ist kein stabiles Training möglich (C. C. Aggarwal, 2018, S. 129 ff.). Eine Lösung für diese Problematik stellen die *residual networks* (ResNets) dar, wel-

che mithilfe von sog. *skip connections* die Rückpropagierung vereinfachen (K. He, Zhang, Ren & Sun, 2016). Eine ResNet-Variante für die Zeitreihen-Klassifikation wurde von Z. Wang, Yan und Oates (2017) vorgestellt.

RNNs unterscheiden sich von herkömmlichen neuronalen Netzen durch eine zusätzliche temporale Komponente und sind auf diese Weise prinzipbedingt in der Lage, zeitliche Abhängigkeiten und Veränderungen zu erkennen. Aus diesem Grund werden sie u. a. im Bereich der maschinellen Übersetzung, für Textsynthese und Zeitreihenanalysen eingesetzt (C. C. Aggarwal, 2018, S. 273).

RNNs endlicher Länge sind äquivalent als sehr tiefe Netze darstellbar, weswegen auch hier die Problematik verschwindender Gradienten auftritt (C. C. Aggarwal, 2018, S. 286). In der Praxis wird daher der Einsatz von *Long-Short-Term-Memory*-Zellen (LSTMs) oder *Gated Recurrent Units* (GRUs) bevorzugt, die durch zusätzliche Mechanismen modellieren, welche Informationen über längere Zeiträume erhalten bleiben (C. C. Aggarwal, 2018, S. 273 f.). Einige Beispiele für RNN-basierte HAR-Systeme finden sich bei Edel und Köppe (2016), Y. Guan und Plötz (2017) und Murad und Pyun (2017).

Hybride Modelle aus CNN- und RNN-Bestandteilen kombinieren die Vorteile der beiden Ansätze. Als Vertreter für die Aktivitätserkennung sind hier u. a. *DeepConvLSTM* (Ordóñez & Roggen, 2016) und *DeepSense* (Yao, Hu, Zhao, Zhang & Abdelzahr, 2017) zu nennen, welche jeweils aus mehreren Faltungsschichten, gefolgt von LSTMs bzw. GRUs, bestehen. Die auf diese Weise entstehenden Netze sind in ihrer Anwendung sehr flexibel und erreichen in der Evaluation deutlich bessere Klassifikationsergebnisse als etwa die traditionellen Verfahren.

Einige der soeben vorgestellten Verfahren (CNN, DeepConvLSTM, Inception-Time und ResNet) werden später in Abschnitt 5.2 für die Erkennung von navigationsspezifischen Aktivitäten erprobt.

Hierarchische Klassifikation

Bei der Entwicklung der HAR-Lösung für die Fußgängernavigation kommt die Idee der hierarchischen Klassifikation zum Einsatz. Anstelle eines flachen Klassifikators, welcher zwischen allen Aktivitäten und Gerätepositionen unterscheidet, wird eine Klassifikator-Hierarchie trainiert. Diese bildet die inhärente Hierarchie der zu unterscheidenden Klassen nach, indem beispielsweise zunächst zwischen Aktivitäten mit und ohne Bewegung differenziert wird, wonach einzelne Subkategorien erkannt werden (mehr dazu in Abschnitt 5.2.1).

Ein Vorteil dieses Ansatzes ist, dass sich die Anzahl der Klassen für jeden der einzelnen Klassifikatoren verringert, wodurch tendenziell eine höhere Trennschärfe erreicht werden kann. Einzelne Klassifikatoren können unabhängig vom Rest des

Systems optimiert oder sogar komplett durch andere Methoden ersetzt werden. Sollen zusätzliche Aktivitäten erkannt werden, ist es ausreichend, nur den betroffenen Teil der Hierarchie zu aktualisieren. Der wichtigste Nachteil ist der erhöhte Bedarf an Speicherplatz und Rechenleistung, der aufgrund der Bereitstellung und Ausführung mehrerer ML-Modelle entsteht. In Folge dessen steigt zudem die Latenz bis zum finalen Klassifikations-Ergebnis. Inwiefern dies in der Praxis eine Rolle spielt, wird später in der Evaluation untersucht.

Im Kontext von ADL wurde der hierarchische Ansatz bereits erfolgreich von Zhang, McCullagh, Nugent und Zheng (2010) eingesetzt. Sie unterscheiden in einem ersten Schritt zwischen bewegungsarmen und bewegungsbetonten Aktivitäten, woraufhin in einem zweiten Schritt mithilfe von Support-Vektor-Maschinen feiner definierte Klassen differenziert werden. Ashqar, Almannaa, Elhenawy, Rakha und House (2019) verfolgen einen alternativen hierarchischen Ansatz. Zur Erkennung der Fortbewegungsart anhand von Smartphone-Sensordaten bestimmen sie zunächst mit einem Mehrklassen-Klassifikator die zwei wahrscheinlichsten Klassen. Das finale Ergebnis kommt anschließend durch nachgeschaltete binäre Klassifikatoren zustande. J. Wang et al. (2019, S. 8) listen das Zusammenwirken von tiefen und flachen Modellen als eine der offenen Herausforderungen für HAR auf, der hierarchische Ansatz sei hierfür eine mögliche Lösung.

5.1.2 Smartphonebasierte Aktivitätserkennung

Gemäß der Zielsetzung dieser Arbeit liegt der Fokus auf HAR-Verfahren, die sich auf die in modernen Smartphones verbauten Sensoren beschränken. Dies schließt einerseits Lösungen aus, die auf zusätzlichen, beispielsweise am Fuß oder an der Hüfte montierten Sensoren basieren (siehe z. B. Bulling, Blanke & Schiele, 2014; Lara & Labrador, 2012). Ebenfalls nicht näher eingegangen wird auf Ansätze, die mittels Umgebungssensoren arbeiten (sog. »device-free«-Verfahren) und Aktivitäten beispielsweise durch Analyse von Videoaufzeichnungen erkennen (siehe z. B. J. K. Aggarwal & Xia, 2014; Ke et al., 2013).

Smartphonebasierte HAR-Systeme können in modernen Geräten auf eine Vielzahl von Sensoren zurückgreifen, welche unterschiedliche physikalische Größen messen (siehe Abschnitt 2.3.2). Je nach Benutzeraktivität und Geräteposition können somit verschiedene Bewegungsmuster erkannt und klassifiziert werden.

In nahezu allen HAR-Ansätzen wird das Accelerometer verwendet, mithilfe dessen sich anhand der gemessenen Beschleunigung bewegungsarme Aktivitäten (z. B. Stehen oder Sitzen) von bewegungsbetonteren Aktivitäten (z. B. Gehen oder Treppensteigen) unterscheiden lassen (siehe z. B. Bayat, Pomplun & Tran, 2014; Fan, Wang & Wang, 2013; Hassan, Uddin, Mohamed & Almogren, 2018). Um

zusätzlich Informationen über die Rotation des Geräts einfließen zu lassen, wird daneben häufig das Gyroskop verwendet (M. Ebner, Fetzer, Bullmann, Deinzer & Grzegorzek, 2020; Shoaib, Bosch, Incel, Scholten & Havinga, 2014), weitere Sensoren wie das Magnetometer oder Barometer kommen seltener zum Einsatz (J. Ye, Li, Zhang, Zhang & Chen, 2020; B. Zhou, Yang & Li, 2019).

Bei smartphonebasierten HAR-Systemen besteht im Vergleich zu anderen Methoden – wie etwa bei einem fest am Gürtel, am Fuß oder an der Brust montierten Sensor – die zusätzliche Herausforderung, dass das zur Aktivitätserkennung verwendete Gerät keine feste Position in Bezug auf die tragende Person hat. Je nach Trageart sind die Bewegungen des Smartphones mehr oder weniger von denen des Benutzers entkoppelt. Wird das Gerät frei schwingend in der Hand gehalten, so sind Beschleunigungs- und Richtungsänderungen nur indirekt oder verzögert vom Gerät registrierbar. Falls sich das Gerät in einer Tasche befindet, können Bewegungen der Arme und des Oberkörpers nur schwer erkannt werden.

Die meisten Arbeiten, in denen mehrere Gerätepositionen berücksichtigt wurden, unterscheiden zumindest zwischen dem Tragen des Gerätes in der Hand und in einer Tasche (M. Ebner et al., 2020; Radu & Marina, 2013). Zusätzlich werden oft am Gürtel befestigte (Murata, Hiroi, Kaji & Kawaguchi, 2014; Saeedi & El-Sheimy, 2015), in der Hand schwingende (Gu, Kealy, Khoshelham & Shang, 2015; Kasebzadeh, Hendeby, Fritsche, Gunnarsson & Gustafsson, 2017; Klein et al., 2018; Saeedi & El-Sheimy, 2015; Sun & Wang, 2017) oder am Kopf gehaltene Geräte untersucht (Klein et al., 2018; Saeedi & El-Sheimy, 2015; Sun & Wang, 2017). Seltener befindet sich das Gerät am Oberarm (Guo, Chen, Chen & Lv, 2016), in einem Rucksack (Kasebzadeh et al., 2017; Saeedi & El-Sheimy, 2015) oder einer Hemdtasche (Guo et al., 2016).

Für die Klassifikation der Geräteposition haben sich sowohl traditionelle ML-Verfahren als auch moderne Deep-Learning-Ansätze als sehr zuverlässig erwiesen (Almaslakh, Artoli & Al-Muhtadi, 2018; Fujinami & Kouchi, 2013). Um dem Umstand unterschiedlicher Gerätepositionen Rechnung zu tragen, werden später in diesem Kapitel u. a. auch Modelle trainiert und evaluiert, die auf spezifische Tragearten spezialisiert sind.

5.1.3 HAR in der Fußgängernavigation

Smartphonebasierte Verfahren wurden in der Literatur bereits erfolgreich zur Erkennung von Aktivitäten während der Fußgängernavigation eingesetzt. In der Frühzeit der Smartphone-Ära herrschten traditionelle Verfahren vor. H. Wang et al. (2012) nutzten einen Entscheidungsbaum und per Hand festgelegte Schwellenwerte und Features, um Gehen, Stehen, Treppensteigen, Aufzüge und Rolltreppen

zu unterscheiden. Als Eingabe dienten Messwerte des Accelerometers und Magnetometers. Die Evaluation zeigte eine Falschpositivrate von 0,2 % und eine Falschnegativrate von 1,1 %. Ein Entscheidungsbaum kam auch bei Susi, Renaudin und Lachapelle (2013) zum Einsatz, um zwischen einem stationären Benutzer, einem »quasi-stabilen« Gerät – dies schließt das Tragen des Geräts vor dem Körper oder in einer Tasche sowie das Telefonieren mit ein –, einem in der Hand schwingenden Gerät und irregulären Bewegungen zu differenzieren. Mithilfe von ausgewählten Features (Energie, Varianz sowie dominante Frequenzen von Gyroskop und Accelerometer) erreichten sie eine Klassifikationsgenauigkeit von 94 %. Ein weiteres Beispiel für traditionelle Methoden stammt von B. Zhou, Li, Mao, Tu und Zhang (2015). Auch hier wurde mithilfe eines Entscheidungsbaums zwischen Gehen, Stehen, Treppensteigen, Aufzügen und Rolltreppen (jeweils aufwärts/abwärts) unterschieden, mit dem Ziel, Aktivitätssequenzen zu erkennen, welche anschließend mit einem Umgebungsmodell abgeglichen werden können. Als Sensoren wurden Accelerometer, Gyroskop, Magnetometer und Barometer eingesetzt. Bei M. Ebner et al. (2020) findet sich ein aktuelles Beispiel für den Einsatz traditioneller Methoden. Anhand von Daten des Accelerometers und Gyroskops wurden die Aktivitäten Stehen, Gehen, Treppensteigen (aufwärts/abwärts) sowie sonstige, nicht zuordenbare Bewegungsmuster klassifiziert. Wie in der vorliegenden Arbeit wurden dabei die im realen Einsatz herrschenden Randbedingungen hinsichtlich der Latenz, Sensorik und Rechenleistung berücksichtigt.

Der Großteil der Forschung der letzten Jahre fokussiert sich auf Deep-Learning-Methoden. So nutzen beispielsweise J. Yang, Cheng, Chen, Zhou und Li (2018) und B. Zhou et al. (2019) in zwei aufeinander aufbauenden Arbeiten CNNs, um insgesamt neun verschiedene Aktivitäten mit einer Genauigkeit von 95 % resp. 98 % zu unterscheiden. J. Ye et al. (2020) führen eine Vergleichsstudie zwischen traditionellen Verfahren und Deep-Learning-Methoden durch, bei der die CNN- und LSTM-basierten Verfahren als Sieger hervorgingen. Bei Shavit und Klein (2021) wird die CNN-Architektur durch sog. *Transformer* (Vaswani et al., 2017) ergänzt, wodurch nochmals höhere Klassifikationsgenauigkeit erreicht wird.

Nach erfolgreicher Klassifikation von Geräteposition und Benutzeraktivität muss die somit gewonnene Information in das Navigations- bzw. Positionierungssystem integriert werden. Informationen über die Geräteposition erfüllen mehrere Zwecke: Zum einen kann dadurch Assistenz besser auf den aktuellen Kontext abgestimmt werden, beispielsweise durch das Anpassen der Modalität der Navigationsanweisungen. Ein Beispiel für haptische Anweisungen, die mithilfe von verschiedenen Vibrationsmustern die Richtung angeben, findet sich bei Pielot, Poppinga, Heuten und Boll (2012). Zum anderen dient die Geräteposition bei der In-

door-Positionierung zur genaueren Bestimmung der Bewegungsrichtung. Anstatt anzunehmen, dass das Gerät vor dem Körper grob in Gehrichtung gehalten wird, kann der Bewegungsvektor durch Analyse periodischer Bewegungsmuster und anschließender Transformation des lokalen Koordinatensystems geschätzt werden. Roy et al. (2014) zeigen dies in ihrem als *WalkCompass* bezeichneten System. Eine weitere Möglichkeit, wie Wissen um die Geräteposition hilfreich sein kann, demonstrieren Susi et al. (2013). Je nachdem, ob das Gerät stabil oder schwingend getragen wurde, ziehen sie einen anderen Sensor (Accelerometer bzw. Gyroskop) für die Schritterkennung heran.

Im Falle von Aktivitäten, die an bestimmte Orte gebunden sind, kann Unsicherheit bezüglich der Nutzerposition verringert werden. Ein Beispiel hierfür liefern Gu, Valae, Khoshelham, Shang und Zhang (2020), die einen sog. Landmarken-Graph definieren, der Informationen über die an verschiedenen Orten möglichen Aktivitäten enthält, und diesen anschließend in ein PDR-System integrieren. In anderen Arbeiten wurden für ähnliche Problemstellungen bereits Partikelfilter (Radu & Marina, 2013) und Hidden-Markov-Modelle (Bobkov, Grimm, Steinbach, Hilsenbeck & Schroth, 2015; Torok, Nagy, Kovats & Pach, 2014; B. Zhou et al., 2015) eingesetzt.

Aufbauend auf H. Wang et al. (2012) stellen Chen et al. (2015) ein System vor, in dem PDR und WLAN mithilfe eines Kalman-Filters fusioniert werden. Zusätzlich wird die Positionierung mithilfe von durch die Sensorik (Accelerometer, Magnetometer, Gyroskop, Barometer und WLAN) registrierten Landmarken stabilisiert. Diese bestehen aus Abbiegungen, Aufzügen, (Roll-)Treppen und Türen, wobei jeweils unterschiedliche Verfahren für die Erkennung aus den Sensor-Rohdaten zum Einsatz kommen, ohne dass dabei jedoch das genaue Vorgehen beschrieben wird. Insbesondere die erwähnte Erkennung von Türdurchgängen anhand der WLAN-Signalstärke erscheint aufgrund der Einschränkungen hinsichtlich der Scan-Frequenz nur schwer umsetzbar.

Ähnlich wie H. Wang et al. (2012) verwenden auch Shang, Gu, Hu und Kealy (2015) zwei Arten von Landmarken: *Seed Landmarks* sind im Voraus anhand der Gebäudegeometrie bekannte Orte wie Aufzüge und Treppen, welche anhand spezifischer Bewegungsmuster erkennbar sind. *Organic Landmarks* hingegen werden mithilfe von Crowd-Sourcing anhand von Anomalien der Sensordaten, beispielsweise des Magnetfelds, erhoben. Die Positionsverfolgung findet mithilfe eines Partikelfilters statt, wobei die Distanz zu den Landmarken für die Gewichtung der Partikel herangezogen wird.

Elhoushi, Georgy, Noureldin und Korenberg (2016, S. 210 f.) schlagen eine regelbasierte Strategie vor, wonach je nach erkannter Bewegungsform unterschiedliche Komponenten des Positionierungssystems aktiv sind. So wird beispielsweise

beim Aufzugfahren nur die Höheninformation aktualisiert, während die 2D-Position unverändert bleibt. Die Autoren gehen auch auf verschiedene Möglichkeiten von Fehlklassifizierungen ein, um etwa die Implikationen von nicht erkannten Treppen, Rolltreppen etc. abzuschätzen.

Fetzer et al. (2018) zeigen eine Anwendung von HAR für die Erkennung von Stockwerkswechseln bei der Navigation in einem historischen Gebäude. Die Einbindung in das partikelfilterbasierte Positionierungssystem geschieht – abhängig von der aktuellen Position und der erkannten Aktivität – durch Manipulation der Partikelgewichte.

Ein aktuelles Beispiel für den Einsatz von HAR in der Indoor-Positionierung findet sich bei Moreira et al. (2021), welche in ihrem System PDR- und WLAN-Ortung mithilfe eines Partikelfilters fusionieren. Auch hier werden die Partikel anhand der HAR-Information gewichtet, um v. a. bei Stockwerkswechseln eine Positionskorrektur zu erreichen. Die Autoren zeigen, dass die Positionierungsgenauigkeit bei Routen mit Stockwerkswechseln gegenüber einer rein auf dem Barometer beruhenden Lösung durch zusätzliche Anwendung von HAR steigt.

Der Überblick zeigt, dass zu HAR nicht nur im Allgemeinen, sondern auch mit Bezug zum Forschungsfeld (Indoor-)Fußgängernavigation eine große Menge an Literatur existiert. Weit weniger häufig wird jedoch die konkrete Integration in Indoor-Positionierungsverfahren thematisiert; die vorliegende Arbeit leistet an dieser Stelle einen wichtigen Beitrag (siehe Abschnitt 5.4.1).

5.1.4 Existierende HAR-Datensätze

Dieser Abschnitt gibt einen kurzen Überblick über bereits in der Literatur veröffentlichte frei verfügbare HAR-Datensätze und begründet anschließend die Notwendigkeit, für die vorliegende Arbeit einen eigenen Datensatz zu erheben.

Eine aktuelle Übersicht über viele HAR-Datensätze und deren Eigenschaften liefern De-La-Hoz-Franco, Ariza-Colpas, Quero und Espinilla (2018) und J. Wang et al. (2019, S. 6). Für das vorliegende Problem nicht anwendbar sind alle Datensätze, die mithilfe von zusätzlichen Sensoren – beispielsweise am Fuß oder Gürtel befestigt – erhoben wurden. Viele der Datensätze sind auf den Forschungsbereich ADL zugeschnitten und enthalten lediglich generische Klassen wie etwa Gehen, Stehen oder Sitzen. Für die Fußgängernavigation im Innenbereich sind jedoch vor allem solche Aktivitäten von Interesse, die Rückschlüsse auf den Standort des Benutzers zulassen, wie etwa das Treppensteigen oder Aufzugfahren. Auch verschiedene stationäre Verhaltensweisen können Informationen liefern, mithilfe derer die Assistenz verbessert werden kann. Andere Aktivitäten wiederum – beispielsweise das Sitzen, Liegen oder auch Joggen – sind für die Fußgängernavigation nur von

geringer Relevanz.

Einige verbreitete Datensätze aus der Literatur sollen an dieser Stelle kurz vorgestellt werden: Der Opportunity-Datensatz (Roggen et al., 2010) wurde mithilfe einer Vielzahl von in der Umgebung, an Objekten und am Körper von Testpersonen befestigten Sensoren aufgezeichnet, darunter auch Inertialsensoren. Der Fokus lag jedoch auf Aktivitäten des täglichen Lebens, u. a. der Zubereitung von Lebensmitteln. Auch der WISDM-Datensatz (Kwapisz, Weiss & Moore, 2011) adressiert die Fußgängernavigation nicht explizit, sondern enthält allgemeine Aktivitäten (Gehen, Joggen, Treppensteigen, Sitzen, Stehen). Für die Datenaufzeichnung wurde ein in der Tasche getragenes Smartphone verwendet, von welchem Beschleunigungsdaten mit einer Samplingrate von 20 Hz erfasst wurden. Ähnlich verhält es sich beim UCI-HAR-Datensatz (Anguita, Ghio, Oneto, Parra & Reyes-Ortiz, 2013), welcher die Aktivitäten Gehen, Stehen, Sitzen, Liegen und Treppensteigen enthält. In zwei Experimentdurchläufen wurde das zur Datenaufzeichnung verwendete Smartphone zunächst am Gürtel befestigt, dann nach Belieben gehalten. Messwerte von Gyroskop und Accelerometer wurden mit einer Samplingrate von 50 Hz erfasst. In einer Erweiterung des Datensatzes sind zusätzlich die Übergänge zwischen den Aktivitäten enthalten (Reyes-Ortiz et al., 2014).

Aktuelle Bemühungen von Moreira et al. (2021) zeigen, dass nach wie vor kein geeigneter HAR-Datensatz für die Fußgängernavigation existiert: Sie erheben Daten von Accelerometer, Gyroskop und Magnetfeldsensor in zwei Gerätepositionen (»Texting« und »Calling«) mit einer je nach Testgerät schwankenden Samplingrate von 50 bis 200 Hz. Dabei wird zwischen stationärem Verhalten, Gehen, Laufen sowie drei Varianten von Stockwerkswechseln (Aufzüge, Treppen, Rampen) unterschieden.

Alle soeben aufgezählten Datensätze erfüllen nur einen Teil der für diese Arbeit aufgestellten Kriterien. Das Hauptproblem ist in vielen Fällen der Fokus auf ADL, weswegen nur wenige der enthaltenen Aktivitäten bei Anwendung in einem Fußgängernavigationssystem Rückschlüsse auf die Nutzerposition ermöglichen. Weiterhin werden bei smartphonebasierten Datensätzen nur selten mehrere Gerätepositionen berücksichtigt. Aus diesen Gründen wird im Folgenden ein spezifisch für die Indoor-Positionierung zugeschnittener HAR-Datensatz erhoben.

5.2 Erkennung von navigationsspezifischen Aktivitäten

Wie der vorige Abschnitt zeigte, ist nur ein Teil der im breiten Forschungsfeld HAR verfolgten Ansätze auf die smartphonebasierte Fußgängernavigation bzw.

die Indoor-Positionierung anwendbar. Im weiteren Verlauf dieses Abschnitts wird daher ein HAR-Verfahren vorgestellt, das sich in mehreren Aspekten von existierenden Ansätzen unterscheidet. Basierend auf den Erkenntnissen der naturalistischen Studie zum Nutzerverhalten während der Navigation werden mehrere verschiedene Geräte-Tragearten berücksichtigt. Um möglichst viele für die Indoor-Positionierung relevante Informationen zu gewinnen, werden die Aktivitäten feiner aufgegliedert. Unter anderem wird im Folgenden untersucht, wie zuverlässig zwischen verschiedenen stationären Verhaltensweisen (still stehen vs. sich aktiv umschauchen) und Fortbewegungsarten (geradeaus gehen, umkehren, links/rechts abbiegen, durch Türen gehen, Treppensteigen) unterschieden werden kann. Ein hierarchischer Aufbau sorgt dafür, dass nicht alle diese Klassen von einem einzigen Klassifikator erkannt werden müssen. Das Konzept sieht zunächst eine grobe Unterscheidung von stationärem Verhalten und Fortbewegung vor, woraufhin sich in einer Baumstruktur angeordnete Klassifikatoren spezifischen Klassifikationsproblemen widmen können. Optional erfolgt vorangeschaltet die Bestimmung des Tragemodus, wodurch der Einsatz von weiteren spezialisierten Klassifikatoren ermöglicht wird.

5.2.1 Aktivitäten und Gerätepositionen

Basierend auf den Erkenntnissen der naturalistischen Studie und ausgehend von der Anforderung, feingliedrige Aktivitäten zu bestimmen, die für die Indoor-Positionierung nützliche Informationen enthalten, werden in dieser Arbeit neun verschiedene Klassen unterschieden. Die nachfolgend verwendeten Bezeichnungen dienen im weiteren Verlauf für den eindeutigen Verweis auf die Klassen.

- **StillStehen**
Der erste stationäre Modus tritt dann auf, wenn BenutzerInnen still stehen und keine anderweitigen Bewegungen vornehmen, beispielsweise beim Lesen von Anweisungen auf dem Smartphone oder von Wegweisern in der Umgebung.
- **Umschauen**
Der zweite stationäre Modus zeichnet sich – obwohl nach wie vor keine Fortbewegung stattfindet – durch höhere physische Aktivität aus, etwa beim aktiven Suchen nach einer referenzierten Landmarke.
- **GeradeausGehen**
Wie in HAR-Datensätzen üblich, ist auch hier eine Klasse enthalten, die reguläre Gehbewegungen in der Ebene repräsentiert.

- **LinksAbbiegen und RechtsAbbiegen**
Richtungsänderungen wurden zuvor mithilfe des GRV bestimmt. Je nach Geräteposition ist dies jedoch nur bedingt regelbasiert möglich, weswegen die beiden Aktivitäten mit aufgenommen wurden.
- **TreppeAuf und TreppeAb**
Das Treppensteigen ist im vorliegenden Anwendungsfall die vorherrschende Form des Stockwerkswechsels. Im Hinblick auf die Positionsbestimmung bieten korrekt erkannte Treppen eine gute Gelegenheit zur Positionskorrektur und somit eine neue PDR-Startposition nach Stockwerksübergängen.
- **TürDrücken und TürZiehen**
Das Passieren von Türen wurde in der Literatur bisher nur selten berücksichtigt und stellt einen wichtigen Beitrag in dieser Arbeit dar. Ebenso wie Stockwerkswechsel sind auch Türdurchgänge eine Gelegenheit zum Abbauen von Unsicherheit in der Positionsschätzung. Die Analyse in Abschnitt 4.4.1 zeigte zudem, dass diese in den real abgerufenen Routen deutlich häufiger vorkommen. Die Unterscheidung der beiden Varianten kann in Fällen, in denen sich Türen nur in eine Richtung öffnen lassen, zusätzliche Informationen über den Ort bzw. die Bewegungsrichtung des Benutzers liefern.

Diese Klassen decken einen Großteil der erwarteten Aktivitäten während der Fußgängernavigation auf dem Regensburger Universitätscampus ab. Im Vergleich zu einigen der in Abschnitt 5.1.4 beschriebenen Datensätze fehlen beispielsweise Stockwerkswechsel anhand von Rolltreppen, Aufzügen oder Rampen. Während diese in anderen Gebäudekomplexen sicherlich eine größere Rolle spielen, sind sie im vorliegenden Fall entweder nicht vorhanden (Rolltreppen) oder selten in Routen enthalten (Aufzüge, Rampen) und wurden daher nicht berücksichtigt.

Ausgehend von den Erkenntnissen der naturalistischen Studie und in Anlehnung an die Literatur werden die folgenden vier Gerätepositionen berücksichtigt:

- **In der Hand, vor dem Körper (vorne)**
Das Gerät wird stabil vor dem Körper gehalten, grob in Gehrichtung zeigend, mit dem Bildschirm nach oben gerichtet. BenutzerInnen können in diesem Modus mit dem Gerät interagieren und auf neue visuelle Anweisungen reagieren. Das Positionierungssystem kann die Geräterichtung unmittelbar zur Bestimmung der Gehrichtung heranziehen.
- **In der Hand, schwingend (schwingend)**
Das Gerät wird in der Hand gehalten, aber nicht stabil vor dem Körper, sondern an der Seite baumelnd. Dadurch ist das Gerät stärker von den Bewegungen der Person entkoppelt und erfährt potenziell höhere Beschleunigungen.

Die Korrelation zwischen der Bewegung des Geräts und der Gehrichtung ist dabei deutlich geringer.

- Im Telefonat (Telefonat)
Das Gerät wird nach oben zeigend ans Ohr gehalten und ist somit an die Bewegungen des Kopfes gekoppelt.
- In der Tasche (Tasche)
Das Gerät befindet sich in einer der Taschen der Person. Gehbewegungen sollten in dieser Position deutlich messbar sein.

Diese vier Positionen decken einen Großteil der in der Literatur verwendeten Klassen ab. Durch Kombination jeder der neun Aktivitäten und vier Gerätepositionen ergeben sich schlussendlich 36 verschiedene Klassen.

5.2.2 Erhebung des HAR-Datensatzes

Die Erhebung der Daten wurde von drei studentischen bzw. wissenschaftlichen Hilfskräften im Verlauf mehrerer Wochen Ende 2019/Anfang 2020 durchgeführt. Zum Zweck der Evaluation im Rahmen dieser Arbeit wurde von mir als Autor ein separater Testdatensatz aufgezeichnet. Als Geräte dienten wie im vorigen Experiment jeweils Google-Pixel-Smartphones.

5.2.2.1 Die Datenerhebungs-App

Für die Erhebung des Datensatzes wurde eine einfache App entwickelt, welche die Auswahl von Aktivität und Geräteposition ermöglicht und eine entsprechende Anweisung anzeigt (siehe Tabelle 5.1). Die Benutzeroberfläche der App ist einfach gehalten und besteht neben den Auswahloptionen für Aktivität und Position aus einer Schaltfläche, mithilfe derer die Aufzeichnung der Sensordaten gestartet werden kann. Bei laufender Datenaufzeichnung besteht die Option, diese im Falle von Fehlern oder Problemen abzubrechen. Wird die Aufzeichnung stattdessen erfolgreich abgeschlossen, aktualisiert sich als Rückmeldung für die Testperson ein für die ausgewählte Kombination aus Aktivität und Geräteposition spezifischer Zähler.

Für jede Kombination aus Aktivität und Geräteposition wurden pro Teilnehmer mindestens 20 Läufe aufgezeichnet, wobei keine Einschränkungen hinsichtlich des genauen Ortes festgelegt wurden. Ebenfalls wurde die exakte Geräteposition (z. B. in welcher Tasche oder in welcher Hand das Gerät zu tragen ist) nicht näher spezifiziert, um eine möglichst realistische Stichprobe zu ziehen.

Aktivität	Anweisung
StillStehen	Bitte still stehen bleiben. (z. B. um ein Schild oder Plakat zu lesen)
Umschauen	Bitte stehen bleiben und suchend in der Umgebung umschauen. (z. B. auf der Suche nach einer Landmarke)
GeradeausGehen	Bitte geradeaus (weiter) gehen.
LinksAbbiegen	Bitte jetzt links abbiegen.
RechtsAbbiegen	Bitte jetzt rechts abbiegen.
TreppeAuf	Bitte jetzt die Treppe hoch gehen.
TreppeAb	Bitte jetzt die Treppe runter gehen.
TürDrücken	Bitte jetzt durch die Tür gehen. (Tür drücken)
TürZiehen	Bitte jetzt durch die Tür gehen. (Tür ziehen)

Tabelle 5.1: Aktivitäten und dazugehörige Anweisungen

5.2.2.2 Verwendete Sensoren

Der Datensatz besteht aus neun Kanälen bzw. Features von verschiedenen Sensoren. Diese sind wie folgt:

- Vom **linearen Accelerometer** sind Daten aller drei Achsen enthalten. Sie geben Aufschluss über die relative Bewegung des Geräts im Raum.
- Ebenso ist das **Gyroskop** mit allen drei Achsen vertreten. Mit ihnen sind Rotationen des Geräts und somit auch Änderungen in der Geräteausrichtung identifizierbar.
- Um Informationen über die absolute Ausrichtung des Geräts in globaler z-Richtung (d. h. die Neigung nach oben/unten) zu gewinnen, ist die entsprechende Achse des **Game-Rotationsvektors** enthalten. Die x- und y-Abschnitte des Vektors bleiben bewusst unberücksichtigt, um eine vom Azimut des Geräts und somit der globalen Bewegungsrichtung unabhängige Klassifikation zu ermöglichen.
- Aus demselben Grund ist vom **Magnetfeldsensor** anstelle der einzelnen Achsen lediglich die Magnitude enthalten. Um zu vermeiden, dass standortbezogene absolute Magnetfeld-Magnituden gelernt werden, wird jeweils die Differenz zum ersten Wert der Zeitreihe gebildet. Auf diese Weise kann der Klassifikator gezielt auf Änderungen in der Magnetfeldstärke trainiert werden.

- Das **Barometer** dient zur Erkennung von Höhenänderungen (z. B. beim Treppensteigen). Da der Luftdruck je nach Örtlichkeit und Wetterlage variiert, ist auch hier wie beim Magnetfeldsensor die Differenz zum jeweils ersten Wert enthalten.
- Nur für die Vorverarbeitung des Datensatzes wurden zusätzlich Messwerte des **Annäherungssensors** aufgezeichnet.

5.2.2.3 Vorverarbeitung des Datensatzes

Vor der Verwendung des Datensatzes mussten einige Vorverarbeitungsschritte vorgenommen werden. Zunächst wurden alle abgebrochenen sowie andere offensichtlich fehlerhaften Aufzeichnungen entfernt. Dies umfasst beispielsweise aufgrund von technischen Fehlern sehr kurze oder lange sowie im Einzelfall zeitlich überlappende Aufzeichnungen. Insgesamt wurden so 27 Einzelaufzeichnungen entfernt.

Mit Ausnahme der Geräteposition vorne mussten die Testpersonen zu Beginn jeder Aufzeichnung vor Ausübung der gewählten Aktivität das Gerät in die richtige Position bringen, z. B. in die Tasche stecken. Analog dazu musste das Gerät am Ende wieder kurz bedient werden, um die Aufzeichnung zu beenden. Um diesem Umstand Rechnung zu tragen, wurden für die Gerätepositionen Telefonat und Tasche anhand des Annäherungssensors die Zeiträume bestimmt, in denen sich das Gerät in der intendierten Position befand. Anschließend wurde für alle Aufzeichnungen pauschal am Anfang und Ende ein Intervall von 500 ms entfernt, um sicherzustellen, dass möglichst keine durch die Handhabung des Geräts verursachten Artefakte enthalten sind.

Schließlich wurden die Daten der verschiedenen Sensoren anhand ihrer Zeitstempel ausgerichtet und – da die Sampling-Rate teilweise variierte (Gyroskop, Accelerometer und GRV wurden mit 100 Hz, der Magnetfeldsensor mit 50 Hz, das Barometer mit ca. 10 Hz aufgezeichnet) – per Upsampling aneinander angeglichen. Zuletzt wurden mehrere Datensätze für unterschiedliche Fenstergrößen (1, 2 und 3 s respektive 100, 200 und 300 Datenpunkte bei 100 Hz) erstellt. Der resultierende Trainingsdatensatz enthält mehr als sechs Stunden an Daten. Für den separaten Testdatensatz, in dem pro Kombination aus Aktivität und Geräteposition sechs Aufzeichnungen enthalten sind – der Umfang beträgt demnach etwa ein Zehntel des Trainingsdatensatzes –, wurden die identischen Vorverarbeitungsschritte durchgeführt.

In Abbildung 5.1 sind beispielhaft die Sensor-Rohdaten vier verschiedener Aktivitäten bei jeweils unterschiedlicher Geräteposition zu sehen. Deutlich erkennbar sind hier beispielsweise die starken periodischen Ausschläge beim Treppensteigen,

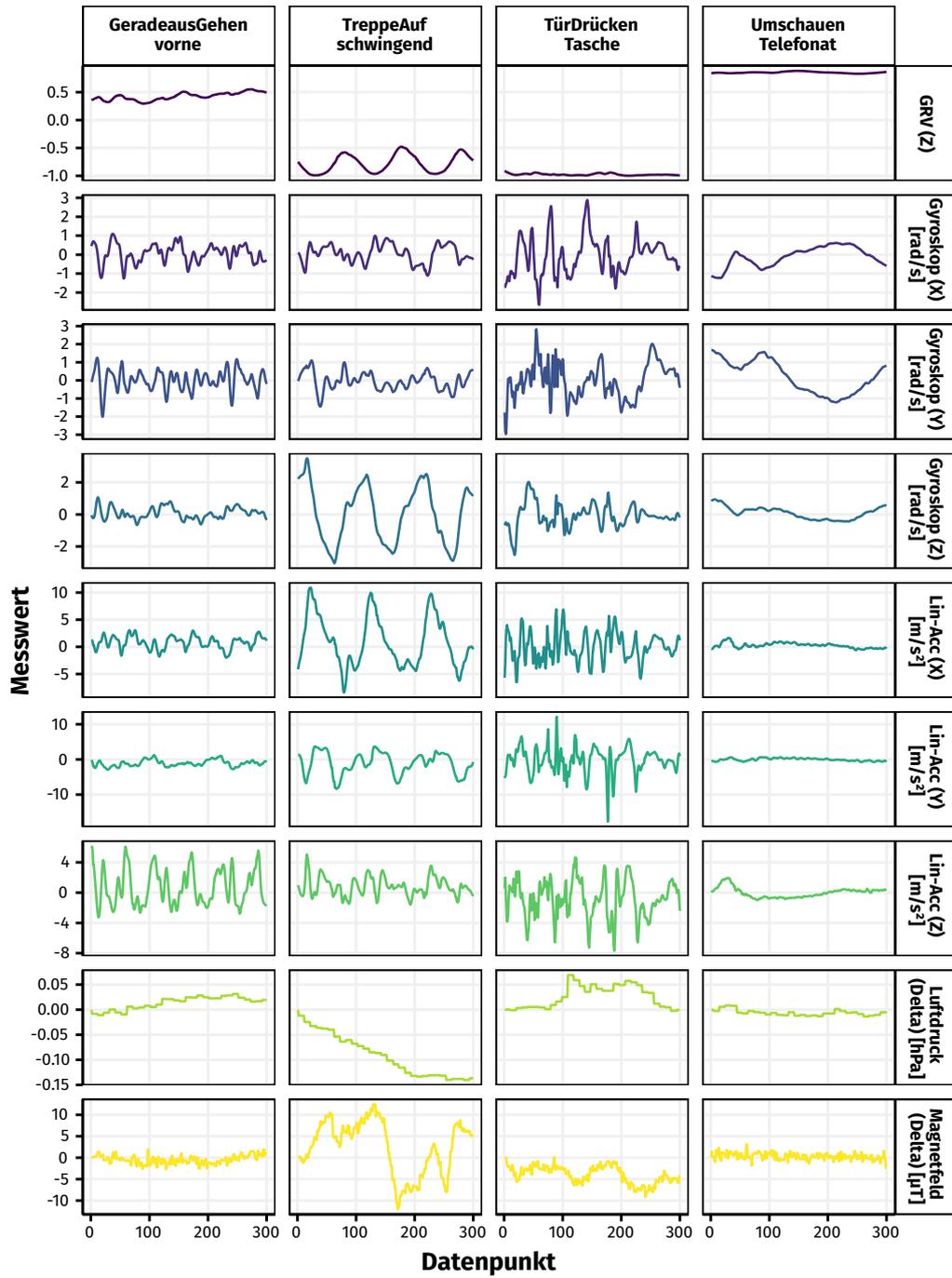


Abbildung 5.1: Beispielkombinationen mit unterschiedlicher Charakteristik aus dem Aktivitäts-Datensatz (jeweils 3-s-Fenster)

die irreguläre Bewegung beim Passieren einer Tür sowie die geringen Veränderungen bei stationärem Verhalten.

5.2.3 Machine-Learning-Methodik

Gemäß dem in Abschnitt 5.1 beschriebenen Stand der Technik kommen zur Erkennung der Aktivitäten moderne Deep-Learning-Verfahren zum Einsatz. Im Folgenden werden die evaluierten Modelle, der Trainings- und Evaluationsprozess sowie die Klassifikator-Hierarchie näher beschrieben.

5.2.3.1 Modelle

Im Fokus dieser Arbeit steht weder die Neuentwicklung von Deep-Learning-Modellarchitekturen noch der Vergleich zu traditionellen Machine-Learning-Methoden, weswegen größtenteils existierende State-of-the-Art-Architekturen für die Verarbeitung von Zeitreihen-Daten verwendet werden. Im Einzelnen sind dies die aus Abschnitt 5.1.1 bekannten Modelle CNN, DeepConvLSTM, ResNet und InceptionTime, die von van Kuppevelt et al. (2020) in der Python-Bibliothek *mcfly* zur Verfügung gestellt werden. Zusätzlich wurde für diese Arbeit eine vereinfachte Variante der DeepConvLSTM-Architektur geschaffen, die auf den vorgeschalteten CNN-Teil verzichtet. Das Resultat ist ein kompakteres Modell, welches sich hinsichtlich Speicherbedarf und Rechenaufwand besser für mobile Geräte eignet. Die Details der Modellarchitektur sind in Tabelle 5.2 zusammengefasst.

Layer	Output Shape	Param #
BatchNormalization	(None, 100, 9)	36
LSTM	(None, 100, 32)	5376
LSTM	(None, 100, 32)	8320
Dropout	(None, 100, 32)	0
TimeDistributed	(None, 100, 4)	132
Activation	(None, 100, 4)	0
Lambda	(None, 4)	0

Tabelle 5.2: Architektur des LSTM-basierten Modells
(hier: 2 LSTM-Schichten mit Breite 32, Fenstergröße 100)

Das *mcfly*-Framework ermöglicht eine einfache Form der Hyperparameter-Optimierung, bei der mehrere Modelle mit innerhalb eines vorgegebenen Intervalls zufällig gewählten Parametern erzeugt und trainiert werden. Da sich einige der Hyperparameter unmittelbar auf die Tiefe und Breite der Modelle auswirken, wurde zunächst in einer Vorstudie der resultierende Speicherbedarf sowie die Laufzeit

auf einem Android-Smartphone ermittelt. Für den Live-Betrieb in einer Navigations-App sollte eine Klassifikation nach Möglichkeit nicht länger als ca. 100 ms dauern, sodass bei Bedarf auch mehrere Modelle hintereinander geschaltet werden können, ohne eine signifikante Verzögerung zu bewirken. Der Speicherplatzbedarf ist bei aktuellen Geräten mit mehreren GB Arbeitsspeicher und dutzenden GB Speicherplatz für Apps und Daten in der Regel kein einschränkender Faktor. Da die Modelle im Praxiseinsatz mit einer Navigations-App verteilt und im Falle von Verbesserungen per Download aktualisiert werden müssen, ist dennoch eine möglichst geringe Größe wünschenswert; als Richtwert werden 5 MB pro Modell angestrebt.

Um in der Vorstudie die Auswirkungen der Hyperparameter zu testen, wurden jene gezielt variiert, um unterschiedlich tiefe Modelle zu erzeugen. Nach einer kurzen Trainingsphase erfolgte die Konvertierung in das speziell für Mobilgeräte optimierte TensorFlow-Lite-Format. Mithilfe des TFLite Model Benchmark Tools (Abadi et al., 2015) wurden alle Modelle mehrfach auf einem Google Pixel ausgeführt, um nach zehn Durchläufen die mittlere Laufzeit eines einzelnen Inferenz-Vorgangs zu bestimmen.

Das Ergebnis zeigte, dass mit den standardmäßig voreingestellten Hyperparameter-Grenzwerten nur wenige Modelle die gesteckten Vorgaben erreichen. Als am langsamsten erwies sich die DeepConvLSTM-Architektur, die bei hohen Filterzahlen bis zu 1146 ms für einen Durchlauf benötigt. Die Größe der Modelle betrug dabei maximal 4,1 MB. Deutlich größer waren CNN (ca. 40 MB) und ResNet (12–63 MB), wobei ersteres mit kurzen Laufzeiten von 22–98 ms punkten konnte (ResNet: 124–451 ms). Insgesamt am besten schnitten InceptionTime sowie das reine LSTM-Modell ab. Sie erreichten Laufzeiten von 154–284 ms (InceptionTime) und 30–200 ms (LSTM) bei einer Größe von maximal 5,2 resp. 1,6 MB.

Um die Zielwerte zu erreichen, wurden die Modelle anschließend hinsichtlich Breite und Tiefe eingeschränkt. Das CNN besteht nun aus maximal drei (statt zehn) Schichten mit maximal 32 (statt 100) Filtern sowie maximal 256 (statt 2000) Neuronen in den vollvernetzten Schichten. In den beiden LSTM-basierten Modellen sind nun maximal jeweils zwei (statt bis zu fünf) LSTM-Schichten gestapelt, mit reduzierter Breite (32 statt 100). Um InceptionTime und ResNet für den mobilen Einsatz zu adaptieren, wurde deren Tiefe auf den minimalen Wert (drei Inception- bzw. zwei ResNet-Module) festgesetzt sowie die Breite in Form von Filteranzahl und Kernelgröße reduziert. Ein weiterer kurzer Test mit diesen Parametern zeigte eine maximale Modellgröße von 2,1 MB. Hinsichtlich der Laufzeit überschritten lediglich manche InceptionTime-Modelle den Zielwert von 100 ms knapp.

5.2.3.2 Training und Evaluation

Da die Anzahl der Daten pro Klasse nicht gleichverteilt ist und da je nach spezifischer Konfiguration mehrere Klassen kombiniert werden, wurde vor dem Training zunächst jeweils mittels zufälligem Undersampling für gleich große Klassen gesorgt. Um robuste Ergebnisse zu erhalten, wurde der Trainingsdatensatz mittels fünffacher stratifizierter Kreuzvalidierung in Trainings- und Validierungsdaten geteilt. Für jede der Modellarchitekturen fand anschließend im ersten Durchlauf der Kreuzvalidierung die Hyperparameteroptimierung statt. Hierfür wurden 25 Modelle erzeugt und jeweils 50 Epochen lang an einer Teilmenge des Datensatzes trainiert. Dasjenige Modell mit der höchsten Validierungsgenauigkeit wurde anschließend mit dem kompletten Trainingsdatensatz weitertrainiert; seine Hyperparameter wurden für die verbleibenden Kreuzvalidierungsdurchläufe übernommen. Algorithmus 5.1 fasst den Trainingsprozess zusammen.

Algorithmus 5.1: Ablauf des Trainingsprozesses

Für alle Fenstergrößen w (1,2,3 s) :
Lade Datensatz d für w
Erzeuge d' mit balancierten Klassen durch Undersampling
Für alle Modellarchitekturen :
 Für 5 Folds :
 Splitte d' in Trainings- und Validationsdatensatz
 Falls Fold 1 :
 Erzeuge 25 Modelle mit unterschiedlichen Hyperparametern
 Trainiere Modelle (50 Epochen, Teil des Datensatzes)
 Wähle bestes Modell m aus
 Ansonsten
 Erzeuge Modell m mit zuvor bestimmten Hyperparametern
 Trainiere m (250 Epochen, ganzer Datensatz)
 Speichere m für die Evaluation

Für die abschließende Bestimmung der Klassifikations-Performance fand eine Evaluation am separaten Testdatensatz statt, womit die in der Realität auftretende Situation nachgebildet wird, in der die Aktivitäten eines bislang unbekanntem Benutzers erkannt werden müssen. Die Treffergenauigkeit (Accuracy) wurde dort mittels Mehrheitsvotum der aus der Kreuzvalidierung hervorgegangenen Modelle bestimmt. Zusätzlich wird im Folgenden das gemittelte, ungewichtete F_1 -Maß angegeben. Dies folgt der Argumentation von M. Ebner et al. (2020, S. 15), wonach es in diesem Anwendungsfall eher von Interesse ist, möglichst alle Aktivitäten gut zu erkennen, anstatt die meiste Zeit richtig zu liegen. In manchen Fällen – immer dann, wenn ein Klassifikator eine der Klassen fälschlicherweise kein einziges Mal

vorhersagt oder wenn für eine der Klassen Precision und Recall gleich Null sind – ist das F_1 -Maß nicht definiert.

Für die Evaluation der kompletten Klassifikatorhierarchie kommt eine speziell dafür angepasste Variante des F_1 -Maßes zum Einsatz, die auch teilweise korrekte Klassifikationen berücksichtigt (Kiritchenko, Matwin & Famili, 2005). Für die Berechnung werden folgende Definitionen von Precision und Recall verwendet:

$$hP = \frac{\sum_i |C_i \cap C'_i|}{\sum_i |C'_i|} \quad (5.1)$$

$$hR = \frac{\sum_i |C_i \cap C'_i|}{\sum_i |C_i|} \quad (5.2)$$

C_i bezeichnet hier die Menge der tatsächlichen Klasse und ihrer Vorfahren in der Hierarchie, während C'_i die Prädiktion und ihre Vorfahren beinhaltet. Das hierarchische F_1 -Maß berechnet sich anschließend auf die herkömmliche Weise:

$$hF_1 = \frac{2 \cdot hP \cdot hR}{hP + hR} \quad (5.3)$$

5.2.3.3 Die Klassifikator-Hierarchie

In der Praxis müssen für die Aufrechterhaltung der Positionsbestimmung nicht zwangsweise alle neun in Abschnitt 5.2.1 beschriebenen Aktivitäten unterschieden werden, sondern es ist ausreichend, zu wissen, ob beispielsweise eine Tür passiert wurde oder ob der Benutzer stehen geblieben ist. Wie eingangs erwähnt, wird daher im Folgenden auch ein hierarchischer Ansatz erprobt, der durch das bedingte Hintereinander-Ausführen von Klassifikatoren realisiert wird.

Im nächsten Abschnitt wird zunächst ein Klassifikator für die Geräteposition evaluiert, welcher optional der weiteren Hierarchie vorangestellt werden kann (im Folgenden als K_0 bezeichnet). Als herkömmlicher Ansatz werden anschließend einzelne, flache Klassifikatoren für alle neun Aktivitäten getestet (K_1 , siehe Abbildung 5.2). Darauf folgt eine zweistufige Anordnung, bei der zunächst zwischen den vier groben Klassen *Stehen/Gehen/Tür/Treppe* unterschieden wird (K_2), gefolgt von jeweils zwei oder drei Subklassen (K_{3-6} , siehe Abbildung 5.3).

5.2.4 Evaluation der HAR-Modelle

In diesem Abschnitt werden die Evaluationsergebnisse der im vorigen Abschnitt beschriebenen Modelle präsentiert. Wie in Abschnitt 5.2.3.2 beschrieben, werden als Metriken die Genauigkeit (*Accuracy*) und das ungewichtete F_1 -Maß angegeben.

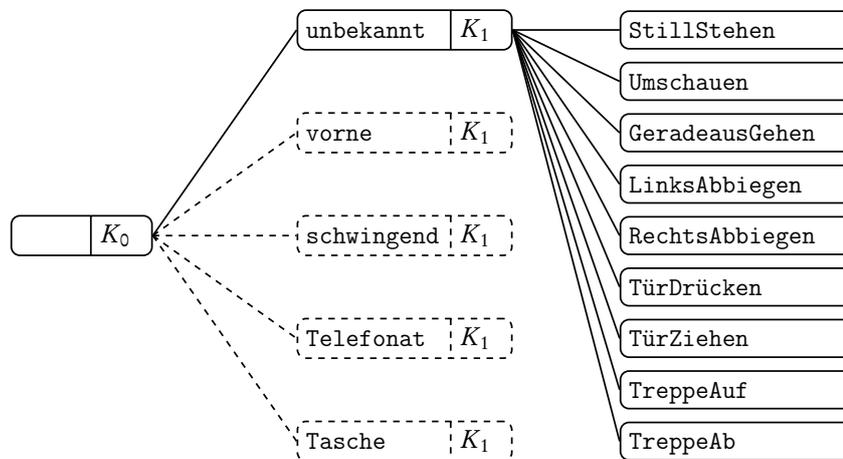


Abbildung 5.2: Klassifikator-Setup mit neun Klassen und optional vorangestelltem Gerätepositions-Klassifikator

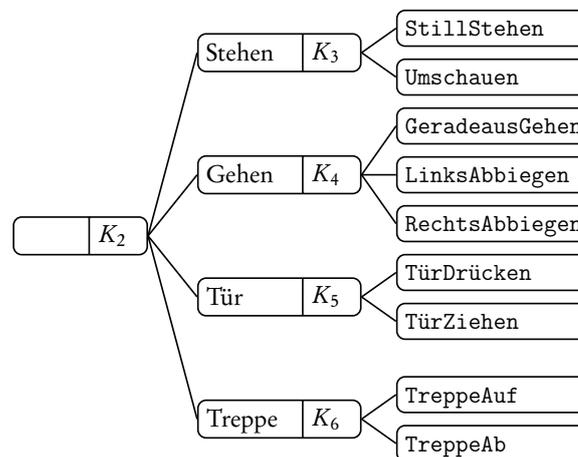


Abbildung 5.3: Zweistufige Klassifikator-Hierarchie mit vier groben Aktivitätsklassen und jeweils zwei/drei Unterklassen

Für Detailbetrachtungen der performantesten Modelle wird die Konfusionsmatrix herangezogen.

5.2.4.1 Bestimmung der Geräteposition

Das Bestimmen der Geräteposition ist Voraussetzung dafür, gezielt trainierte Modelle für die Aktivitätserkennung einzusetzen. Wie aus Abbildung 5.4 sowie Tabelle 5.3 hervorgeht, ist dies in den meisten Fällen äußerst zuverlässig möglich. Mehrere Modelle klassifizieren nahezu alle Datenpunkte im Testdatensatz korrekt, DeepConvLSTM bei einer Fenstergröße von 2 s erreicht gar eine Genauigkeit von 100 %. Abgesehen von zwei Ausreißern liegen alle Modelle bei mehr als 97 % Genauigkeit und weisen keine deutlichen Unterschiede hinsichtlich der Fenstergröße auf.

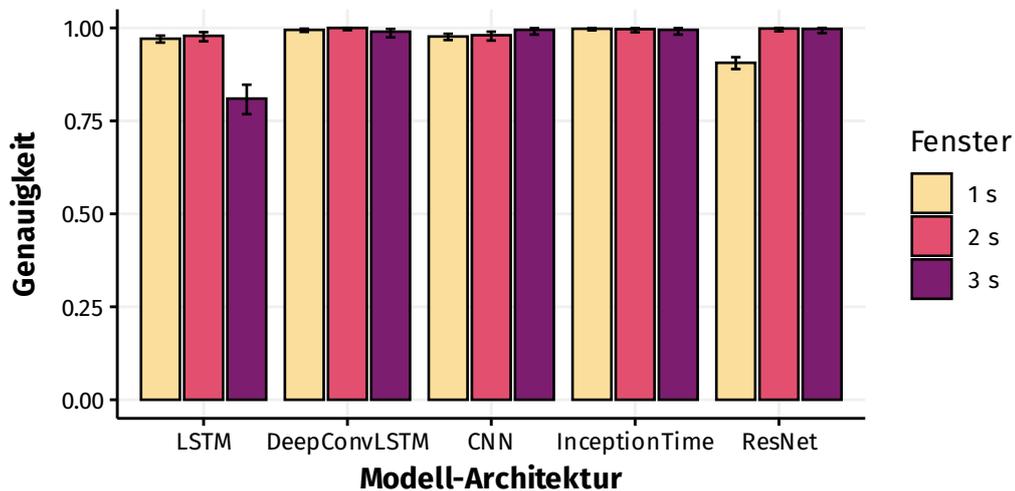


Abbildung 5.4: Treffergenauigkeit der Geräteposition für verschiedene Fenstergrößen und Modell-Architekturen

Fenster	Acc	F_1	Architektur
1 s	99.8%	99.8%	InceptionTime
2 s	100.0%	100.0%	DeepConvLSTM
3 s	99.8%	99.7%	ResNet

Tabelle 5.3: Treffergenauigkeit und F_1 -Maß der besten Gerätepositions-Klassifikatoren für verschiedene Fenstergrößen

Dieses hervorragende Ergebnis deckt sich mit den Erfahrungen aus der Literatur in anderen Datensätzen. Eine mögliche Erklärung dafür sind die deutlichen

Unterschiede zwischen den einzelnen Klassen, u. a. die Geräteausrichtung und die auf das Gerät wirkenden Beschleunigungskräfte betreffend: In der Tasche oder am Arm schwingend zeigt das Gerät in der Regel nach unten, während es beim Telefonieren eher nach oben ausgerichtet ist. Vor dem Körper oder schwingend getragen erfährt es andere Beschleunigungskräfte als bei engerer Koppelung an den Körper bzw. Kopf der Person.

5.2.4.2 Flache Klassifikation

Nach der Geräteposition wird nun untersucht, wie zuverlässig die neun Aktivitäten mithilfe eines flachen Klassifikators (K_1 in Abbildung 5.2) klassifiziert werden können. Zunächst wird der allgemeine Fall betrachtet, bei dem der Klassifikator ohne Wissen um die Geräteposition operiert. Die besten Modelle für diese Konfiguration erreichen eine Genauigkeit von 47,4 % (1 s) bis 64,3 % (3 s). In den meisten Fällen führt hier eine längere Fenstergröße zu besseren Ergebnissen (siehe Abbildung 5.5 bzw. Tabelle 5.4).

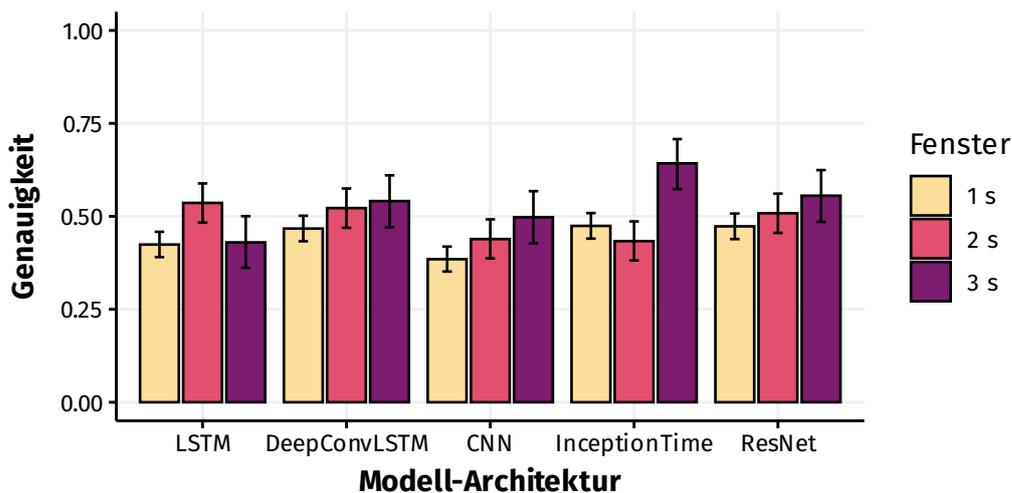


Abbildung 5.5: Treffergenauigkeit für alle neun Klassen für verschiedene Fenstergrößen und Modell-Architekturen bei unbekannter Geräteposition (K_1)

Die Detailergebnisse des besten Modells (InceptionTime 3 s) sind in Abbildung 5.6 als Konfusionsmatrix dargestellt. Probleme treten insbesondere bei der Unterscheidung der beiden Tür-Richtungen sowie beim Geradeausgehen auf. Andere Aktivitäten wie das Stillstehen, Abbiegen oder Treppensteigen werden vergleichsweise zuverlässig erkannt.

Um eine Beeinträchtigung des Resultats durch übermäßiges Reduzieren der Hyperparameter auszuschließen, wurden für diese Konfiguration zusätzlich Modelle

Geräteposition	Fenster	Acc	F_1	Architektur
unbekannt	1 s	47.4%	45.8%	InceptionTime
	2 s	53.6%	51.6%	LSTM
	3 s	64.3%	60.0%	InceptionTime
vorne	1 s	60.8%	58.7%	DeepConvLSTM
	2 s	66.7%	N/A	DeepConvLSTM
	3 s	75.6%	76.2%	DeepConvLSTM
schwingend	1 s	50.4%	46.9%	DeepConvLSTM
	2 s	47.2%	N/A	InceptionTime
	3 s	61.1%	N/A	ResNet
Telefonat	1 s	52.0%	N/A	LSTM
	2 s	60.0%	57.1%	LSTM
	3 s	60.0%	N/A	InceptionTime
Tasche	1 s	47.4%	44.4%	DeepConvLSTM
	2 s	65.7%	N/A	InceptionTime
	3 s	64.8%	N/A	ResNet

Tabelle 5.4: Treffergenauigkeit und F_1 -Maß der besten 9-Klassen-Klassifikatoren (K_1) für verschiedene Gerätepositionen und Fenstergrößen

ohne derartige Einschränkungen trainiert. Die Ergebnisse streuen etwas weniger, zeigen aber insgesamt ein ähnliches Bild. Die maximale Genauigkeit beträgt hier 62,8 %, erreicht von der LSTM-Architektur bei einer Fenstergröße von 3 s. Bei kürzeren, einsekündigen Fenstern verbessert sich das beste Ergebnis auf 51,5 % Genauigkeit (ebenfalls LSTM). Das nicht vollends überzeugende Resultat scheint also nicht den Modellen geschuldet zu sein, sondern vielmehr dem Datensatz und den nicht immer eindeutig voneinander unterscheidbaren Aktivitätsmustern.

Auf bestimmte Gerätepositionen konditionierte Klassifikatoren verbessern die Situation nur in manchen Fällen (siehe nochmals Tabelle 5.4). Die besten Ergebnisse werden erzielt, wenn das Gerät vor dem Körper gehalten wird. Hier erreicht der DeepConvLSTM-Klassifikator bei einer Fenstergröße von 3 s eine Genauigkeit von 75,6 %. Befindet sich das Gerät schwingend in der Hand, so beträgt die Genauigkeit im besten Fall 61,1 %. Beim Telefonieren verringert sich die maximal erreichte Genauigkeit gegenüber dem allgemeinen Fall um 4,3 Prozentpunkte, in der Tasche erhöht sie sich leicht. Nach wie vor ist in den meisten Fällen eine längere Fenstergröße von Vorteil. Wie aus Tabelle 5.4 ersichtlich ist, kann das F_1 -Maß aufgrund von komplett fehlklassifizierten Aktivitäten teilweise nicht berechnet werden.

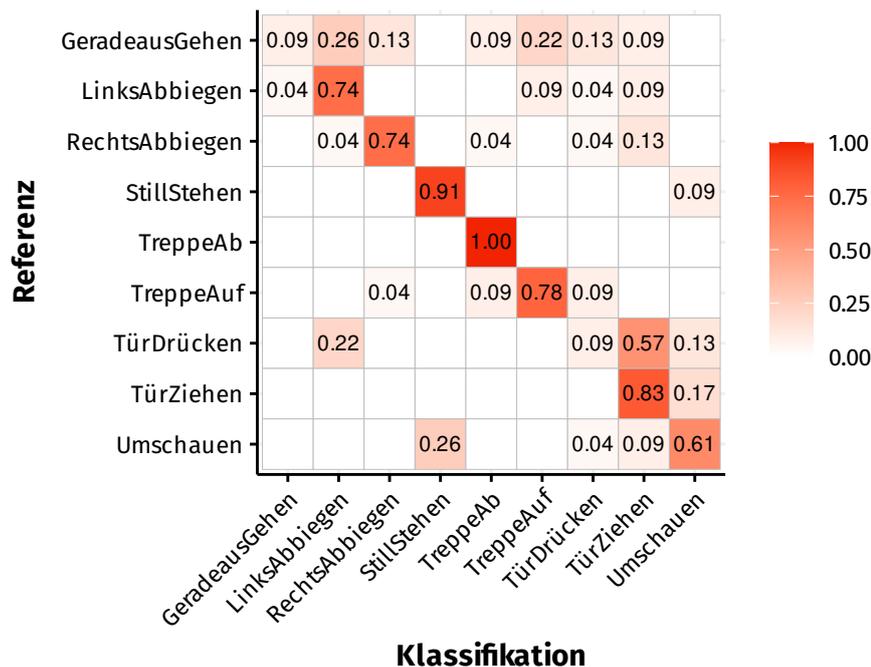


Abbildung 5.6: Konfusionsmatrix des besten 9-Klassen-Klassifikators bei unbekannter Geräteposition (InceptionTime 3 s)

5.2.4.3 Zweistufige Klassifikation

Wie der vorige Abschnitt zeigt, stellen die neun teils sehr ähnlichen Aktivitäten eine gewisse Herausforderung an die Klassifikatoren dar. Daher werden nun zunächst Klassifikatoren für vier grobe Klassen (in Abbildung 5.3 als K_2 bezeichnet) trainiert und evaluiert.

Mit Ausnahme der reinen LSTM-Architektur erreichen alle Klassifikatoren bei einer Fenstergröße von 3 s mindestens 75 % Genauigkeit, angeführt wiederum von InceptionTime (siehe Abbildung 5.7 / Tabelle 5.5). Die Konfusionsmatrix des besten Modells (Abbildung 5.8) zeigt, dass die meisten Fehlerfälle auf die Unterscheidung von Gehen und Türen zurückzuführen sind. Dies ist dadurch erklärbar, dass jeder Türdurchgang unweigerlich Gehbewegungen enthält. Deutlich akkurater ist hingegen die Erkennung von Treppen und stationären Phasen, welche sich, wie bereits aus Abbildung 5.1 hervorgeht, deutlicher hinsichtlich ihrer Sensor-Rohdaten unterscheiden.

Bei den auf bestimmte Gerätepositionen spezialisierten Klassifikatoren zeigt sich ein ähnliches Bild wie zuvor: Wiederum wird eine höhere Genauigkeit erreicht, wenn das Gerät vor dem Körper oder in der Tasche getragen wird. In den anderen Fällen kann keine Verbesserung im Vergleich zum allgemeinen Fall erzielt werden (siehe nochmals Tabelle 5.5).

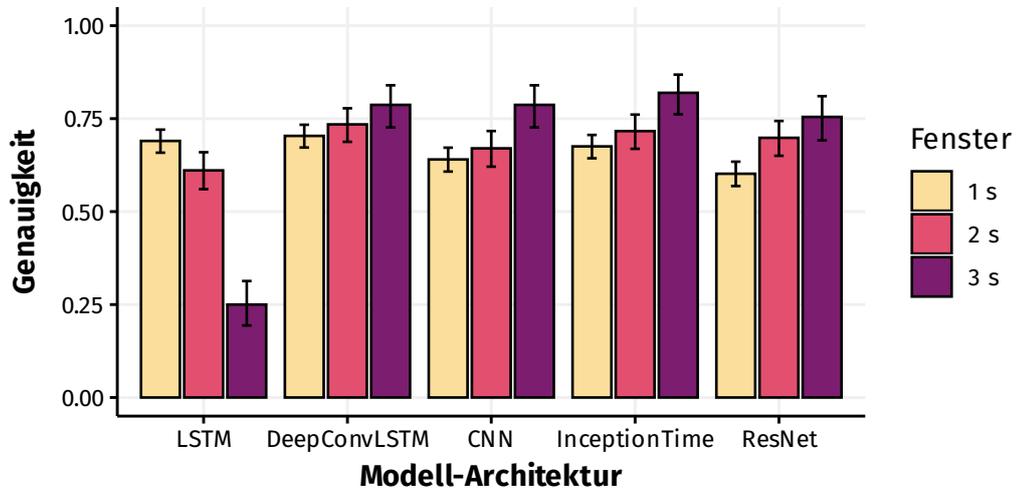


Abbildung 5.7: Klassifikations-Genauigkeit bei vier groben Klassen für verschiedene Fenstergrößen und Modell-Architekturen (K_2)

Geräteposition	Fenster	Acc	F_1	Architektur
unbekannt	1 s	70.4%	68.9%	DeepConvLSTM
	2 s	73.5%	72.3%	DeepConvLSTM
	3 s	81.9%	81.4%	InceptionTime
vorne	1 s	76.0%	74.4%	LSTM
	2 s	88.8%	88.3%	DeepConvLSTM
	3 s	88.6%	88.1%	InceptionTime
schwingend	1 s	64.0%	61.0%	DeepConvLSTM
	2 s	70.2%	68.7%	DeepConvLSTM
	3 s	80.4%	80.3%	DeepConvLSTM
Telefonat	1 s	69.4%	65.5%	LSTM
	2 s	76.2%	73.7%	DeepConvLSTM
	3 s	77.3%	70.4%	ResNet
Tasche	1 s	75.4%	74.4%	DeepConvLSTM
	2 s	77.8%	77.3%	LSTM
	3 s	88.3%	87.8%	InceptionTime

Tabelle 5.5: Treffergenauigkeit und F_1 -Maß der besten 4-Klassen-Klassifikatoren (K_2) für verschiedene Gerätepositionen und Fenstergrößen

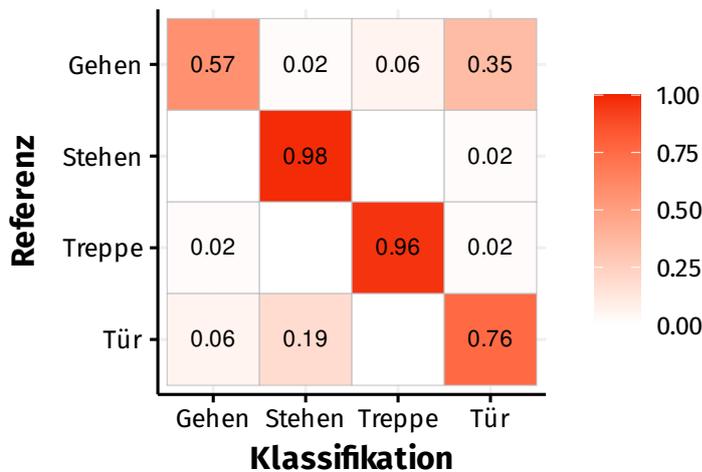


Abbildung 5.8: Konfusionsmatrix des besten 4-Klassen-Klassifikators (K_2)

Gemäß dem oben beschriebenen hierarchischen Ansatz folgen nun die nachgeschalteten Klassifikatoren auf zweiter Ebene:

Stationäres Verhalten (K_3)

Die Aktivitäten `StillStehen` und `Umschauen` als Vertreter des stationären Benutzerverhaltens werden im besten Fall zu 86,5 % (DeepConvLSTM bei einer Fenstergröße von 2 s) korrekt klassifiziert, fast immer liegt die Genauigkeit über 75 %. Dies ist in Anbetracht der Tatsache, dass es sich um zwei recht ähnliche und nicht immer eindeutig voneinander abgrenzbare Aktivitäten handelt, ein erfreuliches Ergebnis.

Tabelle 5.6 gibt eine Übersicht über die maximal erreichte Treffergenauigkeit aller nachgeschalteten Klassifikatoren, aufgeschlüsselt nach Geräteposition, Fenstergröße und Aktivität. Die spezialisierten Varianten des stationären Klassifikators erreichen bei einem vor dem Körper gehaltenen Gerät sowie beim Telefonieren eine teils deutlich bessere Genauigkeit von bis zu 91,7 % bzw. 96,0 %. In der Tasche und insbesondere bei schwingendem Gerät verringert sich die Genauigkeit. Für die Unterscheidung der beiden stationären Klassen erweisen sich mehr Daten nicht immer als Vorteil. In der Mehrzahl der Fälle wird das beste Ergebnis bei einer Fenstergröße von 2 s erreicht.

Geräteposition	Fenster	Stehen K_3	Gehen K_4	Türen K_5	Treppen K_6
unbekannt	1 s	83.9% (81.9%)	54.5% (52.5%)	56.8% (64.3%)	91.2% (91.9%)
	2 s	86.5% (84.7%)	65.8% (65.7%)	54.2% (60.0%)	100.0% (100.0%)
	3 s	81.9% (78.8%)	68.1% (58.8%)	59.6% (68.7%)	100.0% (100.0%)
vorne	1 s	85.1% (82.5%)	71.9% (71.8%)	56.2% (61.8%)	100.0% (100.0%)
	2 s	91.7% (90.9%)	71.4% (69.8%)	55.6% (66.7%)	100.0% (100.0%)
	3 s	84.8% (82.9%)	88.9% (88.6%)	80.0% (80.0%)	100.0% (100.0%)
schwingend	1 s	74.4% (67.7%)	47.4% (44.2%)	64.8% (66.7%)	92.4% (93.0%)
	2 s	78.8% (74.6%)	61.1% (59.3%)	70.8% (63.2%)	96.4% (96.6%)
	3 s	75.0% (76.4%)	61.1% (N/A)	41.7% (58.8%)	100.0% (100.0%)
Telefonat	1 s	89.6% (88.6%)	57.6% (56.5%)	56.5% (67.7%)	100.0% (100.0%)
	2 s	91.0% (90.4%)	66.7% (64.1%)	55.0% (69.0%)	100.0% (100.0%)
	3 s	96.0% (96.0%)	66.7% (61.6%)	50.0% (66.7%)	100.0% (100.0%)
Tasche	1 s	78.7% (77.9%)	47.4% (47.5%)	56.9% (54.5%)	89.3% (90.0%)
	2 s	84.8% (83.3%)	54.5% (54.0%)	50.0% (48.0%)	100.0% (100.0%)
	3 s	77.5% (72.7%)	61.1% (N/A)	57.1% (62.5%)	100.0% (100.0%)

Tabelle 5.6: Treffergenauigkeit und F_1 -Maß (in Klammern) der besten nachgeschalteten Klassifikatoren (K_{3-6}) für verschiedene Gerätepositionen und Fenstergrößen

Gehen (K_4)

Der 9-Klassen-Klassifikator hatte unerwartete Probleme, das Geradeausgehen korrekt zu identifizieren. Auch beim auf Gehmuster spezialisierten Klassifikator ist dies nicht immer zuverlässig möglich, mit einer maximalen Treffergenauigkeit von

68,1 % im Fall unbekannter Geräteposition (siehe wiederum Tabelle 5.6). Da es sich in diesem Fall um 3 Klassen handelt, ist der Vergleich mit den übrigen nachgeschalteten Klassifikatoren nicht unmittelbar möglich (diese sind jeweils binär).

Die Klassifikation ist am zuverlässigsten (bis zu 88,9 % Treffergenauigkeit bei einer Fenstergröße von 3 s), wenn das Gerät vor dem Körper getragen wird. Bei allen anderen Positionen werden maximal zwischen 61,1 % und 66,7 % erreicht, wobei in manchen Fällen wiederum aufgrund kompletter Fehlklassifizierung einer Klasse kein F_1 -Maß errechnet werden kann.

Eine mögliche Erklärung für die Schwierigkeiten ist die Tatsache, dass sich die Klassen hauptsächlich durch eine graduelle Rotation des Geräts um eine der Achsen (bzw. das Ausbleiben davon beim Geradeausgehen) unterscheiden. Diese kann je nach Geräteposition sehr leicht im Rauschen der sonstigen Bewegungen untergehen, während sich das Abbiegen bei einem nach vorne zeigenden Gerät direkt auf die z-Achse des Gyroskops auswirkt.

Türen (K_5)

Die Unterscheidung der Tür-Richtungen ist im allgemeinen Fall bei unbekannter Geräteposition nicht zuverlässig möglich, der beste Klassifikator erreicht lediglich eine Genauigkeit von 59,6 %.

Bei bekannter Geräteposition verbessert sich das Ergebnis in den Fällen eines vor dem Körper getragenen Geräts auf maximal 80,0 %, schwingend in der Hand werden bis zu 70,8 % erreicht. Die Schwankungsbreite je nach Fenstergröße ist in beiden Fällen jedoch sehr groß. Bei den Gerätepositionen Telefonat und Tasche ist keine Verbesserung zu beobachten (siehe Tabelle 5.6).

Mögliche Gründe für dieses Ergebnis sind zum einen unterschiedliche Arten von Türen: Bei der Erhebung des Datensatzes wurde diesbezüglich keine konkrete Vorgabe gemacht. Es ist daher möglich, dass die zum Training verwendeten Bewegungsabläufe in dieser Hinsicht nicht sehr gut generalisierbar sind. Zum anderen finden sich in den aufgezeichneten Daten unmittelbar vor und nach jedem Türdurchgang reguläre Gehbewegungen.

Treppen (K_6)

Der letzte Klassifikator widmet sich der Unterscheidung der beiden Treppen-Richtungen aufwärts und abwärts. Dies funktioniert in allen Fällen äußerst zuverlässig mit einer Genauigkeit von bis zu 100 %. Hier wird deutlich, wie robust die Klassifikation dank des Barometers als dediziertem Sensor für Höhenveränderungen ist, wobei einschränkend erwähnt werden muss, dass das Treppensteigen in diesem Datensatz die einzige betrachtete Art des Stockwerkswechsels darstellt.

Aus Abschnitt 4.4.4 ist bekannt, dass das Barometer in den von den BenutzerInnen der URwalking-App verwendeten Geräten unter allen betrachteten Sensoren am seltensten verbaut ist. Vergleichsweise ohne die Luftdruckmesswerte trainierte Modelle erreichen hier eine Treffergenauigkeit zwischen 58,6 % und 84,7 %.

Gesamte Klassifikator-Hierarchie

Zuletzt wird nun – wie oben beschrieben – die gesamte Klassifikator-Hierarchie mit dem angepassten F_1 -Maß evaluiert. Da dabei auch teilweise korrekte Klassifikationen positiv angerechnet werden, sind die Ergebnisse zwischen Hierarchien unterschiedlicher Tiefe nicht unmittelbar vergleichbar. Im Realbetrieb sollte eine Klassifikator-Hierarchie idealerweise für jedes Teilproblem den am besten geeigneten Klassifikator verwenden. Um keine Erkenntnisse aus den bis dato präsentierten Ergebnissen einfließen zu lassen und somit die Trennung zwischen Trainings- und Testdatensatz aufrechtzuerhalten, wird in dieser Evaluation darauf verzichtet.

Bei unbekannter Geräteposition, d. h. einer Hierarchie aus grobem (K_2) und mehreren nachgeschalteten (K_{3-6}) Klassifikatoren, beträgt das maximale hierarchische F_1 -Maß 0,71 (InceptionTime 3 s). Mit vorgeschalteter Geräteposition und somit einer dreistufigen Hierarchie wird ein F_1 -Maß von 0,78 erreicht (siehe Abbildung 5.9).

5.2.4.4 Fazit

Die Evaluation der HAR-Modelle zeigt, dass sehr granulare Aktivitäten wie beispielsweise unterschiedliche Tür-Richtungen nur schwer voneinander unterscheidbar sind. Dies ist jedoch in erster Linie als eine Machbarkeitsstudie zu sehen und hat nur geringe Auswirkungen für die Einsetzbarkeit im Positionierungssystem. Wichtiger – und in Anbetracht des Ergebnisses erfreulicher – ist die grundsätzliche Erkennung von Türen. Ebenso funktioniert die Bestimmung der Geräteposition sowie die Unterscheidung von Gehen, Stehen und Treppensteigen sehr zuverlässig.

Inwiefern die Verbesserungen, die der hierarchische Ansatz mit sich bringt, in der Praxis den Einsatz mehrerer Klassifikatoren rechtfertigen, muss je nach Anwendungsfall entschieden werden. Im Vergleich zu den hier präsentierten Zahlen besteht ein gewisses Optimierungspotential durch gezieltes *Cherry-Picking* der jeweils besten Einzel-Klassifikatoren, d. h. durch Verwendung von unterschiedlichen Fenstergrößen und Architekturen für die spezifischen Teilprobleme.

Vor der Integration der besten Einzelklassifikatoren in das Positionierungssystem werden sie im folgenden Abschnitt auf den naturalistischen Log-Datensatz der URwalking-App angewendet.

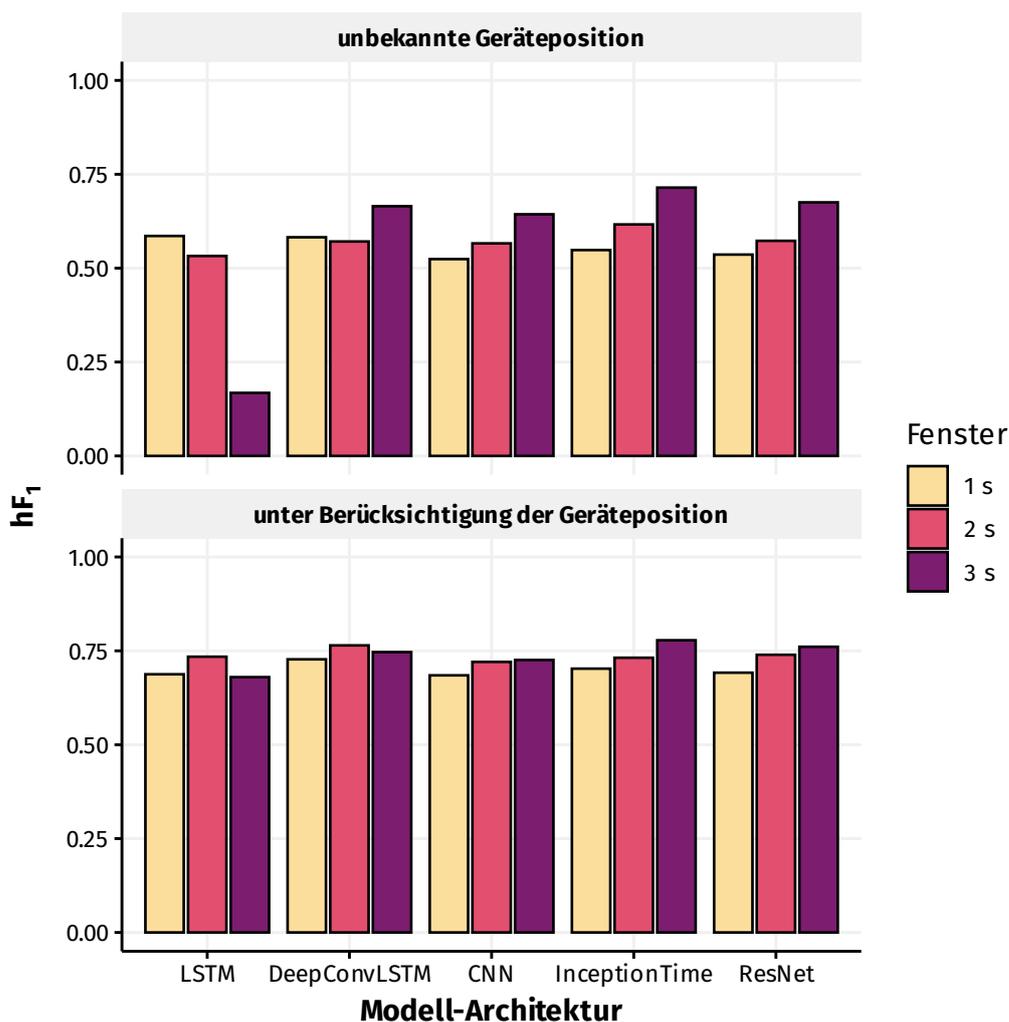


Abbildung 5.9: Evaluationsergebnisse (angepasstes F_1 -Maß) der hierarchischen Klassifikation

5.3 Anwendung von HAR in naturalistischen Studien

In Kapitel 4 blieb die Frage nach der vorherrschenden Geräteposition unter realistischen Navigationssituationen offen, welche nun mithilfe des soeben beschriebenen HAR-Verfahrens beantwortet werden soll. Weiterhin werden im Folgenden die Benutzeraktivitäten während der Campus-Navigation mit der URwalking-App untersucht, wobei sich die Analysen auf diejenigen Klassifikatoren beschränken, die in der Evaluation eine hohe Genauigkeit aufwiesen und deren Prädiktionen als verlässlich gelten können. Dennoch ist bei der Interpretation der nachfolgend präsentierten Ergebnisse zu beachten, dass die Klassifikatoren nicht

perfekt arbeiten und keine Ground Truth zur Verfügung steht, um diese zu validieren. Der Fokus liegt daher nicht auf einzelnen NutzerInnen oder Navigationssitzungen, sondern auf der Untersuchung allgemeiner Trends zum Navigationsverhalten, wodurch etwaige Fehler oder mit Rauschen behaftete Prädiktionen geringeren Einfluss haben. Im Einzelnen werden im Folgenden die Geräteposition sowie die Verteilung von stationären und Bewegungsphasen – jeweils im Verlauf der Navigationssitzungen – untersucht.

Für die Analysen wird wiederum der heuristisch gefilterte Log-Datensatz verwendet, in dem sich nur die tatsächlichen Navigationssitzungen befinden. Zusätzlich wurden diejenigen Sitzungen entfernt, in denen nicht alle für das HAR-Verfahren benötigten Sensoren vorhanden sind, wodurch ein Datensatz von 258 Navigationssitzungen verbleibt. Für jede der Sitzungen wurden ähnliche Vorverarbeitungsschritte wie im HAR-Datensatz durchgeführt: Nach Ausrichtung und Zusammenführung der verschiedenen Sensor-Messwerte anhand ihrer Zeitstempel wurden die Daten in überlappende Fenster mit einem Abstand von 0,5 s aufgeteilt. Dies bildet die im Realbetrieb angestrebte Aktualisierungsrate von 2 Hz nach und sorgt zugleich dafür, dass mehr Datenpunkte als bei nicht überlappenden Fenstern generiert werden.

Für jedes Teilproblem wird das jeweils performanteste Modell aus der vorherigen Evaluation herangezogen, wodurch auch die Fenstergröße der verwendeten Eingabedaten variiert. Nach Ausführung der Modelle entsteht für jede Sitzung eine Liste variabler Länge von Klassifikationsergebnissen, die per Zeitstempel mit einem Zeitpunkt während der Navigationssitzung korrespondieren. Um die Änderungen im Zeitverlauf für unterschiedlich lange Sitzungen vergleichen zu können, wurde die Sitzungslänge in 100 gleich große Abschnitte unterteilt und für diese die jeweils vorherrschende Prädiktion bestimmt.

Zur groben Einschätzung der Güte der Klassifikation wird der Softmax-Output interpretiert, welcher zwar kein direktes Maß für die Konfidenz eines Klassifikators darstellt, nach Hendrycks und Gimpel (2018, S. 3) aber als eine gute Baseline dienen kann.

5.3.1 Geräteposition während der Navigation

In Abschnitt 4.5.4 wurde die Geräteposition bereits kurz regelbasiert anhand der z-Richtung des GRV untersucht. Zudem wurden Beispiele für Sensordaten verschiedener Gerätepositionen gezeigt, die zur Auswahl der vier Klassen vorne, schwingend, Tasche und Telefonat beitragen (siehe Abbildungen 4.22 und 4.23). Durch Anwendung des daraus hervorgegangenen Klassifikators auf den naturalistischen Datensatz lässt sich die vorherrschende Geräteposition während tatsächli-

cher Navigation bestimmen.

Die Klassifikation der Geräte-Position findet mithilfe des DeepConvLSTM-Modells bei einer Fenstergröße von 2 s statt – mit dieser Konfiguration wurden während der Evaluation am Testdatensatz sehr gute Ergebnisse erzielt. Um mit niedriger Konfidenz klassifizierte Instanzen zu filtern, gilt ein Schwellenwert von 0,9 für den Softmax-Output.

Wie bereits in den Positionierungsexperimenten beobachtet, wird das Gerät hauptsächlich vor dem Körper getragen (siehe Abbildung 5.10). Der Anteil dieser Geräteposition verringert sich im Laufe der Zeit von anfänglich knapp 70 % auf unter 50 %. Den Großteil der restlichen Zeit verbringt das Gerät in der Tasche, mit leicht zunehmender Tendenz bei fortschreitender Dauer der Navigationssitzung. Die anderen beiden Tragearten spielen eine geringere Rolle und sind vergleichsweise stabil zeitlich verteilt.

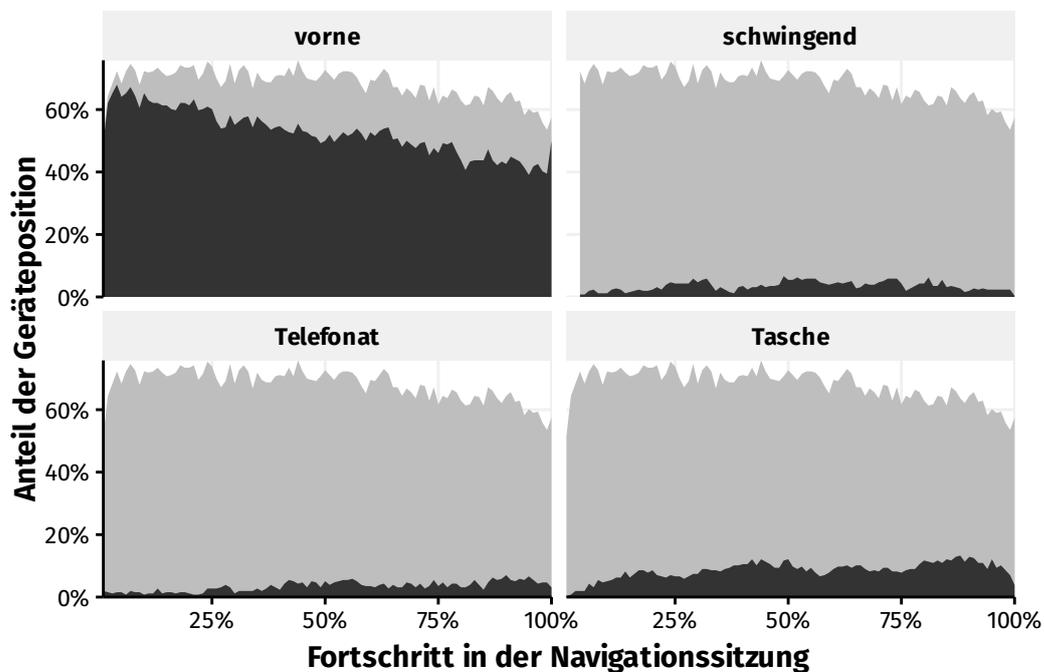


Abbildung 5.10: Verteilung der Geräteposition im Zeitverlauf der Navigationssitzungen (auf die Sitzungslänge normiert).
In Grau: mit hoher Konfidenz klassifizierter Anteil

Auffällig wird die Veränderung der Gerätepositionsverteilung, wenn man die Navigationssitzungen nach ihrer Länge gruppiert betrachtet (Abbildung 5.11): In kurzen und mittleren Sitzungen (hier festgelegt als eine Dauer von maximal 10 Minuten) ist das vor dem Körper getragene Gerät klar vorherrschend, während es sich bei längeren Sitzungen immer häufiger in der Tasche befindet. Parallel dazu

steigt auch der Anteil an Gerätepositionen stark an, die nicht zuverlässig klassifiziert werden können.

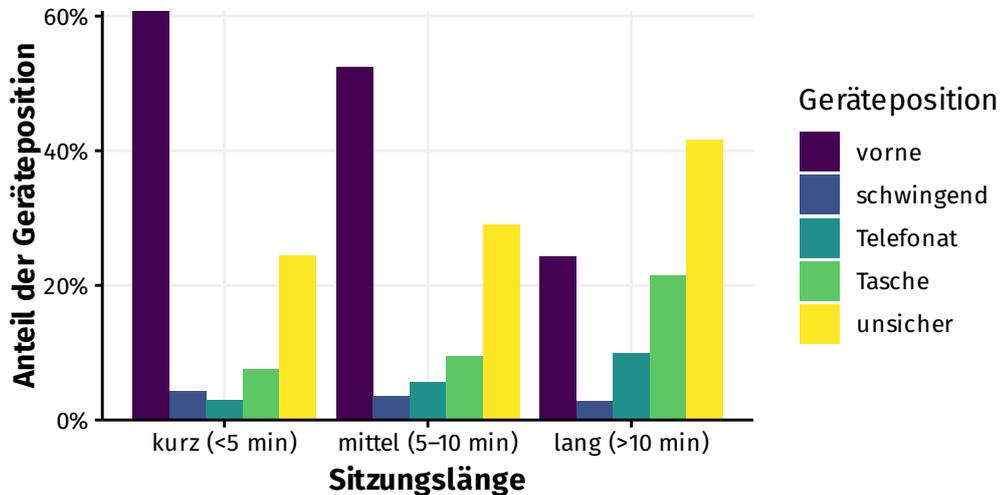


Abbildung 5.11: Verteilung der Geräteposition für unterschiedlich lange Navigationssitzungen

5.3.2 Aktivitäten während der Navigation

Nach der Geräteposition wird nun die Benutzeraktivität näher analysiert. Da sich die Unterscheidung der fein untergliederten Klassen bereits im HAR-Datensatz als schwierig erwiesen hat, kommt hier lediglich der grobe Vier-Klassen-Klassifikator K_2 zum Einsatz. Wie zuvor gilt ein Schwellenwert von 0,9 als Kriterium für die Konfidenz der Klassifikation. In diesem Fall führt dies dazu, dass ca. die Hälfte aller Eingabe-Daten nicht zuverlässig klassifiziert werden können.

Wie Abbildung 5.12 zeigt, überwiegen unter den klassifizierten Daten zu Beginn stationäre Aktivitäten, was durch initiale Orientierungs- bzw. Wayfinding-Phasen erklärt werden kann. In der ersten Hälfte der Navigationssitzungen dominiert dann die Fortbewegung, wobei immer ein deutlicher stationärer Anteil erhalten bleibt. Dieser nimmt in der zweiten Hälfte der Sitzungen wieder zu, bis am Ende grob dieselbe Verteilung wie zu Beginn herrscht. Wie schon zuvor in Abschnitt 4.5.2 zeigt sich auch hier, dass stationäre Phasen einen gewissen Anteil der im realen Betrieb stattfindenden Navigationssitzungen ausmachen und im Positionierungssystem berücksichtigt werden sollten.

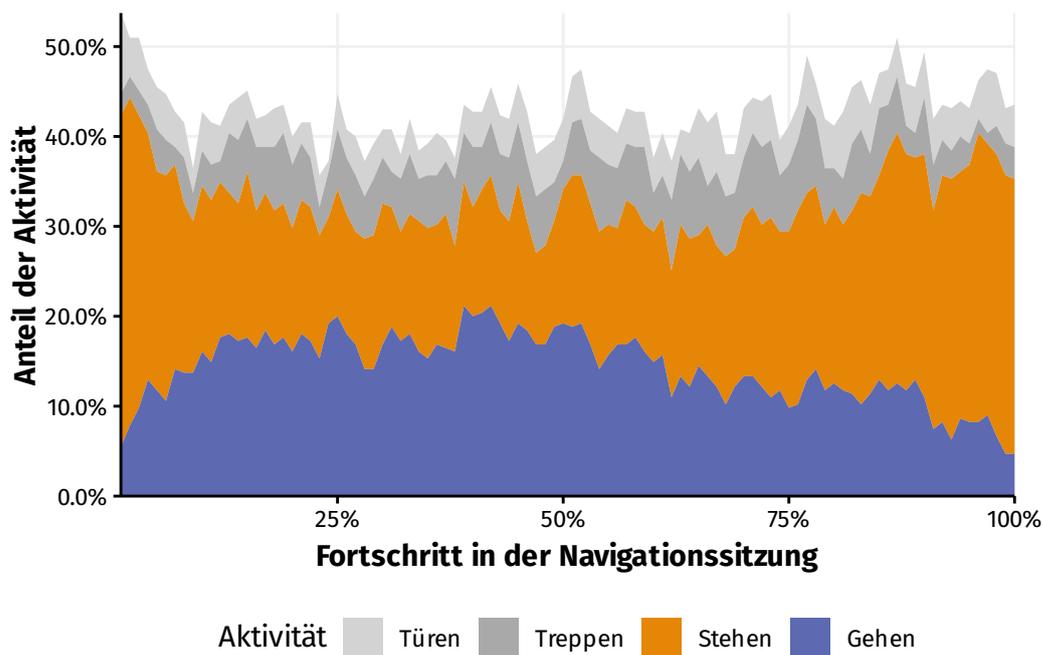


Abbildung 5.12: Verteilung der Benutzeraktivität im Zeitverlauf der Navigationssitzungen (auf die Sitzungslänge normiert)

5.4 HAR zur Verbesserung der Indoor-Positionierung

Im letzten Abschnitt dieses Kapitels werden die Ergebnisse aus dem Bereich HAR mit dem Kernthema der Arbeit, der Indoor-Positionierung, verknüpft. Im Rahmen der in dieser Arbeit berücksichtigten Randbedingungen wird insbesondere die Frage beantwortet, inwiefern durch Einsatz von HAR die Langzeitstabilität der Positionierung durch Erkennung von ortsgebundenen Aktivitäten erhöht werden kann.

Im Folgenden wird zunächst die technische Umsetzung der Einbindung von HAR in das existierende Positionierungssystem beschrieben, bei der wieder vom Partikelfilter-Verfahren Gebrauch gemacht wird. In einem abschließenden Experiment (N = 69) wird das komplette System – inklusive der in Abschnitt 3.3.3 beschriebenen Anpassungen wie der Schrittlängen-Kalibrierung – daraufhin evaluiert.

5.4.1 Integration von HAR in das Positionierungssystem

Um in Verbindung mit dem Umgebungsmodell aus den HAR-Informationen Nutzen für die Indoor-Positionierung zu ziehen, sind Erweiterungen und Anpassungen im Positionierungsverfahren nötig. Wie Beispiele aus der Literatur belegen, ist die partikelfiltergestützte Implementierung gut dafür geeignet, zusätzliche Informationen in die Positionsbestimmung einfließen zu lassen (vgl. z. B. Fetzer et al., 2018; Moreira et al., 2021). Nicht immer jedoch finden sich dort konkrete Implementierungsdetails, weswegen im Folgenden verschiedene Varianten beschrieben und evaluiert werden:

Die erste Variante sieht lediglich eine Gewichtung der existierenden Partikel anhand ihrer Distanz zum nächsten Ort, der die jeweilige Aktivität erlaubt, vor. Dieses Vorgehen wird beispielsweise ebenfalls von Shang et al. (2015) und Moreira et al. (2021) vorgeschlagen.

In der zweiten Variante wird die durch HAR gewonnene Information dazu verwendet, die Partikelmenge direkt zu manipulieren, indem an möglichen Aktivitätsorten neue Partikel erzeugt werden. Bei Thrun et al. (2006, S. 261 f.) wird dieses Verfahren unter der Bezeichnung *Mixture Monte Carlo Localization* beschrieben; als Anteil der neu zu ziehenden Partikel werden 5 % vorgeschlagen.

Zur Bestimmung der Aktivitäten kommt der auf die Geräteposition konditionierte 4-Klassen-Klassifikator (K_2) zum Einsatz, welcher zwischen Gehen, Stehen, Treppensteigen und dem Passieren von Türen unterscheidet. Da sich die Klassifikation von Abbiegungen als unzuverlässig erwiesen hat, werden diese anhand der Winkeländerung aufeinanderfolgender Schritte bestimmt.

Die Integration in das Indoor-Positionierungssystem variiert zwischen ortsbezogenen und ortsunabhängigen Aktivitäten. Nach Erkennung einer ortsbezogenen Aktivität, d. h. beim Treppensteigen, beim Durchqueren einer Tür oder beim Abbiegen, wird für jedes Partikel die Distanz zum nächstgelegenen korrespondierenden Ort bestimmt. Im Falle von Türen werden dabei nur Gebäudeeingänge und Durchgangstüren berücksichtigt, nicht aber beispielsweise Bürotüren, die entlang der Route liegen – unter der Annahme, dass Navigierende diese Räume nicht ohne Aufforderung betreten. Die Orte, an denen Abbiegungen möglich sind, werden anhand der Graphstruktur identifiziert.

Je nach Systemvariante unterscheiden sich die darauf folgenden Schritte. In der ersten Konfiguration wird überprüft, ob die Distanz kleiner als ein vordefinierter maximaler Abstand ist. Falls ja, so erfolgt eine Höhergewichtung des Partikels invers proportional zur Distanz zum relevanten Objekt. Ist der Abstand höher als der Schwellenwert, so ändert sich das Gewicht nicht. Durch dieses Vorgehen beeinflussen falschpositive Klassifikationen fernab von aktivitätsrelevanten Objek-

ten die Positionsverfolgung nicht auf negative Weise (siehe Algorithmus 5.2).

Die alternative Konfiguration sieht vor, dass bei Erkennung einer Aktivität alle oder ein Teil der Partikel instantan zum nächstgelegenen Ort springen, an dem diese plausibel möglich ist – wiederum unter Berücksichtigung einer maximalen Distanz –, und dort neu initialisiert werden. Auf diese Weise erfolgt – unter der Annahme einer zuverlässigen Aktivitätserkennung – die sofortige Korrektur der Position, während sich in der ersten Variante der Positionsfehler im Laufe mehrerer Gewichtung- und Resampling-Vorgänge allmählich reduziert. Nachteil dieses Vorgehens ist, dass durch das direkte Manipulieren der Partikelposition PDR-Informationen verloren gehen, wenn Partikel bei mehrmaliger aufeinanderfolgender Erkennung einer Aktivität immer an denselben Ort gezwungen werden.

Algorithmus 5.2: Partikel-Update bei Erkennung ortsbezogener Aktivitäten

Input: Aktivitätsorte D , Partikelgewicht ω_0 , Partikelposition p

Output: Partikelgewicht ω

- 1: // Distanz zum nächstgelegenen Aktivitätsort bestimmen:
 - 2: $d_{\text{act}} = \min_{d \in D}(\text{distance}(p, d))$
 - 3: **Falls** ($d_{\text{act}} < d_{\text{thresh}}$):
 - 4: // Gewichtsdelta berechnen (modelliert durch \mathcal{N} , mit α als Faktor):
 - 5: $\omega_d = \alpha * \text{density}(\mathcal{N}, d_{\text{act}})$
 - 6: **Ansonsten**
 - 7: $\omega_d = 0$
 - 8: // Partikelgewicht aktualisieren:
 - 9: $\omega = \omega_0 + \omega_d$
 - 10: **Return** ω
-

Ortsunabhängige Aktivitäten, d. h. im vorliegenden Fall die Pausen während der Navigation, sind im Vergleich dazu einfacher zu verarbeiten. Wird eine Gehpause erkannt, so werden für diesen Zeitraum etwaige Updates des Schritterkenners ignoriert, bis sich die Aktivität wieder ändert. Auf diese Weise werden Hand- oder Armbewegungen, welche vom Gerätesensor fälschlicherweise als Schritte interpretiert wurden, aus den für die Positionierung verwendeten Daten gefiltert.

5.4.2 Positionierungsexperiment mit HAR

Das Gesamtsystem mit HAR-Unterstützung wird nun in einem abschließenden, groß angelegten Indoor-Positionierungsexperiment evaluiert. Das Experiment findet wieder in Form von Simulationsstudien, basierend auf Sensordaten aus einer realen Navigationssituation, statt. Dies erlaubt die reproduzierbare Gegenüberstellung unterschiedlicher Systemkonfigurationen mit vertretbarem Aufwand.

5.4.2.1 Experimentaufbau und Datenerhebung

Im Gegensatz zu den Vorstudien liegt nun die Teilnehmerzahl deutlich höher, um statistisch belastbare Aussagen treffen zu können. Anstelle mehrerer kurzer Routen wird eine einzige, längere Route bevorzugt, welche alle zu testenden Aspekte beinhaltet und auf der daher durch den Einsatz von HAR eine Verbesserung der Positionierungsgenauigkeit möglich ist.

Die Datenerhebung fand im Rahmen des Softwareergonomie-Vertiefungsseminars im Wintersemester 2018/2019 statt, geleitet von meiner damaligen Kollegin Christina Bauer. Die Studierenden waren in 12 Gruppen à 2–6 Personen aufgeteilt und für die Akquise der Testpersonen verantwortlich. Während des Experiments agierten die Studierenden als TestleiterInnen. Am Experiment nahmen 102 Personen teil, davon waren 50 weiblich und 52 männlich. Die Testpersonen waren zwischen 18 und 47 Jahre alt ($M = 22,3$; $SD = 3,4$) und überwiegend selbst Studierende (89 TeilnehmerInnen).

Das Experiment fand im westlichen Naturwissenschaftstrakt der Universität Regensburg statt (siehe Abbildung 2.1). Die Testroute beginnt im Durchgang des Mathematik-Gebäudes, welcher eine vielbesuchte Nord-/Süd-Verbindung auf dem Campus darstellt (vgl. auch Abbildung 4.16). Sie führt dann weiter durch den kompletten Physik-Trakt und endet im ersten Obergeschoss des Biologie-Neubaus bei Hörsaal 53. Auf der Route wechseln sich Abbiegungen und gerade Abschnitte ab, es müssen mehrere Durchgangstüren sowie eine Treppe passiert werden. Aufgrund von Bauarbeiten während der mehrere Wochen andauernden Datenerhebung musste der Startpunkt der Route für einen Teil der Experimente zum zweiten Routenabschnitt beim Übergang von Mathematik- und Physikgebäude verlegt werden. Um die Vergleichbarkeit der Ergebnisse zu gewährleisten, beginnen die Simulationsstudien einheitlich am vorverlegten Startpunkt. Aus dem gleichen Grund entfällt auch der letzte Abschnitt: Nach dem Ankommen im ersten Obergeschoss wurde Hörsaal 53 aufgrund einer missverständlichen Navigationsanweisung nur von einem Teil der Testpersonen erreicht. In vielen Fällen musste das Experiment zuvor abgebrochen werden, nachdem sich die Testpersonen verlaufen hatten. Die gekürzte Route ist ca. 270 m lang und besteht aus 13 Abschnitten, jeweils mit dazugehörigen Navigationsanweisungen, welche vom URwalking-Routenserver generiert wurden (siehe Tabelle 5.7 und Abbildung 5.13). Hinsichtlich ihrer Länge und Komplexität ist sie vergleichbar mit den im realen URwalking-Betrieb angefragten Routen. Zudem enthält sie – trotz der vorgenommenen Kürzung – alle für die Evaluation der HAR-Erweiterungen nötigen Merkmale.

Wie im vorherigen Positionierungsexperiment wurden zwei Google-Pixel-Geräte verwendet. Auf dem Gerät der Testperson lief eine leicht angepasste Versi-



Abbildung 5.13: Die Testroute im abschließenden Positionierungsexperiment

Abschnitt	Anweisung	Länge
-	Betreten Sie den Bereich UR Physik, Erdgeschoss.	38,0 m
0	Gehen Sie durch die Tür.	46,1 m
1	Gehen Sie durch die Tür.	52,4 m
2	Gehen Sie durch die Tür.	13,2 m
3	Biegen Sie rechts ab.	5,3 m
4	Gehen Sie durch die Tür.	21,8 m
5	Gehen Sie an den Treppen vorbei.	17,5 m
6	Betreten Sie den Bereich »UR Biologie Ausweichgebäude, Erdgeschoss«.	26,2 m
7	Gehen Sie durch die Tür.	12,1 m
8	Gehen Sie durch die Tür links.	6,5 m
9	Gehen Sie am Wegpunkt » Spinde « vorbei.	27,4 m
10	Gehen Sie durch die Tür.	16,0 m
11	Biegen Sie rechts ab.	17,0 m
12	Nehmen Sie die Treppe und gehen Sie ein Stockwerk nach oben in das 1. Stockwerk.	8,7 m
-	Gehen Sie durch die Tür links.	29,2 m
-	Ihr Ziel liegt vor Ihnen.	-

Tabelle 5.7: Navigationsanweisungen für die Testroute im abschließenden Experiment. Ausgegraute Abschnitte werden für die Simulation nicht verwendet.

on der URwalking-Navigations-App, welche alle später benötigten Sensordaten sowie ein Video der Frontkamera aufzeichnete und Statusinformationen an das Testleitergerät übertrug. Die Zustimmung zur Videoaufzeichnung wurde im Voraus eingeholt und war Voraussetzung für die Teilnahme am Experiment.

Für eine möglichst realistische Navigationssituation wurden abermals keine Vorgaben hinsichtlich der Haltung oder Ausrichtung des Geräts gemacht. Um sicherzustellen, dass keine Anweisungen übersehen werden – insbesondere, da einige aufeinanderfolgende Anweisungen gleich lauteten –, war die Testperson in diesem Experiment dazu aufgefordert, nach Ausführung einer Anweisung selbst zum nächsten Abschnitt zu wechseln. Die TestleiterInnen waren angewiesen, der Testperson in geringem Abstand zu folgen, nur auf explizite Nachfrage Informationen zum Routenverlauf zu geben und den aktuellen Zustand des Navigationsvorgangs festzuhalten – z. B. ob sich die Testperson verläuft, stehen bleibt oder um Hilfe fragt.

Nach dem Absolvieren der Route wurden die Testpersonen wie im vorigen Experiment gebeten, die einzelnen Routenabschnitte auf einer fünfstufigen end-

punktbenannten Likert-Skala hinsichtlich ihrer Vertrautheit zu bewerten (Frage: »Wie vertraut waren Sie vor dem Experiment mit diesem Abschnitt?«, Antworten von »Gar nicht« bis »Sehr«). Die Bewertungen fallen – nahezu kontinuierlich – von 3,81 ($SD = 1,43$) im ersten Abschnitt auf 2,50 ($SD = 1,52$) im letzten Abschnitt. Die Teststrecke führt also von einem im Allgemeinen recht vertrauten Ort zu einem weniger häufig besuchten Areal, wodurch eine realistische Navigationssituation widerspiegelt wird.

5.4.2.2 Datenaufbereitung

Die durch das selbstständige Weiterschalten der Navigationsanweisungen implizit erhobene Ground Truth stellte sich als nicht zuverlässig genug für die Positionierungsexperimente heraus. Viele Testpersonen bestätigten das Absolvieren von Abschnitten zu spät, teils verwirrten auch repetitive Navigationsanweisungen. Aus diesem Grund wurde nachträglich anhand der Videoaufnahmen eine akkurate Ground Truth mit einer Genauigkeit von ca. 1 m bestimmt. Hierfür wurden in der Aufzeichnung markante Objekte an den Abschnittswechseln identifiziert und die entsprechenden im Video enthaltenen Zeitstempel extrahiert. Dies wurde durch die Tatsache, dass das Gerät meist in stabiler Position vor dem Körper gehalten wurde, erleichtert. In manchen Fällen waren die Abschnittswechsel aufgrund von schlechten Lichtverhältnissen nicht erkennbar und wurden dort unter Annahme konstanter Gehgeschwindigkeit durch zeitliche Interpolation bestimmt.

Ein Teil der Daten eignete sich nicht als Grundlage für das Positionierungsexperiment. In 18 Fällen verließen die Testpersonen deutlich die Route, beispielsweise durch fehlerhaftes Abbiegen oder Treppensteigen. Um die Vergleichbarkeit der Testläufe zu gewährleisten, wurden diese daher entfernt. Kleinere Abweichungen wie etwa das temporäre Verpassen einer Abzweigung mit unmittelbar folgender selbstständiger Korrektur blieben jedoch im Datensatz enthalten. Acht Testläufe wurden aufgrund von zu großer Entfernung zum festgelegten Startpunkt ausgeschlossen. In sechs Fällen konnten aufgrund eines technischen Fehlers die Zeitstempel im Video nicht mit denen der Sensordaten abgeglichen werden; in einem weiteren Durchlauf lag kein Video zur Bestimmung der Ground Truth vor. Die verbleibenden 69 Testläufe sind im Mittel 252,4 s lang ($SD = 38,5$ s) und umfassen einen Zeitraum von knapp 5 h.

Der mittlere Abstand zwischen zwei Schritten beträgt 633 ms ($Mdn = 561$ ms). Es handelt sich demnach um eine weniger schiefe Verteilung als im naturalistischen Datensatz mit weniger langen Pausen; der längste Zeitraum ohne erkannten Schritt dauert ca. 46 s.

5.4.2.3 Hypothesen

Im weiteren werden einige Hypothesen untersucht. Die allgemeine Frage ist, ob sich im vorliegenden Fall ein grundsätzlicher Vorteil durch den Einsatz von HAR einstellt:

- H_1 : Die Positionierungsgenauigkeit verbessert sich durch den Einsatz von HAR.
- H_0 : Die Positionierungsgenauigkeit verbessert sich durch den Einsatz von HAR nicht.

Dies wiederum wird in Form einzelner Teilaspekte erforscht. Zunächst stehen die stationären Phasen im Fokus:

- H_1 : Die Positionierungsgenauigkeit verbessert sich durch die Erkennung stationärer Phasen.
- H_0 : Die Positionierungsgenauigkeit verbessert sich durch die Erkennung stationärer Phasen nicht.

Anschließend wird die Positionskorrektur anhand von standortbezogenen Aktivitäten untersucht:

- H_1 : Die Positionierungsgenauigkeit verbessert sich durch die Erkennung standortbezogener Aktivitäten.
- H_0 : Die Positionierungsgenauigkeit verbessert sich durch die Erkennung standortbezogener Aktivitäten nicht.

5.4.2.4 Ergebnisse

Wie zuvor wurde in der Simulationsstudie jeder pro Testperson aufgezeichnete Durchlauf zehnmals mit unterschiedlichen Startwerten für den Zufallsgenerator wiederholt. Das Partikelfilter war wiederum mit 1000 Partikeln konfiguriert.

Baseline-System

Erste mit dem Baseline-System durchgeführte Simulationen zeigen, dass die Route in diesem Experiment im Vergleich zu den Vorstudien höhere Anforderungen an das Positionierungssystem stellt.

Ohne Schrittlängenadaption und Routenbias liegt die mittlere Trefferquote der Routenabschnitte bei 60,3 % ($SD = 20,5$ Prozentpunkte). Mit Routenbias steigt dieser Wert auf 60,8 % ($SD = 20,3$ Prozentpunkte), während das Aktivieren der Schrittlängenadaption zu einer Trefferquote von 65,8 % ($SD = 18,5$ Prozentpunkte)

führt. Beide Optimierungen zusammen ergeben einen Wert von 66,2%; in diesem Fall ist auch die Streuung unter den Testpersonen mit einer Standardabweichung von 17,6 Prozentpunkten am geringsten. Ein Kruskal-Wallis-Rangsummentest weist bei geringer Effektstärke auf statistisch signifikante Unterschiede zwischen den Konfigurationen hin ($\chi^2(3) = 21; p < 0,01$). Ein post-hoc durchgeführter Dunnett-Tukey-Kramer-Test ist auf einem α -Niveau von 0,05 paarweise immer dann signifikant, wenn in einer der Konfigurationen die Schrittlängenadaption aktiviert ist. Wie aus der Verteilung der Ergebnisse über die Testpersonen hinweg erkennbar ist, sinkt durch die Schrittlängenadaption die Anzahl der Ausreißer nach unten deutlich (siehe Abbildung 5.14). Alle weiteren Experimentdurchläufe werden daher mit den beiden Optimierungen durchgeführt.

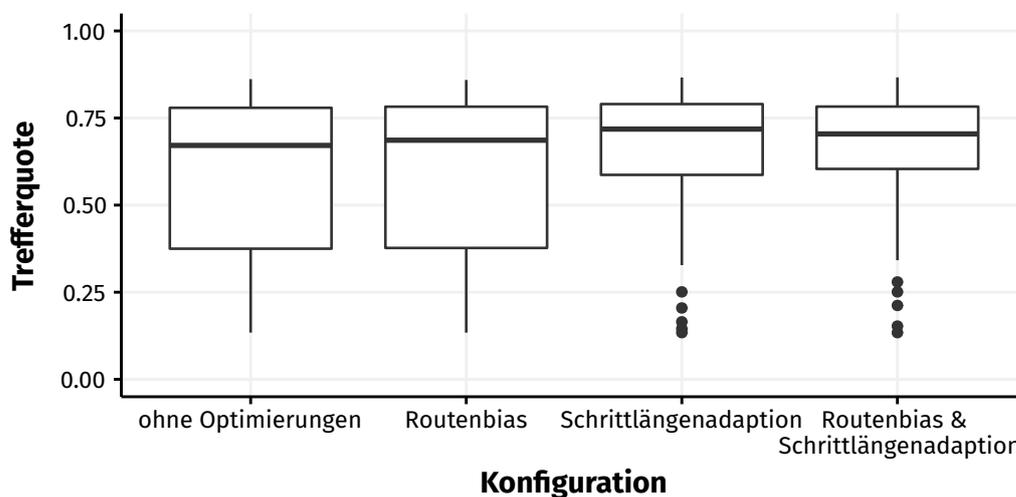


Abbildung 5.14: Trefferquote des Baseline-Systems in vier verschiedenen Konfigurationen

Abbildung 5.15 gibt einen Überblick über die Zuverlässigkeit der Positionsbestimmung im Routenverlauf für die vier Konfigurationen. Hier ist wie schon in den vorherigen Experimenten erkennbar, dass insbesondere nach Abbiegungen, die auf lange Geradeaus-Passagen folgen, die Genauigkeit sinkt. Die Betrachtung einzelner Simulationsdurchläufe zeigt, dass v. a. ohne Schrittlängenadaption an diesen Stellen oft die Positionsverfolgung nicht aufrecht erhalten werden kann. Dennoch geht aus der abschnittswisen Analyse auch hervor, dass das System grundsätzlich in der Lage ist, fehlerhafte Positionsschätzungen selbst zu korrigieren. So ist die Trefferquote in den letzten drei Abschnitten der Route beispielsweise konsistent höher als in den unmittelbar davorliegenden.

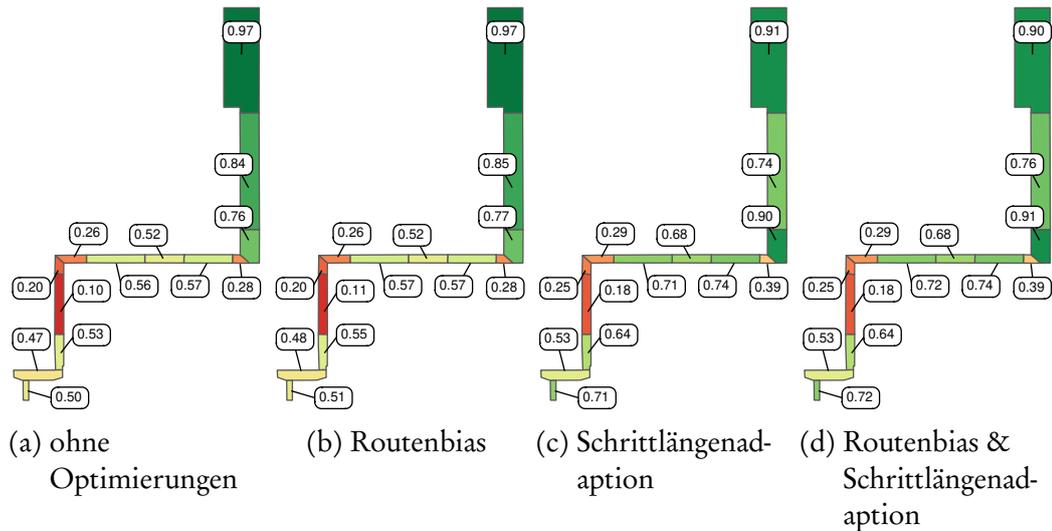


Abbildung 5.15: Trefferquote des Baseline-Systems in den einzelnen Routenabschnitten

Stationäre Phasen

Durch die mittels HAR erkannten stationären Phasen werden die vom Schrittkenner bereitgestellten Informationen komplementiert, wodurch sich im Idealfall die Anzahl der fälschlicherweise vorgenommenen Positionsupdates verringert. In den im Experiment erhobenen Daten wurden lediglich 2,7 % aller Schritte während (gemäß der HAR-Prädiktion) stationärer Phasen registriert. Bei der überwiegenden Anzahl der Testpersonen findet dies auf der gesamten Teststrecke nicht oder nur einige wenige Male statt; der Großteil der so erkannten Schritte entfällt auf wenige Durchläufe. Die Diskrepanz ist in diesen Fällen v. a. auf offensichtlich fehlerhafte Aktivitätserkennung zurückzuführen, bei der trotz Bewegung zeitweise stationäres Verhalten angenommen wurde.

Unter diesen Vorbedingungen überrascht es nicht, dass keine Verbesserung der Positionierungsgenauigkeit feststellbar ist. Die Trefferquote liegt bei 65,5 % ($SD = 16,6$ Prozentpunkte) und damit leicht, aber – wie ein durchgeführter Wilcoxon-Rangsummentest indiziert – nicht statistisch signifikant unter dem Baseline-System ($W = 228110, p < 0,2$). Die Alternativhypothese, dass durch Berücksichtigung der stationären Phasen die Positionierungsgenauigkeit steigt, kann somit nicht angenommen werden.

Standortbezogene Aktivitäten

Die standortbezogenen Aktivitäten teilen sich auf Treppensteigen, Türdurchgänge und Abbiegungen auf. Auf der getesteten Route besteht Abschnitt 12 (siehe

Abbildung 5.13) aus einer Treppe, welche auf halber Höhe einen kurzen ebenen Absatz enthält. Wie aufgrund der in Abschnitt 5.2.4.3 vorgestellten Ergebnisse zu erwarten ist, wurde das Treppensteigen bei allen Testpersonen erkannt. Die Aktivitätsverteilung im relevanten Abschnitt ist demnach vom Treppensteigen dominiert (78,0%), reguläres Gehen macht einen Anteil von 15,1% aus. Letzteres kommt teils durch das erwähnte flache Teilstück der Treppe, teils durch das anfängliche prinzipbedingte Hinterherhinken der Klassifikation um die Länge des betrachteten Fensters (in diesem Fall 3 s) zustande. Beide Aspekte werden aus Abbildung 5.16 ersichtlich.

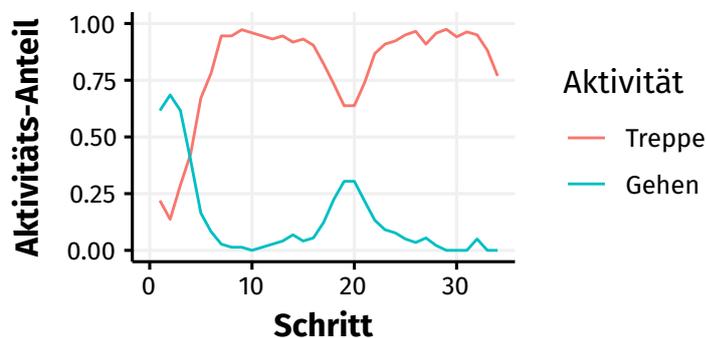


Abbildung 5.16: Erkannte Aktivitäten während des Stockwerkswechsels in Routenabschnitt 12

Die Erkennungsrate der Türdurchgänge liegt erwartungsgemäß niedriger. Anhand der Videoaufzeichnungen zeigt sich, dass pro Testperson im Median vier der sieben in der Route enthaltenen Türen zu relevanten Bewegungsmustern führen; in den restlichen Fällen standen die Türen offen oder wurden in Einzelfällen von anderen Personen offengehalten. Während der potentiell erkennbaren Türdurchgänge werden knapp 60% als solche erkannt, weitere 30% entfallen auf reguläres Gehen. Die Klassifikations-Konfidenz liegt in diesen Fällen im Mittel bei 0,80 (Tür) bzw. 0,81 (Gehen).

Um die oben beschriebenen Konfigurationsvarianten zu evaluieren, wird eine Simulationsreihe durchgeführt, bei der zwei für die HAR-Integration relevante Parameter – der Wirkungsbereich und der Anteil der direkt neu gezogenen Partikel – variiert werden. Die Werte für den Wirkungsbereich betragen 5, 10 und 20 m. Der Resampling-Anteil wird bei den Extremwerten von 0% (reine Gewichtung der Partikel) und 100% (Sampling aller Partikel anhand HAR) sowie zusätzlich bei 5, 10, 20 und 50% evaluiert.

Die besten Ergebnisse sind bei moderatem Resampling-Anteil (zwischen 5 und 50%) und Wirkungsradius ab 10 m zu beobachten; die Trefferquote liegt dort zwi-

schen 75,9 % und 77,4 % (bei einer Standardabweichung zwischen 18,2 und 18,7 Prozentpunkten). Dies stellt – so das Ergebnis eines durchgeführten Kruskal-Wallis-Rangsummentests mit anschließendem Dunnett-Tukey-Kramer-Test – eine statistisch signifikante Verbesserung gegenüber dem Baseline-System dar ($\chi^2(8) = 50$; $p < 0,01$; moderate Effektstärke ($\eta_H^2 = 0,07$)).

Die reine Gewichtung der Partikel hingegen führt zu keiner messbaren Verbesserung im Vergleich zum Baseline-System ($\chi^2(3) = 0,6$; $p = 0,9$). Die deutlichsten Ausreißer nach unten treten bei Resampling aller Partikel auf. Insbesondere bei einem hohen Wirkungsbereich steigt dort die Gefahr, die Positionsschätzung durch das Springen zu inkorrekten Aktivitätsorten zu beeinträchtigen.

Zusammenfassend kann die Alternativhypothese, dass durch Berücksichtigung von standortbezogenen Aktivitäten die Positionierungsgenauigkeit steigt, für einige Konfigurationen des Systems angenommen werden. Die Ergebnisse der Testreihe sowie des Baseline-Systems sind in Abbildung 5.17 zusammengefasst.

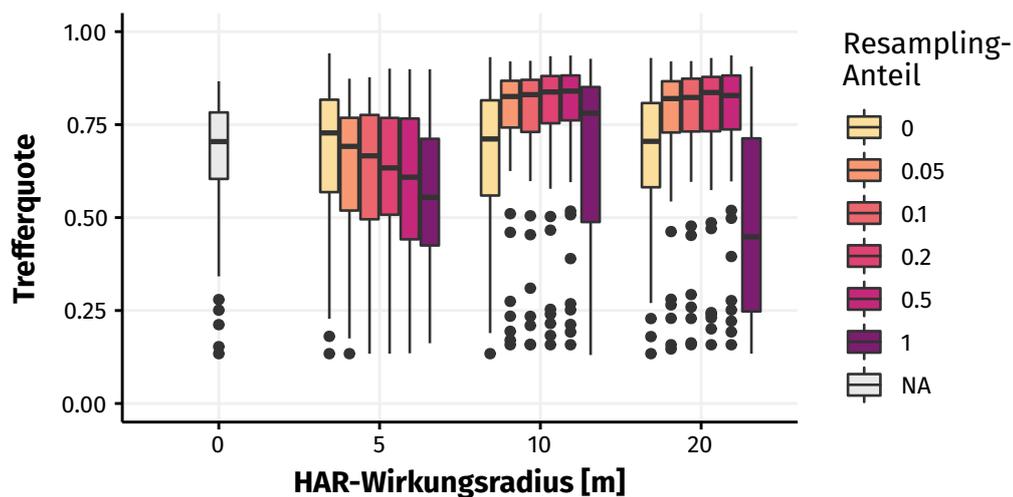


Abbildung 5.17: Trefferquote des Systems mit HAR-Einbindung

Ein genauerer Blick auf die einzelnen Routenabschnitte zeigt, dass in den performantesten Konfigurationen die Trefferquote im Vergleich zum Baseline-System nicht überall zunimmt, sondern teils auch unterhalb der zuvor erzielten Werte liegt. Das bessere Gesamtergebnis wird vor allem durch eine höhere Langzeitstabilität erzielt. Abbildung 5.18 stellt die abschnittsweise erreichten Trefferquoten des Baseline-Systems und einer HAR-Konfiguration gegenüber. Hier wird ersichtlich, dass insbesondere im mittleren Streckenabschnitt sowie gegen Ende der Route die Fähigkeit zur Selbstkorrektur zum Tragen kommt, was sich in teils hohen Trefferquoten von bis zu 80 % äußert.

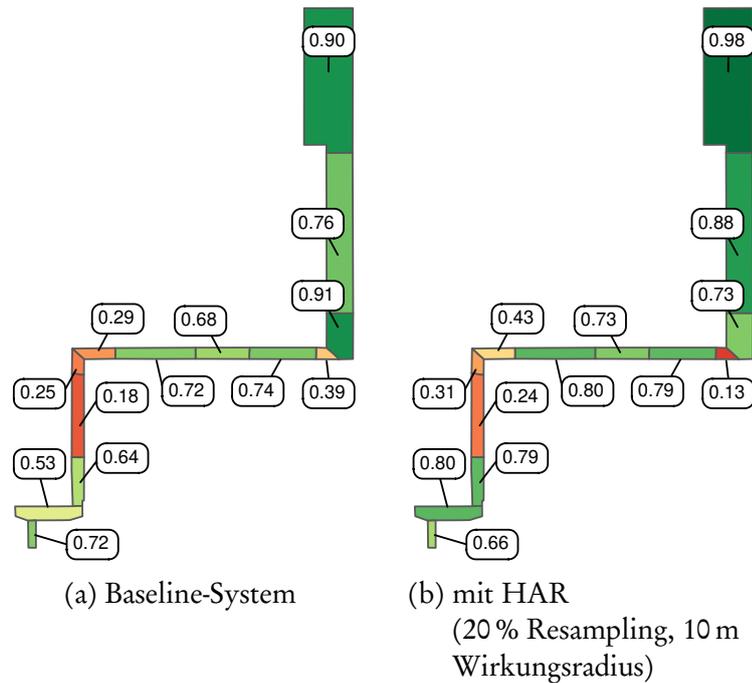


Abbildung 5.18: Vergleich der Trefferquote in den einzelnen Routenabschnitten

5.4.2.5 Detailstudie

Nach der allgemeinen Ergebnisübersicht besteht nun die Gelegenheit, anhand eines beispielhaft ausgewählten Simulationsdurchlaufs zu untersuchen, wie HAR bei der Positionierung hilft und welche Fehlerfälle auf diese Weise reduziert werden können. Zur Illustration bietet sich insbesondere die Positionskorrektur durch Türdurchgänge an.

Die Testroute führt vor der ersten Abbiegung (Routenabschnitt 2/3) zunächst lange geradeaus. Ohne Möglichkeit zur Korrektur der Schrittlänge streut die Partikelverteilung längs der Gehrichtung, wodurch sich zum Zeitpunkt der Abbiegung meist nur ein Teil der Partikel am korrekten Ort befindet.

Abbildung 5.19a zeigt beispielhaft den betroffenen Ausschnitt der Route. Korrekt zugewiesene Positionen werden durch blaue, im falschen Abschnitt befindliche Positionen durch rote Punktmarkierungen visualisiert. Zum Zeitpunkt **a** liegt die Positionsschätzung weit zurück; tatsächlich wurde hier soeben Abschnitt 2 betreten. Die unmittelbar folgenden Positionen liegen demnach im falschen Routenabschnitt und sind dementsprechend rot markiert. Bei **b** und **c** wird ein Türdurchgang detektiert, wodurch sich die Positionsschätzung erst leicht, dann sehr deutlich zum entsprechenden Ort korrigiert (im Beispiel werden 20 % der Partikel anhand der HAR-Information neu gezogen). Dies wird dadurch ermöglicht, dass sich einige der Partikel aufgrund der beschriebenen Streuung entlang der Gehrichtung

tung bereits in der Nähe der Tür befinden (siehe Partikelverteilung in Abbildung 5.19b). Diese Streuung ist zu Zeitpunkt d schließlich deutlich geringer, ebenso hat sich die Distanz zur Ground Truth verringert – erkennbar an der Anzahl der roten Positionsupdates in Abschnitt 4.

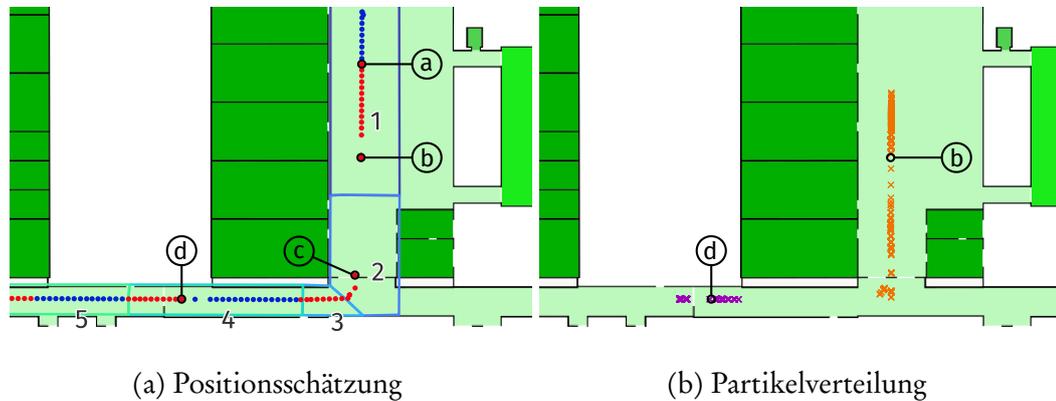


Abbildung 5.19: Beispiel für die Positionskorrektur durch Türerkennung

Problemfälle stellen Situationen dar, in denen die mittels PDR vorgenommene Positionsschätzung zum Zeitpunkt einer erkannten ortsbezogenen Aktivität bereits weiter als der konfigurierte Radius von der tatsächlichen Position abweicht. In diesen Situationen ist bestenfalls keine Positionskorrektur möglich, wodurch die mittels HAR gewonnene Information ungenutzt bleibt. Wird jedoch aufgrund der schon im Voraus abweichenden Positionsschätzung die falsche Landmarke herangezogen, so hat dies in aller Regel negative Auswirkungen auf die Positionsverfolgung im weiteren Verlauf der Navigation. Ähnlich verhält es sich, falls sich mehrere mögliche Landmarken im Umkreis der geschätzten Position befinden und zu Ambiguität hinsichtlich der tatsächlichen Position führen.

5.5 Fazit zur Aktivitätserkennung

Smartphonebasierte Aktivitätserkennung mithilfe von Machine-Learning-Methoden wurde in der Literatur bereits ausführlich untersucht. Im vergangenen Kapitel stand insbesondere der Einsatz im Kontext der Fußgänger-Indoornavigation im Fokus, wofür die üblicherweise verwendeten Aktivitäten ergänzt wurden. Dank der Vielzahl der in modernen Smartphones verfügbaren Gerätesensoren sind die groben Aktivitätsklassen meist gut unterscheidbar.

Wie die Ergebnisse des abschließenden Positionierungsexperiments zeigen, leistet die Einbindung von HAR einen Beitrag zur Langzeitstabilität der PDR-basierten Positionsverfolgung. Unter gewissen Voraussetzungen können somit die

Vorteile von infrastrukturabhängigen Verfahren, d. h. die Korrektur anhand von absoluten Standorten, genutzt werden.

Kapitel 6

Diskussion: Nutzerzentrierte Indoorpositionierungssysteme

Die drei vorigen Kapitel beleuchteten das Thema der nutzerzentrierten Indoor-Positionierung aus verschiedenen Richtungen: Zunächst stand die Entwicklung eines Basis-Positionierungssystems mitsamt Anpassungen für bessere Nutzeradaptation im Fokus. Anschließend wurden mithilfe von naturalistischen Nutzerstudien die Anforderungen an ein Positionierungssystem für den Realbetrieb in einer Navigationsanwendung erforscht, mit besonderem Augenmerk auf typische Aktivitäten während der Fußgängernavigation. Und schließlich wurde HAR für die Erkennung dieser Aktivitäten vorgestellt und eine darauf basierende Lösung in das Positionierungssystem integriert.

In diesem Kapitel werden die Ergebnisse der durchgeführten Studien zusammengetragen und gemeinsam im Hinblick auf die Forschungsfragen der Arbeit diskutiert. Wo immer es angemessen erscheint, werden auf Basis der gewonnenen Erkenntnisse Empfehlungen für das Design nutzerzentrierter Positionierungssysteme formuliert. Dabei fließen ebenso Erfahrungen mit ein, die während der Entwicklung und des Betriebs des Campus-Navigationssystems URwalking, insbesondere der Android-App, gewonnen wurden.

Die folgenden Abschnitte sind anhand der zu Beginn der Arbeit formulierten Forschungsfragen untergliedert.

6.1 Grenzen von reinen PDR-Systemen

Die erste Forschungsfrage beschäftigt sich mit der Fähigkeit von rein PDR-basierten Positionierungssystemen, die Indoorposition über längere Zeit zu verfolgen, sowie mit den Grenzen derartiger Lösungen.

Wie die Analyse des Baseline-Systems (ohne HAR) im abschließenden Positionierungsexperiment zeigt, ist auf einer für die Campusnavigation typischen Route auch ohne zusätzliche Korrektur durch externe Informationen in vielen Fällen adäquate Positionsverfolgung möglich. Eine signifikante Verbesserung wird durch die Anpassung an nutzerspezifische Eigenschaften, insbesondere durch die Adaption der Schrittlänge, erzielt. Neben höherer Trefferquote der einzelnen Routenabschnitte äußert sich dies v. a. in Form von deutlich reduzierter Varianz zwischen den Ergebnissen der einzelnen Testpersonen. Im Folgenden werden einige Ursachen diskutiert, die für die verbleibenden Fehlerfälle in Frage kommen.

6.1.1 Evaluationsmethodik

In den Positionierungsexperimenten in dieser Arbeit wurde eine aufgabenorientierte Evaluationsmetrik verwendet, die – verglichen mit dem üblicherweise in der Literatur beobachteten Vorgehen – stärkeren Fokus auf die Auswirkungen der Positionsbestimmung in einem Navigationskontext legt. Ein pragmatischer Vorteil des beschriebenen Verfahrens ist, dass die Ground Truth nicht kontinuierlich, d. h. bei jedem Positionsupdate bestimmt werden muss, sondern lediglich an den Grenzen zwischen Routenabschnitten. Dieser Ansatz bringt demnach die Voraussetzung mit sich, dass eine zuvor festgelegte Route existiert, die in Abschnitte unterteilt werden kann. Um der Grundidee der Metrik gerecht zu werden, müssen diese Abschnitte mit Navigationsanweisungen korrespondieren, welche überall innerhalb eines Abschnitts gültig und ausführbar sind.

Ein Teil der Fehlerfälle, in denen das Indoor-Positionierungssystem im abschließenden Positionierungsexperiment nicht den korrekten Routenabschnitt bestimmen kann, kommt als Artefakt der Evaluationsmethodik zustande: Sofern die mittels PDR geschätzte Position gegenüber der tatsächlichen Position vorausläuft oder hinterherhinkt – im Experiment ist meist Letzteres der Fall –, führt dies bei Abschnittswechseln zu vorübergehender Fehlklassifikation.

Der Effekt auf die berechnete Trefferquote des Positionierungssystems ist abhängig von der Anzahl bzw. der Länge der einzelnen Routenabschnitte. In einer Route, die aus sehr kurzen Abschnitten besteht, ist bei identischen sonstigen Bedingungen prinzipbedingt mit einer niedrigeren Trefferquote zu rechnen, während im gegenteiligen Fall – man stelle sich als Extremfall eine aus nur einem einzigen Abschnitt bestehende Route vor – die Möglichkeit von Fehlklassifikationen abnimmt. Dies bildet die Navigationsrealität nach: Im Falle komplexer Routen, die durch viele einzelne Navigationsanweisungen mit dazugehörigen Abschnitten beschrieben werden – sei es automatisiert oder intellektuell –, ist für die Bereitstellung der korrekten Anweisung eine höhere Positionierungsgenauigkeit vonnöten.

Die erzielten Werte sind daher nur dann zwischen unterschiedlichen Routen oder Umgebungen vergleichbar, wenn die Routenabschnitte ähnlich groß gewählt werden oder das Ergebnis anhand ihrer Größe normiert wird. Am sinnvollsten einsetzbar erscheint die Metrik, wie mehrfach in dieser Arbeit geschehen, für den Vergleich von Systemvarianten auf ein- und derselben Route.

Die Einteilung der Route in disjunkte Abschnitte vereinfacht die Berechnung der Positionierungsgenauigkeit, stellt aber eine Abstraktion dar: In der Realität existieren für einen Standort oft mehrere, unterschiedlich präzise Navigationsanweisungen, die zu Fortschritt entlang der Route führen. Beispielsweise kann bereits auf eine Tür am Ende eines Korridors verwiesen werden, bevor etwaige Landmarken auf dem Weg dorthin passiert worden sind. Navigationsanweisungen sind daher für menschliche BenutzerInnen nicht immer nur genau innerhalb des korrespondierenden Abschnitts nützlich. Die Übergänge dazwischen sind insbesondere dann fließend, wenn Abschnittsgrenzen nicht durch architektonische Einheiten (z. B. Abbiegungen, Türen) definiert sind, sondern anhand von entlang der Route gelegenen Landmarken, deren Sichtbarkeit graduell zunimmt. Derartige Details, die in hohem Maße von den tatsächlichen Gegebenheiten vor Ort und zusätzlich von nutzerspezifischen Eigenschaften abhängen, können in einer allgemein anwendbaren Evaluationsmetrik nur schwer berücksichtigt werden.

Die Reduzierung der Positionierungsgenauigkeit auf einen einzelnen Wert führt zwar zu einfacher Vergleichbarkeit (mit den oben genannten Einschränkungen), kann aber nicht die komplette Charakteristik des Positionierungssystems abbilden. Diese Abwägung zwischen Prägnanz und Aussagekraft muss auch bei den traditionell verwendeten Metriken vorgenommen werden: Während sich beispielsweise das 75%-Quantil des (euklidischen) Positionsfehlers gut eignet, um die Ergebnisse eines Wettbewerbs zu sortieren, werden in der Literatur üblicherweise ECDF-Diagramme generiert, um die Fehlercharakteristik eines Positionierungsverfahrens zu illustrieren. Es ist daher keineswegs die Intention, mit der aufgabenorientierten Evaluation die etablierten Metriken abzulösen, sondern vielmehr – wo sinnvoll – eine zusätzliche Informationsquelle und Grundlage für weitere Interpretationen bereitzustellen.

6.1.2 Modellierung der Umgebung

Im URwalking-Projekt wurde ein graphbasiertes Modell des Regensburger Campus entwickelt, welches als Datengrundlage für effiziente Routenberechnung und Positionierung dient. Bei der Erstellung eines derartigen Modells gilt es abzuwägen, mit welchem Detailgrad die Umgebung abgebildet werden soll. Schmale Korridore mit einer Längskante zu approximieren und somit die Freiheitsgrade zu re-

duzieren, ist ein gängiges Vorgehen (siehe z. B. Hilsenbeck et al., 2014, S. 4). Die Modellierung von offenen Flächen lässt mehrere unterschiedliche Ansätze zu: Eine Beschränkung auf wenige Kanten – wie etwa bei Lin Liao et al. (2003) zu sehen – hat den Vorteil eines geringeren Aufwands bei der Routenberechnung. Bildet man viele Pfade möglichst exakt nach, ist dies im Umkehrschluss der realitätsgetreuen Positionsverfolgung zuträglich, sorgt jedoch auch für höheren Modellierungs- und Rechenaufwand.

Prinzipbedingt schränkt jede Abstraktion von der Realität auf eine Graphstruktur die Möglichkeit des Positionierungssystems ein, beliebige Trajektorien durch den Raum zu verfolgen. Um diesen Zielkonflikt aufzulösen, kommt im vorliegenden System die beschriebene hybride Struktur zum Einsatz, bei der in offenen Flächen der Graph nicht zur Positionskorrektur herangezogen wird. Wie in der zweiten Vorstudie deutlich wurde, führt dies im Vergleich zu direkt verbundenen Knoten oder einer Mesh-Struktur zu einer höheren Positionierungsgenauigkeit. Da sich ohne Stabilisierung durch den Graph die Unsicherheit der Positionsbestimmung akkumuliert, ist dies nur so lange skalierbar, wie die Randknoten der eingegrenzten Fläche zuverlässig erreicht werden können, um die graphbasierte Positionierung wiederaufzunehmen.

In der ersten Vorstudie wurde deutlich, welchen Einfluss die Modellierung von realitätsnahen Pfaden im Graphmodell auf die Fähigkeit zur Positionsverfolgung in demselben haben kann. Durch eine einfache Modifikation – das Hinzufügen von schrägen Kanten an einer problematischen Abbiegung – erhöhte sich die Trefferquote in allen nachfolgenden Abschnitten. Während derartige Änderungen für den Zweck einzelner Positionierungsexperimente schnell durchführbar sind, ist eine einheitliche Umsetzung für das komplette Campus-Modell deutlich aufwändiger und bringt die Gefahr mit sich, implizite Annahmen des Routing-Algorithmus hinsichtlich der Beschaffenheit des Graphs zu verletzen. Weiterhin wird das Modell mit steigender Realitätstreue der möglichen Pfade komplexer, was im Widerspruch zum Ziel steht, eine möglichst abstrakte Repräsentation der Umgebung zu erzeugen. Dieser Zielkonflikt wird später in Abschnitt 6.4.3 im Zusammenhang mit HAR erneut aufgegriffen.

6.2 Navigationsverhalten und Nutzung der URwalking-App

Die mithilfe der URwalking-Android-App durchgeführten naturalistischen Studien haben zu Erkenntnissen über das Nutzungs- und Navigationsverhalten geführt, die im Folgenden diskutiert werden. Das Augenmerk liegt dabei auf den

Unterschieden zwischen dem Verhalten in Experimentsituationen und unter realen Bedingungen sowie auf den Konsequenzen für das Design eines Indoor-Positionierungssystems.

6.2.1 Navigationsitzungen allgemein

Bei Analyse der zeitlichen Verteilung der Navigationsitzungen wurden sowohl längerfristige Muster – die deutlichen Peaks zu Semesterbeginn – als auch wöchentliche Periodik in Übereinstimmung mit den Vorlesungszeiten sichtbar. Dies ist ein deutlicher Hinweis darauf, dass die Nutzerschaft der Campus-Navigations-App – wie nicht anders zu erwarten – zu einem großen Teil aus Studierenden besteht.

Bemerkenswerter hingegen ist die Erkenntnis, dass viele der App-Sitzungen keine Navigation im eigentlichen Sinn enthalten, sondern lediglich zum Planen von Routen im Voraus dienen. Dies ist vor allem daran erkennbar, dass sich die NutzerInnen fernab vom Startpunkt der Route, oft auch abseits des Campus befinden.

Die Mehrheit der geplanten Routen umfasst Stockwerkswechsel oder Gebäude- bzw. Indoor-/Outdoor-Übergänge. Diese stellen, sofern sie zuverlässig erkannt werden können, neben Türen und Abbiegungen Möglichkeiten zur Stabilisierung der Positionsschätzung dar. Die Routenlänge beträgt meist weniger als 500 m und ist damit geringer als die 544 m bzw. 1 km in den jüngsten IPIN-Wettbewerben (Potorti et al., 2020; Renaudin et al., 2019).

6.2.2 Pausen und Unterbrechungen

Die Analyse von Daten des Schritterkenners und des Vorder- und Hintergrundstatus der Navigations-App führte zu Erkenntnissen über Pausen und Unterbrechungen im Navigationsvorgang. Diese treten in einem Großteil der Sitzungen auf und sind als Teil des durch *Locomotion* und *Wayfinding* charakterisierten Navigationsprozesses ein erwartetes Phänomen. Zwischen der realen App-Nutzung und dem Verhalten in den kontrollierten Positionierungs-Studien herrscht hinsichtlich der Gehpausen eine deutliche Diskrepanz; im ersten Fall sind mehr und längere Pausen zu verzeichnen. Eine naheliegende Erklärung hierfür sind die Experimentsituation und die Anwesenheit der TestleiterInnen, die bei größerer Unsicherheit Hilfestellung leisteten.

Ebenso ist es dem Experimentaufbau geschuldet, dass unter kontrollierten Bedingungen die App zu keiner Zeit im Hintergrund läuft – im realen Betrieb machen diese Perioden einen beträchtlichen Anteil von durchschnittlich knapp 30 % aus. Eine mögliche Ursache für diese Unterbrechungen ist die vorübergehende Nutzung von anderen Apps, die vom Anwender höher priorisiert werden als die

laufende Navigation, beispielsweise eingehende Anrufe oder Textnachrichten. Da in den kontrollierten Experimenten nicht die privaten Geräte der Testpersonen verwendet wurden, entfällt diese Option. Über die tatsächlichen Gründe für die Unterbrechungen kann die naturalistische Studie keinen Aufschluss geben, es lassen sich jedoch einige grundlegende Fälle unterscheiden:

Findet während der Hintergrund-Periode eine Bewegung entlang der geplanten Route in Richtung Ziel statt, so kann angenommen werden, dass zu diesem Zeitpunkt keine Navigationsassistentz benötigt wird. Eine grobe Positionsverfolgung, die den Fortschritt entlang der Route prüft, ist in diesem Fall ausreichend. Ein mögliches Szenario ist die Navigation zu einem Ziel in einem anderen Gebäude(-komplex), bei der man zwar den Weg zum Gebäude kennt, nicht aber zum Zielort innerhalb des Gebäudes. Findet keine Bewegung statt, so deutet dies auf etwaige momentane Ablenkung hin, aber nicht zwangsläufig auf den vorzeitigen Abbruch des Navigationsvorgangs. Für das Positionierungssystem hat derartige Verhalten zunächst keine Auswirkungen, solange sichergestellt ist, dass durch erratische Bewegungen keine Schritte erkannt werden. Wird die Route verlassen, während sich die Navigations-App im Hintergrund befindet, kann dies entweder auf einen vorsätzlichen Abbruch des Navigationsvorgangs oder aber auf ein versehentliches Verlaufen zurückzuführen sein. In diesem Fall erscheint eine explizite Interaktion per Benachrichtigung o. Ä. sinnvoll, um gegebenenfalls die Positionsverfolgung zu beenden.

6.2.3 Geräteposition und Trageart

Durch Auswertung von Sensordaten und Anwendung der trainierten ML-Modelle wurde die Art und Weise, wie das Smartphone während des Navigationsvorgangs getragen wird, untersucht. Ebenso wie zuvor bei den Unterbrechungen ist auch hier eine Diskrepanz zwischen dem unter realen Bedingungen und in kontrollierten Experimenten beobachteten Verhalten feststellbar.

Aus den naturalistischen Logdaten geht hervor, dass Personen das Gerät zu meist in der Hand und vor dem Körper halten, aber auch andere Positionen wie z. B. das Tragen in einer Tasche vorkommen.

In den kontrollierten Studien hingegen befand sich das Gerät nahezu in allen Fällen in der Hand und vor dem Körper, obwohl den Testpersonen freigestellt war, es nach Belieben zu tragen. Die naheliegende Erklärung ist wiederum, dass die Testpersonen in der Experimentsituation möglichst keine Anweisungen verpassen wollten und ihre gesamte Aufmerksamkeit dem Navigationstask widmeten.

Aufgrund der Tatsache, dass sich das Gerät fast ausnahmslos in derselben Position befand, kam ein Teilaspekt des entwickelten HAR-Verfahrens, namentlich die

vorgeschaltete Erkennung der Geräteposition mit anschließender Auswahl von spezifisch dafür trainierten Modellen, nicht zum Tragen.

6.3 Aktivitätserkennung mithilfe von Gerätesensoren

Die nächste Forschungsfrage widmete sich der Erkennung von navigationsspezifischen Aktivitäten anhand der von den Gerätesensoren gelieferten Rohdaten. Insbesondere soll hier die erreichte Genauigkeit und das vorgestellte Konzept der hierarchischen Klassifikation kritisch diskutiert werden.

6.3.1 Klassifikation der Geräteposition

Die Unterscheidung von unterschiedlichen Gerätepositionen ermöglicht es dem Positionierungssystem, auf die jeweiligen Umstände zu reagieren und das PDR-System entsprechend zu adaptieren. Weiterhin kann diese Information im Kontext eines Navigationssystems dazu genutzt werden, die Modalität von Navigationsanweisungen an die aktuelle Situation anzupassen. Wie die Ergebnisse aus Abschnitt 5.2.4.1 zeigen, ist die Unterscheidung der vier in dieser Arbeit betrachteten Gerätepositionen im erhobenen HAR-Datensatz sehr zuverlässig möglich. Dies deckt sich mit den Erfahrungen aus der Literatur, in der diese Klassifikation – teils auch mit deutlich einfacheren Mitteln – bereits zahlreiche Male erfolgreich erprobt wurde (siehe z. B. Fujinami & Kouchi, 2013).

Die vier in dieser Arbeit verwendeten und aus der Literatur abgeleiteten Tragearten können freilich nicht das komplette Spektrum der in der Realität auftretenden Verhaltensweisen abbilden. Dies wird dadurch deutlich, dass bei Anwendung der ML-Modelle auf den naturalistischen Datensatz ein beträchtlicher Anteil nicht mit hoher Konfidenz klassifiziert werden kann. Für den praktischen Einsatz sollte der Fokus daher darauf liegen, die am häufigsten vorkommenden Gerätepositionen und Tragearten möglichst zuverlässig zu erkennen.

6.3.2 Klassifikation der Aktivitäten

Für die Klassifikation der navigationsspezifischen Aktivitäten wurden mehrere Varianten erprobt, die auf unterschiedlichen Granularitätsstufen arbeiten. Der flache 9-Klassen-Klassifikator hatte teils deutliche Probleme, insbesondere eng verwandte Aktivitäten wie das Durchschreiten von Türen in verschiedene Richtungen zu

unterscheiden. Die vier gröber aufgelösten Klassen konnten erwartungsgemäß zuverlässiger differenziert werden: Stationäres Verhalten und Treppensteigen sind nahezu eindeutig erkennbar, während Verwechslungen hauptsächlich zwischen regulärem Gehen und dem Durchschreiten von Türen auftreten. Die erreichte Genauigkeit der in der Hierarchie nachgeschalteten Klassifikatoren variiert je nach spezifischer Aktivität, Geräteposition und Länge des betrachteten Zeitfensters.

Im Allgemeinen erweist sich HAR in der vorgenommenen Studie dann am zuverlässigsten, wenn sich das Gerät in der Hand vor dem Körper befindet. Dies ist insofern erfreulich, da es sich dabei sowohl im realen Betrieb der Navigations-App als auch in den kontrollierten Experimenten um die vorherrschende Trageart handelt.

Mit Ausnahme des Treppensteigens ist die Unterscheidung zwischen stationären Aktivitäten in allen Fällen deutlich zuverlässiger möglich als zwischen Aktivitäten, die auf Fortbewegung basieren. Insbesondere beim regulären Gehen ist eine große Schwankungsbreite zwischen Klassifikationsgenauigkeiten von unter 50 % bis zu knapp 90 % beobachtbar. Eine Erklärung hierfür sind die unterschiedlichen Charakteristiken der Aktivitäten: Während Stillstehen/Umschauen, Geradeausgehen oder Treppensteigen aus längeren, teils periodischen Bewegungsabläufen bestehen, sind Abbiegungen und Türdurchgänge einmalige, vergleichsweise kurze Aktivitäten. Letztere gehen zwangsweise mit Schrittbewegungen einher, welche je nach Fenstergröße mehr oder weniger in die Klassifikation mit einfließen und somit die Trennschärfe zwischen den Klassen verringern.

Unter den nachgeschalteten Klassifikatoren am wenigsten zuverlässig ist die Unterscheidung zwischen dem Drücken und Ziehen von Türen – in den meisten Fällen nicht besser als ein Münzwurf. Die Bewegungsmuster sind offenbar zu ähnlich, als dass sie anhand von vom Körper entkoppelten Sensoren unterschieden werden können. Die Idee der Erkennung der Türrichtung war von vornherein experimenteller Natur, da derartige Informationen zum jetzigen Zeitpunkt nicht im URwalking-Umgebungsmodell enthalten sind, weswegen das erzielte Resultat nicht allzu sehr enttäuscht. Für das Positionierungssystem ist vielmehr die Information wichtig, dass überhaupt eine Tür passiert wurde, um auf diese Weise eine Positionskorrektur einzuleiten.

Die Treppen-Erkennung inklusive der Richtung hingegen arbeitet äußerst zuverlässig. Dies ist in erster Linie dem Barometer geschuldet, welches in diesem Fall als dedizierter Sensor für Höhenänderungen dient. Der ohne Barometer durchgeführte Test zeigt jedoch, dass auch andernfalls anhand der restlichen Sensorik akzeptable Erkennungsraten möglich sind. Einschränkend muss hier erwähnt werden, dass das Treppensteigen als einzige Form des Stockwerkswechsels im Datensatz enthalten ist und sich die hohe Klassifikationsgenauigkeit durch Berücksich-

tigung weiterer Varianten (z. B. Aufzüge oder Rolltreppen) voraussichtlich reduzieren wird.

HAR-Verfahren aus der Literatur erreichen für manche Datensätze eine Klassifikationsgenauigkeit von bis zu 95 % (J. Wang et al., 2019, S. 8). Die meisten der hier erzielten Resultate liegen deutlich unter diesem Wert, was jedoch nicht durch die Modelle erklärbar ist – schließlich wurden neben dem eigens angepassten LSTM-Modell vier State-of-the-Art-Verfahren verwendet, die in anderen Studien erfolgreich eingesetzt wurden. Vielmehr sind die Ergebnisse durch den Datensatz zu erklären, der einige sehr ähnliche und – je nach Fenstergröße – teils nur schwer unterscheidbare Klassen enthält. Auch die beschriebene Einbettung von kurzen Aktivitäten in längere Gehbewegungen trägt dazu bei, dass diese prinzipbedingt schwieriger zu erkennen sind. Da diese Problematik jedoch auch im realen Betrieb existiert, erscheint es wenig sinnvoll, dies auf Ebene des Datensatzes anders zu handhaben.

Ein positiveres Bild zeichnet der Vergleich der Ergebnisse mit einer realitätsnahen HAR-Studie, wie sie beispielsweise von Fetzer et al. (2018) durchgeführt wurde. Die Klassifikationsgenauigkeit der dort betrachteten vier Aktivitäten (Stehen, Gehen, Aufwärts-/Abwärtsbewegung) liegt über mehrere Testläufe hinweg insgesamt bei knapp 70 %, einzelne Aktivitäten werden teils mit über 80 % erkannt. Diese Werte bewegen sich im Rahmen des in dieser Arbeit erzielten Resultats und indizieren, dass Aktivitätsmuster unter realistischen Bedingungen heterogener sind, als es durch einen kontrolliert erhobenen Datensatz abgebildet werden kann. Dies wird auch bei Anwendung der Klassifikatoren auf die naturalistischen Daten der URwalking-App deutlich, bei der nur ein Teil mit hoher Konfidenz klassifiziert werden kann. Mögliche Ursachen dafür sind das Vorhandensein noch anderer, nicht berücksichtigter Aktivitäten oder auch eine mangelnde Generalisierbarkeit der Modelle.

6.4 Anwendung von HAR in der Indoor-Positionierung

Zuletzt werden die Ergebnisse der mit HAR durchgeführten Positionierungsexperimente und die dort auftretenden Problematiken diskutiert. Ein besonderes Augenmerk gilt den Auswirkungen von falschpositiven und falschnegativen Klassifikationen.

6.4.1 Trefferquote allgemein

Die in dieser Arbeit mit HAR-Unterstützung maximal erreichte Trefferquote des Positionierungssystems im abschließenden Experiment liegt knapp unter 80 %. Dies stellt eine deutliche Verbesserung gegenüber dem Baseline-System dar, ist aber dennoch ein Stück weit entfernt von einem perfekten Ergebnis, d. h. hundertprozentiger Trefferquote auf allen Abschnitten. Letzteres ist unter realen Bedingungen freilich nur schwer erreichbar und für eine adäquate Navigationsassistenz nicht zwingend notwendig. Schließlich bezweifelt kaum jemand den Nutzen von satellitengestützter Navigation im Außenbereich, obwohl auch dort die Positionsbestimmung nicht immer zuverlässig arbeitet. Man bedenke ebenfalls, dass die meisten ProbandInnen im Experiment mit simulierter Positionierung ans Ziel gefunden haben; Probleme wurden vielmehr durch unklare Navigationsanweisungen verursacht.

Im Fall der erhobenen Daten, so zeigen es die Ergebnisse der Simulationsstudie, führt die Erkennung von stationären Phasen durch HAR-Verfahren nicht zu einer höheren Positionierungsgenauigkeit. In einigen Fällen verringert offensichtlich fehlerhafte Aktivitätserkennung sogar die Fähigkeit des Systems zur Positionsverfolgung. Dass im Durchschnitt pro Testperson nur wenige vom Schritterkennner detektierte Schritte mithilfe der Aktivitätserkennung entfernt werden, zeigt, dass der eingebaute Schritterkennner meist zuverlässig zwischen sporadischen Bewegungen und tatsächlichem Gehen unterscheiden kann.

Alleine durch Schritterkennung ist jedoch keine Unterscheidung von verschiedenen stationären Verhaltensweisen (Stillstehen und suchendem Umschauen) möglich, was mithilfe von HAR mit gewissem Erfolg demonstriert wurde. Für die Positionsbestimmung selbst ist dieser Anwendungsfall weniger relevant, im Kontext einer Navigations-App aber sinnvoll, da damit Rückschlüsse auf das Verständnis von Navigationsanweisungen gezogen werden können.

Die Einbindung von standortgebundenen Aktivitäten in das Positionierungssystem bringt eine messbare Verbesserung der Positionierungsgenauigkeit sowie höhere Langzeitstabilität. Da durch das entwickelte HAR-Verfahren jedoch nur Klassen von Landmarken, nicht aber einzelne Objekte detektiert werden, liefert es keine absolute Position, sondern ermöglicht lediglich die Eingrenzung derselben auf einige Standorte. Aus diesem Grund ist eine einigermaßen akkurate Positionsverfolgung mittels PDR unerlässlich, um die durch HAR gewonnenen Informationen sinnvoll einbinden zu können; andernfalls kann sich das Verfahren potenziell auch kontraproduktiv auswirken.

6.4.2 Falschpositive und falschnegative Klassifikation

Die Implementierung des HAR-Verfahrens im Positionierungssystem sieht eine Höhergewichtung von Partikeln bei Erkennung von standortgebundenen Aktivitäten vor. Dies beeinflusst die Art und Weise, wie sich Fehler 1. und 2. Art auf die Positionierung auswirken:

Falschpositive Klassifikation – z. B. das Erkennen von Türdurchgängen oder Treppensteigen, während der Nutzer tatsächlich geradeaus geht – hat immer dann keine negativen Auswirkungen, wenn sich kein relevanter Aktivitätssort in der Nähe befindet; ist dies jedoch der Fall, wird die Positionsschätzung verfälscht. Ein Lösungsansatz hierfür ist die gezielte Anpassung des Konfidenz-Schwellenwerts, ab dem die HAR-Information genutzt wird, oder aber ein an das Umgebungsmodell abgestimmter Aktivitätsradius. Um den Einfluss sporadischer Fehlklassifikationen zu vermindern, sind optional auch zeitliche Glättung oder das Abwarten von mehreren aufeinanderfolgenden Treffern denkbar.

Fehler 2. Art, also z. B. das Nichterkennen von Treppen oder Türen trotz tatsächlich vorliegender Aktivität, haben aufgrund der konkreten Implementierung keine unmittelbaren negativen Auswirkungen. Das Passieren von Tür- oder Treppenknotten ist auch dann möglich, wenn keine korrespondierende Aktivität vorliegt. Dies ist einerseits aufgrund der nicht komplett verlässlichen Klassifikation, andererseits aufgrund der Möglichkeit von offenstehenden Türen oder Treppen mit ebenen Abschnitten notwendig. Als Folge der falschnegativen Klassifikation wird jedoch die Chance zur Positionskorrektur verpasst, wodurch mehr Unsicherheit als bei korrekter Aktivitätserkennung herrscht.

6.4.3 Einbindung in das Umgebungsmodell

Zuletzt soll an dieser Stelle ein praktischer Aspekt diskutiert werden, der bei der Verwendung einer gemeinsamen Datengrundlage für Routenberechnung, Positionierung und zur Generierung von Navigationsanweisungen auftritt.

Dieser Ansatz hat den Vorteil, dass während der Navigation ein unmittelbarer Zusammenhang zwischen der geschätzten Position und dem aktuellen Navigationsschritt besteht. Eine zu lösende Problematik ist jedoch die Übereinstimmung der Komplexität der berechneten Route bzw. der dafür erzeugten Navigationsanweisungen mit dem tatsächlich zu absolvierenden Pfad. Oft herrscht ein eindeutiger Zusammenhang, beispielsweise wenn die Anweisung »Gehen Sie geradeaus« einer oder mehrerer Kanten in einem Korridor entspricht, wenn in einem flächigen Indoor-Areal »Biegen Sie links ab« gefordert wird oder wenn »Gehen Sie durch die Tür« eindeutig bei einer Verbindungstür verortet ist.

Andere Bereiche erfordern komplexere Handlungen, die jedoch – aus nachvollziehbaren Gründen – üblicherweise nicht explizit in den Navigationsanweisungen formuliert werden. Stockwerkswechsel in Form von Treppen oder Aufzügen werden beispielsweise in URwalking durch »*Nehmen Sie die Treppe und gehen Sie 2 Stockwerke nach oben in das 2. Stockwerk*« bzw. »*Fahren Sie mit dem Aufzug in das 3. Stockwerk (2 Stockwerke nach oben)*« eingeleitet. Um derartige Anweisungen zu generieren, reicht eine sehr abstrakte Modellierung der Umgebung aus. Im Falle des URwalking-Graphs werden Treppenhäuser und Aufzüge als mit entsprechendem Typ versehene Kanten zwischen Knoten zweier Stockwerke abgebildet. Für BenutzerInnen des Navigationssystems bedingt dies jedoch insbesondere bei einer Differenz von mehreren Stockwerken unterschiedliche Bewegungsabläufe. Abgesehen von der offensichtlichen Diskrepanz zwischen dem Aufzugfahren und Treppensteigen erfordert Letzteres je nach konkreter Ausprägung des Treppenhauses beispielsweise das Wenden auf Treppenabsätzen auf halber Höhe oder das Umrunden des Treppenhauses auf jeder Ebene, um zu den nächsten Treppenstufen zu gelangen; unter den im URwalking-Graph verzeichneten Treppen befinden sich sogar Wendeltreppen, welche wiederum andere Bewegungsmuster erzeugen. Die atomare Navigationsanweisung entspricht in solchen Fällen also einer Abfolge von Aktivitäten, die sich nicht auf die vereinfachte Repräsentation der Realität abbilden lässt. Auch Brush et al. (2010, S. 81) beschäftigen sich mit dieser Problematik und kommen zum Schluss, dass ein zu hoher Detailgrad der Navigationsanweisungen für die BenutzerInnen hinderlich ist.

Für das PDR-basierte Positionierungssystem jedoch ist, wie in dieser Arbeit experimentell gezeigt, ein möglichst realitätsgetreues Umgebungsmodell von Vorteil. Im Falle besonders charakteristischer Aktivitätsorte bietet sich perspektivisch sogar die Gelegenheit, anhand ihrer Eigenschaften mithilfe von geeigneten HAR-Verfahren noch konkretere Positionsangaben zu generieren. Schlussendlich führt diese Problematik zu einem ähnlichen Zielkonflikt, wie er bereits in Abschnitt 6.1.2 beschrieben wurde: Einer realitätsgetreuen Modellierung der Umgebung und aller dort ausführbarer Aktivitäten steht das Streben nach einer auf die grundlegenden baulichen Merkmale reduzierten Repräsentation entgegen. In dieser Arbeit wurde gezeigt, dass es dennoch möglich ist, eine gemeinsame Datenstruktur für die Routenberechnung, Positionierung und zur Generierung von Navigationsanweisungen zu verwenden. Je nach konkretem Anwendungsfall erscheint es jedoch sinnvoll, mit unterschiedlichen Detailstufen zu arbeiten bzw. den Abstraktionsgrad anzupassen.

Kapitel 7

Fazit und Ausblick

Im letzten Kapitel dieser Dissertation werden zunächst die Ergebnisse zusammengefasst, gefolgt von der Einordnung des wissenschaftlichen Beitrags und einer kurzen Diskussion der Limitierungen der Arbeit. Ein Ausblick auf noch offene Fragen und mögliche anknüpfende Untersuchungen schließt das Kapitel ab.

7.1 Zusammenfassung

In dieser Arbeit wurde die Entwicklung und Evaluation eines Indoor-Positionierungssystems für die smartphonebasierte Fußgängernavigation im URwalking-Projekt beschrieben. Spezieller Fokus lag darauf, reales Nutzerverhalten zu analysieren und die während der Navigation auftretenden Aktivitäten vorteilhaft für die Positionierung zu nutzen.

Durch die Vorarbeiten aus dem URwalking-Projekt existierten einige Randbedingungen, die bei der Entwicklung des Systems berücksichtigt werden mussten und die Designentscheidungen des Positionierungssystems beeinflussten. Insbesondere ist hier das vorhandene graphbasierte Umgebungsmodell zu nennen, welches im Basis-System für die Stabilisierung der mithilfe von PDR vorgenommenen Positionsschätzung sorgt. Durch Anwendung des etablierten Partikelfilter-Verfahrens implementierte Erweiterungen, allen voran die nutzerspezifische Adaption an variable Schrittlängen, verbesserten die Fähigkeit des Systems, die Position zu verfolgen. Dies wurde in Simulationsstudien untersucht, auf Grundlage von unter realistischen Navigationssituationen erhobenen Sensordaten und unter Anwendung einer an den Navigationstask angepassten Evaluationsmetrik.

Ergänzend zu den kontrollierten Studien wurde das Nutzerverhalten unter realen Bedingungen mithilfe der entwickelten Campus-Navigationsapp URwalking in einer naturalistischen Log-Studie erforscht. Hierbei zeigte sich zunächst, dass

nur ein Teil der Sitzungen tatsächliche Navigation beinhaltet. Die Charakteristik der angefragten Routen, insbesondere die Häufigkeit des Auftretens von Stockwerkswechseln und Türen, ermöglicht es, die Anforderungen an die Langzeitstabilität der Positionsverfolgung einzuschätzen. Die Auswertung der Sensordaten ergab, dass während der Navigation verschiedene Gerätepositionen vorherrschen und dass der Fortschritt auf der Route immer wieder durch Gehpausen unterbrochen wird.

Um die für die Positionierung nutzbaren ortsbezogenen Aktivitäten systematisch auszuwerten, wurden anschließend unter Anwendung von State-of-the-Art-Methodik HAR-Modelle trainiert. Grundlage dafür war ein separater Datensatz, der indoornavigationsspezifische Aktivitäten in Kombination mit mehreren Gerätepositionen enthält. Nach Anwendung dieser Modelle auf die naturalistischen Logdaten wurden sie in das Basis-Positionierungssystem integriert. Die abschließende Evaluation in einer realitätsnahen Navigationssituation zeigte, dass sich die Positionierungsgenauigkeit durch gezielte Berücksichtigung von ortsbezogenen Aktivitäten signifikant verbessert und sich die Fehlerfälle deutlich verringern. Unter gewissen Voraussetzungen verfügt das System somit auch ohne externe Infrastruktur über die Fähigkeit zur Selbstkorrektur.

7.2 Wissenschaftlicher Beitrag

Nach der inhaltlichen Zusammenfassung resümiert dieser Abschnitt noch einmal separat den wissenschaftlichen Beitrag dieser Arbeit.

Graphbasiertes Positionierungssystem mit HAR-Integration

Das in dieser Arbeit entwickelte Indoor-Positionierungssystem arbeitet auf Basis einer vergleichsweise schnell und günstig erstellbaren Datengrundlage. Hinsichtlich der Auflösung der Graphstruktur herrschen keine besonderen Voraussetzungen, solange diese die begehbaren Wege in der modellierten Umgebung abbildet. Während im Verlauf der Arbeit der Regensburger Universitätscampus als Testareal diente, ist das Verfahren somit auch problemlos auf weitere Umgebungen übertragbar. Das Positionierungsverfahren wurde so konzipiert, dass es auf Standard-Smartphones lauffähig ist. Es basiert auf bewährten PDR- und Partikelfilter-Verfahren und kombiniert sie mit Deep-Learning-gestützter Aktivitätserkennung zur Stabilisierung der Positionsschätzung.

Während die einzelnen Komponenten des Verfahrens – graphbasierte Modelle, partikelfiltergestützte Positionierung, HAR – aus der Literatur bekannt sind, so

stellt die konkrete Umsetzung in einem real einsetzbaren System den eigentlichen Beitrag dar.

Datensatz und Methodik für realitätsnahe Positionierungsexperimente

Dass das entwickelte System unter realitätsnahen Bedingungen funktioniert, wurde experimentell untersucht. Die dafür erhobenen Daten und die angewendete Methodik stellen einen weiteren Beitrag dieser Arbeit dar.

Die Aufzeichnung des Datensatzes fand in einer deutlich realistischeren Navigationssituation als in der Literatur üblich statt. Anstatt die Erhebung mit einigen wenigen ortskundigen Personen vorzunehmen (siehe z. B. Potorti et al., 2020), wurde die Testroute von einer Vielzahl von BenutzerInnen einer Navigations-App absolviert. Die Orientierung erfolgte nicht mithilfe von künstlich angebrachten Markierungen, sondern anhand der tatsächlich vom Navigationssystem verwendeten Landmarken. Letzteres ermöglicht es, die in dieser Arbeit entwickelte Evaluationsmethodik mit direktem Bezug zum realen Navigationstask zu verwenden. Für jeden Routenabschnitt ist eine Ground-Truth-Position verfügbar – bei Bedarf kann diese nachträglich anhand der Videoaufnahmen verfeinert werden. Die von mehreren Sensoren und mit hoher Samplingrate erfassten Messwerte sind auch für die Erforschung und Entwicklung anderer Positionierungsverfahren geeignet.

Naturalistischer Navigations-Log-Datensatz

Der mithilfe der URwalking-App erhobene Datensatz ermöglicht Einblicke in reales Nutzerverhalten während der Fußgängernavigation. Der gesamte Datensatz umfasst einen mehrjährigen Zeitraum und enthält mehr als 11 000 Sitzungen. Ein Teil davon – 600 Sitzungen von 351 Personen – besteht nach den in dieser Arbeit angewendeten Kriterien aus tatsächlich vor Ort absolvierten Navigationssitzungen. Ein ähnlicher Datensatz ist mangels vergleichbarer, im Realbetrieb nutzbarer Systeme, nicht aus der Literatur bekannt. Der Datensatz wurde bisher hauptsächlich im Hinblick auf die für die Positionierung relevanten Aspekte hin ausgewertet und ermöglicht weitere Studien zum Navigationsverhalten.

HAR-Datensatz für navigationsspezifische Aktivitäten

Neben dem Navigations-Log-Datensatz wurde im Rahmen dieser Arbeit ein HAR-Datensatz erhoben. Dieser bietet im Vergleich zu anderen, frei verfügbaren Datensätzen ebenfalls einige Alleinstellungsmerkmale: Während in der Literatur häufig nur grobe, allgemeine Aktivitäten zum Einsatz kommen, liegt der Fokus hier auf Bewegungsmustern, die für die Fußgänger-Indoornavigation relevant sind. Für alle diese Aktivitäten wurden in vier verschiedenen Gerätepositionen Messwerte

von fünf Sensoren aufgezeichnet – die meisten HAR-Datensätze beschränken sich hingegen auf Accelerometer und Gyroskop.

Die Evaluation von HAR-Modellen in dieser Arbeit zeigt, dass Verbesserungspotenzial hinsichtlich der Klassifikationsgenauigkeit zwischen den fein differenzierten Klassen besteht. Der erhobene Datensatz eignet sich somit zur zukünftigen Evaluation von anderen HAR-Verfahren, die für den Einsatz in Indoor-Navigationssystemen konzipiert sind.

7.3 Limitierungen

Einige der von Bauer (2018, S. 193 ff.) genannten Einschränkungen – insbesondere hinsichtlich des verwendeten Experimentaufbaus – gelten auch hier. Spezifisch für diese Arbeit sind weiterhin einige zusätzliche Limitierungen zu nennen.

Realitätsnähe der Methodik

Trotz aller Bemühungen um einen hohen Grad an Realismus während der Positionierungsexperimente unterscheiden sich diese nach wie vor von einer natürlichen Navigationssituation, was an mehreren Aspekten erkennbar ist: Das Navigationsziel und die Route sind zu Beginn des Experiments bereits vorgegeben und werden nicht von den Testpersonen selbst aufgrund eines Informationsbedürfnisses und individueller Präferenzen gewählt. Während des Experiments ist ein Testleiter anwesend, welcher im Falle von Problemen eingreifen kann und ansprechbar ist. Dies steht im Kontrast zu einer lediglich von der App unterstützten Navigation. Durch die Verwendung von dedizierten Testgeräten anstelle des eigenen Smartphones sind Unterbrechungen durch eingehende Nachrichten o. Ä. ausgeschlossen, die Navigations-App erhält immer den vollen Fokus.

Diese Einschränkungen sind jedoch nötig, um unter möglichst ähnlichen Bedingungen auf ein- und derselben Route vergleichbare Daten zu erheben, welche später als Grundlage für Simulationsstudien dienen können. Ein Abgleich mit den Ergebnissen der naturalistischen Studie zeigt, dass beispielsweise die Art und Weise, wie das Gerät gehalten und getragen wird, außerhalb der Experimentsituation deutlich variabler ist. Dies muss beim Schließen auf die Performance des Positionierungssystems im Realbetrieb berücksichtigt werden.

Auswahl der Stichprobe

Die Testpersonen in der abschließenden Positionierungsstudie setzten sich größtenteils aus Studierenden zusammen. Ähnliches kann aufgrund der Nutzungsstatistik für die Benutzerschaft der URwalking-App im Allgemeinen angenommen

werden, auch wenn dort keine demografischen Daten erfasst wurden. Die Personengruppe ist demnach hinsichtlich ihrer Ausbildung und ihres Alters recht homogen, bildet jedoch gleichzeitig eine der Zielgruppen der Campus-Navigation ab.

Da die Schritterkennung ein Kernbestandteil des Indoor-Positionierungssystems und Voraussetzung für die Positionsverfolgung ist, konnten keine Testpersonen mit körperlichen Einschränkungen an den Experimenten teilnehmen.

Verallgemeinerbarkeit der Testroute

Die Testroute der abschließenden Positionierungsstudie enthält alle für die Evaluation des mit HAR erweiterten Positionierungssystems nötigen Eigenschaften, kann jedoch selbstverständlich nicht das komplette Spektrum an möglichen Navigationssituationen – selbst beschränkt auf den Kontext der Fußgängernavigation auf dem Campus – abdecken. Wie die Auswertung der naturalistischen Logdaten zeigt, sind beispielsweise Wechsel zwischen Indoor- und Outdoorbereichen ein häufiger Bestandteil der abgefragten Routen.

Bei der Durchführung der Experimente wurde darauf geachtet, keine routenspezifischen Annahmen zu treffen. Um das System in anderen Umgebungen bzw. auf anderen Routen zu testen, sind mehr annotierte Daten nötig, deren Erhebung zeitaufwändige Experimente erfordert.

Vergleichbarkeit mit anderen Systemen

Schließlich bleibt die Frage, inwiefern die Ergebnisse dieser Arbeit mit anderen Ansätzen zur Indoor-Positionierung vergleichbar sind. Trotz Bemühungen darum existiert nach wie vor kein universell verwendeter Standard zur Evaluation von Indoor-Positionierungssystemen, weswegen eine Vergleichbarkeit zwischen unterschiedlichen Systemen am besten durch die regelmäßig stattfindenden Wettbewerbe, etwa im Rahmen der IPIN-Konferenz, gegeben ist. Aufgrund der ständig weiterentwickelten Evaluationsmethodik und den wechselnden Testumgebungen ist jedoch auch hier ein langfristiger Vergleich schwierig.

In der vorliegenden Arbeit liegt immer eine tatsächliche Navigationssituation zugrunde, während der das Positionierungssystem zur Gewährleistung der Navigationsassistenz den korrekten Routenabschnitt bestimmen muss. In den genannten Wettbewerben hingegen liegt das Augenmerk auf einem möglichst geringen metrischen Positionsfehler zu jedem Zeitpunkt; der absolvierte Pfad führt oft scheinbar willkürlich durch das Gebäude, anstatt eine realistische Route zu einem Zielort zu verfolgen. Aus diesen Gründen und aufgrund der teils abweichenden Sensorik – es werden keine Daten von u. a. Schritterkennung und GRV bereitgestellt

– wurde im Rahmen dieses Projekts auf eine Teilnahme an den im Rahmen der IPIN-Konferenz veranstalteten Indoor-Positionierungswettbewerben verzichtet.

7.4 Ausblick

Wie zu Beginn dieses Promotionsvorhabens vor einigen Jahren gilt auch heute noch, dass nach wie vor keine verbreitete Lösung für die Indoor-Positionierung existiert. Auf technischer Ebene sind jedoch einige Fortschritte zu verzeichnen, die Auswirkungen auf die Indoor-Positionierung haben. Die weiter steigende Rechenleistung von Smartphones erlaubt es, immer komplexere Berechnungen durchzuführen. ML-Inferenz muss häufig nicht mehr auf einen Server ausgelagert werden, sondern kann vor Ort auf dem Gerät – teils durch spezielle Koprozessoren beschleunigt – stattfinden. Im Hinblick auf die Sensorik hat zuletzt die UWB-Technik Einzug in Consumer-Geräte gefunden und sich im Bereich der infrastrukturabhängigen Positionierung als Alternative bzw. Ergänzung zu WLAN und Bluetooth etabliert. Weiterer Forschungsbedarf in diesem Feld ist also auf jeden Fall gegeben.

In der Campus-Navigation URwalking hat sich das in dieser Arbeit entwickelte Verfahren als effektiv erwiesen, zu einer »Serienreife« ist jedoch noch ein weiter Weg. Um das System unter verschiedenen Bedingungen zu verifizieren, sind weitere Positionierungsexperimente und Simulationsstudien vonnöten. Diese auf den gesamten Campus zu skalieren, ist aufgrund der zeitintensiven Datenerhebung durch menschliche Testpersonen nicht praktikabel. Stattdessen ist eine Generierung synthetischer Datensätze denkbar, welche auf gesammelten realen Nutzungsdaten und daraus abgeleiteten Modellen basieren.

Um die in dieser Arbeit trainierten, statischen HAR-Modelle besser an die jeweiligen NutzerInnen der Navigations-App anzupassen, ist ein Vorgehen denkbar, bei dem die Modelle im Laufe der Zeit verfeinert werden – möglicherweise sogar direkt auf dem mobilen Gerät mittels *Transfer Learning*. Die dafür nötigen persönlichen Aktivitätsdaten können durch in den Navigationsvorgang eingebundene Rückfragen (z. B. »Bist du gerade durch eine Tür gegangen?«) erhoben werden.

Generell bietet der Kontext einer Navigations-App bzw. allgemein das Vorhandensein einer Benutzeroberfläche den Vorteil, dass das System mit dem Benutzer in Dialog treten und etwaige Unklarheit über die geschätzte Position ausräumen kann. Bei einer landmarkengestützten Fußgängernavigation ist dies insbesondere von Relevanz, da dort immer auf konkrete Objekte Bezug genommen werden kann. Je nach Ausgestaltung und Komplexität dieses Dialogs sind dafür Verfahren des *Natural Language Understandings* vonnöten, wodurch sich ein weiterer For-

schungszweig anschließt.

Die Analysen der naturalistischen Navigations-Logdaten der URwalking-App beschränkten sich in dieser Arbeit auf die für die Positionierung und die Erkennung von Benutzeraktivitäten relevanten Aspekte. Die URwalking-App befindet sich nach wie vor im Betrieb und steht für weitere Studien zum Navigationsverhalten zur Verfügung. Nachdem die Benutzung aufgrund der Covid-19-Pandemie und der damit verbundenen reduzierten Menge an Präsenzveranstaltungen zuletzt abgenommen hatte, war zu Beginn des Wintersemesters 2021/22 wieder ein Zuwachs der Nutzerschaft zu verzeichnen.

Abbildungsverzeichnis

2.1	Luftbild des Regensburger Campus in Richtung Norden	12
2.2	Visualisierung des graphbasierten Umgebungsmodells. Kartenmaterial von OpenStreetMap (Haklay & Weber, 2008)	14
2.3	SQLite-Schema des Navigationsgraphs	17
2.4	Von der Sensor-API verwendetes Koordinatensystem	18
2.5	Beispieldaten des Beschleunigungssensors	20
2.6	Rohe Schritt-Trajektorien des Rotationsvektors (blau) und des Game-Rotationsvektors (grün) sowie die tatsächlich gegangene Route	22
2.7	Luftdruckveränderung beim Wechseln von Stockwerken	23
2.8	Screenshots der URwalking-App (Suche und Navigation)	26
2.9	Screenshots der URwalking-App (Kartenmodus)	27
3.1	Durchschreiten des Navigationsgraphs (rot: Schrittvektoren; blau: geschätzte Position auf der wahrscheinlichsten Kante)	45
3.2	Initialisierung des Partikelfilters	48
3.3	Low-Variance-Resampling-Verfahren	50
3.4	Beispiel für das Schrittlängen-Korrekturverfahren beim Abbiegen	54
3.5	Das Testareal des ersten Experiments mit 14 durch Landmarken und architektonische Einheiten definierten Abschnitten	58
3.6	Trefferquote des Basis-Positionierungssystems	60
3.7	Streuung der Trefferquote in den 15 Testläufen	60
3.8	Zusätzlich eingefügte Kanten	61
3.9	Experimentaufbau mit zwei per Bluetooth verbundenen Geräten	63
3.10	Die drei Testrouten im PT-Gebäude für Experiment 2	65
3.11	Auswirkung der Partikelanzahl auf die Positionierungsgenauigkeit	66
3.12	Beispiele für die drei Modellierungs-Varianten im PT-Foyer	68
3.13	Auswirkung der Modellierung von Freiflächen auf die Positionierungsgenauigkeit	69
3.14	Auswirkung der Schrittlängenadaption auf die Positionierungsgenauigkeit	70

3.15	Verlauf von Partikelverteilung und geschätzter Position für einen ausgewählten Simulationsdurchlauf	71
3.16	Auswirkung der Richtungskorrektur auf die Positionierungsgenauigkeit	72
4.1	Wöchentliche Sitzungen im Verlauf der Studie	87
4.2	Verteilung der Sitzungen im Tages- und Wochenverlauf	87
4.3	Verteilung der Start- und Zielpunkte der Routen auf dem Campus	88
4.4	Anzahl unterschiedlicher Bereiche/Gebäude pro Route	89
4.5	Indoor-/Outdoor-Wechsel und Stockwerkswechsel pro Route	90
4.6	Verteilung der Sitzungslängen	91
4.7	Einfluss von Wochentag und Tageszeit auf die Sitzungsdauer	92
4.8	Verteilung der Routenlängen	92
4.9	Vom Schritterkenner registrierte Schritte pro Sitzung	93
4.10	Nutzungsstatistiken pro Benutzer	95
4.11	Kumulative Verteilung der pro Benutzer getätigten Schritte (nicht abgebildet: 62 BenutzerInnen mit mehr als 3000 Schritten)	96
4.12	Verfügbarkeit ausgewählter Sensoren auf den Geräten im Datensatz	97
4.13	Vergleich der GPS-Genauigkeit innerhalb und außerhalb von Gebäuden	98
4.14	Berichtete Genauigkeit von ausgewählten Sensoren	98
4.15	Darstellung des Clustering-Ergebnisses mithilfe des t-SNE-Verfahrens	101
4.16	Heatmap des Verlaufs aller Routen im gefilterten Datensatz	102
4.17	Schrittfrequenz-Verteilung	103
4.18	Beispiele für durch Gehpausen unterbrochene Navigationssitzungen	104
4.19	Vorder- und Hintergrundstatus der Navigations-App in beispielhaft ausgewählten Sitzungen	105
4.20	Verlauf des Vorder-/Hintergrundstatus im Zeitverlauf der Navigationssitzungen (auf die Sitzungslänge normiert)	106
4.21	Verteilung der Ausrichtung des Geräts nach oben/unten	107
4.22	Beispielsitzung mit in der Tasche getragenen Gerät	107
4.23	Beispielsitzung mit in der Hand schwingendem Gerät	108
5.1	Beispielkombinationen mit unterschiedlicher Charakteristik aus dem Aktivitäts-Datensatz	125
5.2	Klassifikator-Setup mit neun Klassen und optional vorangestelltem Gerätepositions-Klassifikator	130

5.3	Zweistufige Klassifikator-Hierarchie mit vier groben Aktivitätsklassen und jeweils zwei/drei Unterklassen	130
5.4	Treffergenauigkeit der Geräteposition für verschiedene Fenstergrößen und Modell-Architekturen	131
5.5	Treffergenauigkeit für alle neun Klassen für verschiedene Fenstergrößen und Modell-Architekturen bei unbekannter Geräteposition (K_1)	132
5.6	Konfusionsmatrix des besten 9-Klassen-Klassifikators bei unbekannter Geräteposition (InceptionTime 3 s)	134
5.7	Klassifikations-Genauigkeit bei vier groben Klassen für verschiedene Fenstergrößen und Modell-Architekturen (K_2)	135
5.8	Konfusionsmatrix des besten 4-Klassen-Klassifikators (K_2)	136
5.9	Evaluationsergebnisse (angepasstes F_1 -Maß) der hierarchischen Klassifikation	140
5.10	Verteilung der Geräteposition im Zeitverlauf der Navigationssitzungen	142
5.11	Verteilung der Geräteposition für unterschiedlich lange Navigationssitzungen	143
5.12	Verteilung der Benutzeraktivität im Zeitverlauf der Navigationssitzungen (auf die Sitzungslänge normiert)	144
5.13	Die Testroute im abschließenden Positionierungsexperiment	148
5.14	Trefferquote des Baseline-Systems in vier verschiedenen Konfigurationen	152
5.15	Trefferquote des Baseline-Systems in den einzelnen Routenabschnitten	153
5.16	Erkannte Aktivitäten während des Stockwerkswechsels in Routenabschnitt 12	154
5.17	Trefferquote des Systems mit HAR-Einbindung	155
5.18	Vergleich der Trefferquote in den einzelnen Routenabschnitten	156
5.19	Beispiel für die Positionskorrektur durch Türerkennung	157

Tabellenverzeichnis

5.1	Aktivitäten und dazugehörige Anweisungen	123
5.2	Architektur des LSTM-basierten Modells	126
5.3	Treffergenauigkeit und F_1 -Maß der besten Gerätepositions-Klassifikatoren für verschiedene Fenstergrößen	131
5.4	Treffergenauigkeit und F_1 -Maß der besten 9-Klassen-Klassifikatoren (K_1) für verschiedene Gerätepositionen und Fenstergrößen	133
5.5	Treffergenauigkeit und F_1 -Maß der besten 4-Klassen-Klassifikatoren (K_2) für verschiedene Gerätepositionen und Fenstergrößen	135
5.6	Treffergenauigkeit und F_1 -Maß (in Klammern) der besten nachgeschalteten Klassifikatoren (K_{3-6}) für verschiedene Gerätepositionen und Fenstergrößen	137
5.7	Navigationsanweisungen für die Testroute im abschließenden Experiment	149

Literaturverzeichnis

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2015). *TensorFlow, Large-scale machine learning on heterogeneous systems*. Zugriff auf <https://github.com/tensorflow/tensorflow/tree/master/tensorflow/lite/tools/benchmark>
- Abbas, M., Elhamshary, M., Rizk, H., Torki, M. & Youssef, M. (2019). WiDeep: WiFi-based Accurate and Robust Indoor Localization System using Deep Learning. In *2019 IEEE International Conference on Pervasive Computing and Communications (PerCom)* (S. 1–10). IEEE.
- Abdelnasser, H., Mohamed, R., Elgohary, A., Alzantot, M. F., Wang, H., Sen, S., ... Youssef, M. (2016). SemanticSLAM: Using Environment Landmarks for Unsupervised Indoor Localization. *IEEE Transactions on Mobile Computing*, 15 (7), 1770–1782.
- Aggarwal, C. C. (2018). *Neural Networks and Deep Learning: A Textbook*. Springer International Publishing.
- Aggarwal, J. K. & Xia, L. (2014). Human activity recognition from 3D data: A review. *Pattern Recognition Letters*, 48, 70–80.
- Almaslukh, B., Artoli, A. & Al-Muhtadi, J. (2018, November). A Robust Deep Learning Approach for Position-Independent Smartphone-Based Human Activity Recognition. *Sensors*, 18 (11), 3726.
- Álvarez García, J. A., Barsocchi, P., Chessa, S. & Salvi, D. (2013). Evaluation of localization and activity recognition systems for ambient assisted living: The experience of the 2012 EvAAL competition. *Journal of Ambient Intelligence and Smart Environments*, 5, 119–132.
- Anguita, D., Ghio, A., Oneto, L., Parra, X. & Reyes-Ortiz, J. L. (2013). A Public Domain Dataset for Human Activity Recognition Using Smartphones. In *ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (S. 437–442). ESANN.
- Ashqar, H. I., Almannaa, M. H., Elhenawy, M., Rakha, H. A. & House, L. (2019).

- Smartphone Transportation Mode Recognition Using a Hierarchical Machine Learning Classifier and Pooled Features From Time and Frequency Domains. *IEEE Transactions on Intelligent Transportation Systems*, 20 (1), 244–252.
- Bahl, P. & Padmanabhan, V. N. (2000). RADAR: An In-building RF-based User Location and Tracking System. In *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies* (Bd. 2, S. 775–784). IEEE.
- Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. In J. J. Grefenstette (Hrsg.), *Proceedings of the Second International Conference on Genetic Algorithms* (Bd. 206, S. 14–21). Lawrence Erlbaum Associates.
- Banerjee, D., Agarwal, S. K. & Sharma, P. (2015). Improving floor localization accuracy in 3D spaces using barometer. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers - ISWC '15* (S. 171–178). ACM Press.
- Bauer, C. (2018). *Unterstützung der Orientierung im Innenbereich: Analyse landmarkenbasierter Karten-Interfaces anhand des Blickverhaltens der Nutzer* (Dissertation, Universität Regensburg). Zugriff auf <https://epub.uni-regensburg.de/37666/>
- Bayat, A., Pomplun, M. & Tran, D. A. (2014). A Study on Human Activity Recognition Using Accelerometer Data from Smartphones. *Procedia Computer Science*, 34, 450–457.
- Beauregard, S. & Haas, H. (2006). Pedestrian Dead Reckoning: A Basis for Personal Positioning. In *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication* (S. 27–35).
- Beauregard, S., Widyawan & Klepal, M. (2008). Indoor PDR performance enhancement using minimal map information and particle filters. In *Position, Location and Navigation Symposium, 2008 IEEE/ION* (S. 141–147).
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... Thomas, D. (2001). *Manifesto for Agile Software Development*. Zugriff auf <http://agilemanifesto.org/>
- Bobkov, D., Grimm, F., Steinbach, E., Hilsenbeck, S. & Schroth, G. (2015). Activity recognition on handheld devices for pedestrian indoor navigation. In *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)* (S. 1–10). IEEE.
- Bohannon, R. W. & Williams Andrews, A. (2011). Normal walking speed: a descriptive meta-analysis. *Physiotherapy*, 97 (3), 182–189.
- Borenstein, J. & Ojeda, L. (2010). Heuristic Drift Elimination for Personnel Tracking Systems. *Journal of Navigation*, 63 (4), 591–606.

- Bortz, J. & Schuster, C. (2010). *Statistik für Human- und Sozialwissenschaftler* (7. Aufl.). Springer.
- Bousdar Ahmed, D., Diez, L. E., Diaz, E. M. & Garcia Dominguez, J. J. (2020). A Survey on Test and Evaluation Methodologies of Pedestrian Localization Systems. *IEEE Sensors Journal*, 20 (1), 479–491.
- Brajdic, A. & Harle, R. (2013). Walk Detection and Step Counting on Unconstrained Smartphones. *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '13*.
- Brown, B. & Laurier, E. (2012). The normal natural troubles of driving with GPS. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (S. 1621–1630). ACM.
- Brown, B., McGregor, M. & McMillan, D. (2014). 100 Days of iPhone Use: Understanding the Details of Mobile Device Use. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services – MobileHCI '14* (S. 223–232). ACM Press.
- Brush, A. B., Hammil, K., Levi, S., Karlson, A. K., Scott, J., Sarin, R., ... Sinclair, M. (2010). User experiences with activity-based navigation on mobile devices. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services – MobileHCI '10* (S. 73–82). ACM Press.
- Brändle, N., Bauer, D. & Seer, S. (2006). Track-Based Finding of Stopping Pedestrians – a Practical Approach for Analyzing a Public Infrastructure. In *2006 IEEE Intelligent Transportation Systems Conference* (S. 115–120). IEEE.
- Bulling, A., Blanke, U. & Schiele, B. (2014). A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys*, 46 (3), 1–33.
- Böhmer, M., Hecht, B., Schöning, J., Krüger, A. & Bauer, G. (2011). Falling Asleep with Angry Birds, Facebook and Kindle: A Large Scale Study on Mobile Application Usage. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services* (S. 47–56). ACM.
- Caduff, D. & Timpf, S. (2008). On the assessment of landmark salience for human navigation. *Cognitive processing*, 9 (4), 249–267.
- Carboni, D., Manchinu, A., Marotto, V., Piras, A. & Serra, A. (2015). Infrastructure-free indoor navigation: a case study. *Journal of Location Based Services*, 9 (1), 33–54.
- Carrascal, J. P. & Church, K. (2015). An In-Situ Study of Mobile App & Mobile Search Interactions. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (S. 2739–2748). ACM.

- Chen, Z., Zou, H., Jiang, H., Zhu, Q., Soh, Y. C. & Xie, L. (2015). Fusion of WiFi, Smartphone Sensors and Landmarks using the Kalman Filter for Indoor Localization. *Sensors (Switzerland)*, 15 (1), 715–732.
- Chintalapudi, K., Padmanabha Iyer, A. & Padmanabhan, V. N. (2010). Indoor localization without the pain. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking* (S. 173–184).
- Combettes, C. & Renaudin, V. (2015). Comparison of Misalignment Estimation Techniques between Handheld Device and Walking Directions. In *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE.
- Cotera, P., Velazquez, M., Cruz, D., Medina, L. & Bandala, M. (2016). Indoor Robot Positioning Using an Enhanced Trilateration Algorithm. *International Journal of Advanced Robotic Systems*, 13 (3), 110.
- Davidson, P. & Piche, R. (2016). A Survey of Selected Indoor Positioning Methods for Smartphones. *IEEE Communications Surveys & Tutorials*, 19 (2), 1347–1370.
- De-La-Hoz-Franco, E., Ariza-Colpas, P., Quero, J. M. & Espinilla, M. (2018). Sensor-Based Datasets for Human Activity Recognition – A Systematic Review of Literature. *IEEE Access*, 6, 59192–59210.
- Diez, L. E., Bahillo, A., Otegui, J. & Otim, T. (2018). Step Length Estimation Methods Based on Inertial Sensors: A Review. *IEEE Sensors Journal*, 18 (17), 6908–6926.
- Do, T. M. T., Dousse, O., Miettinen, M. & Gatica-Perez, D. (2015). A probabilistic kernel method for human mobility prediction with smartphones. *Pervasive and Mobile Computing*, 20, 13–28.
- Do, T. M. T. & Gatica-Perez, D. (2013a). Human interaction discovery in smartphone proximity networks. *Personal and Ubiquitous Computing*, 17 (3), 413–431.
- Do, T. M. T. & Gatica-Perez, D. (2013b). The places of our lives: Visiting patterns and automatic labeling from longitudinal smartphone data. *IEEE Transactions on Mobile Computing*, 13 (3), 638–648.
- Do, T. M. T. & Gatica-Perez, D. (2014). Where and what: Using smartphones to predict next locations and applications in daily life. *Pervasive and Mobile Computing*, 12, 79–91.
- Doucet, A., Freitas, N. d. & Gordon, N. (Hrsg.). (2001). *Sequential Monte Carlo Methods in Practice*. Springer New York.
- Dumais, S., Jeffries, R., Russell, D. M., Tang, D. & Teevan, J. (2014). Understanding User Behavior Through Log Data and Analysis. In J. S. Olson & W. A. Kellogg (Hrsg.), *Ways of Knowing in HCI*. Springer New York.

- Dunnett, C. W. (1980). Pairwise multiple comparisons in the unequal variance case. *Journal of the American Statistical Association*, 75 (372), 796–800.
- Ebner, F., Deinzer, F. & Lukas, K. (2014). Robust Self-Localization using Wi-Fi, Step/Turn-Detection and Recursive Density Estimation. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)* (S. 627–635).
- Ebner, F., Fetzer, T., Deinzer, F., Köping, L. & Grzegorzec, M. (2015). Multi sensor 3D indoor localisation. In *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)* (S. 1–11).
- Ebner, M., Fetzer, T., Bullmann, M., Deinzer, F. & Grzegorzec, M. (2020). Recognition of Typical Locomotion Activities Based on the Sensor Data of a Smartphone in Pocket or Hand. *Sensors*, 20, 6559.
- Edel, M. & Köppe, E. (2016). Binarized-BLSTM-RNN based Human Activity Recognition. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)* (S. 1–7).
- Elhoushi, M., Georgy, J., Noureldin, A. & Korenberg, M. J. (2016). Motion Mode Recognition for Indoor Pedestrian Navigation Using Portable Devices. *IEEE Transactions on Instrumentation and Measurement*, 65 (1), 208–221.
- Fan, L., Wang, Z. & Wang, H. (2013). Human Activity Recognition Model Based on Decision Tree. In *2013 International Conference on Advanced Cloud and Big Data* (S. 64–68). IEEE.
- Fang, Z., Li, Q. & Shaw, S.-L. (2015). What about people in pedestrian navigation? *Geo-spatial Information Science*, 18 (4), 135–150.
- Faragher, R. & Harle, R. (2013). SmartSLAM – An Efficient Smartphone Indoor Positioning System Exploiting Machine Learning and Opportunistic Sensing. In *Proceedings of the 26th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2013)* (S. 1006–1019).
- Ferris, B., Haehnel, D. & Fox, D. (2006). Gaussian Processes for Signal Strength-Based Location Estimation. In *Robotics: Science and Systems II*. Robotics: Science and Systems Foundation.
- Fetzer, T., Ebner, F., Bullmann, M., Deinzer, F. & Grzegorzec, M. (2018). Smartphone-Based Indoor Localization within a 13th Century Historic Building. *Sensors*, 18 (12), 4095.
- Fujinami, K. & Kouchi, S. (2013). Recognizing a Mobile Phone’s Storing Position as a Context of a Device and a User. In K. Zheng, M. Li & H. Jiang (Hrsg.), *Mobile and Ubiquitous Systems: Computing, Networking, and Services* (Bd. 120, S. 76–88). Springer Berlin Heidelberg.
- Garcia, E., Poudereux, P., Hernandez, A., Urena, J. & Gualda, D. (2015). A robust UWB indoor positioning system for highly complex environments. In *2015 IEEE International Conference on Industrial Technology (ICIT)* (S.

- 3386–3391). IEEE.
- Garrett, J. J. (2011). *The Elements of User Experience: User-Centered Design for the Web and Beyond* (2. Aufl.). New Riders.
- Gigl, T., Janssen, G. J., Dizdarevic, V., Witrisal, K. & Irahhten, Z. (2007). Analysis of a UWB Indoor Positioning System Based on Received Signal Strength. In *2007 4th Workshop on Positioning, Navigation and Communication* (S. 97–101). IEEE.
- Giorgi, G., Frigo, G. & Narduzzi, C. (2017). Dead Reckoning in Structured Environments for Human Indoor Navigation. *IEEE Sensors Journal*, 17 (23), 7794–7802.
- Godha, S. & Lachapelle, G. (2008). Foot mounted inertial system for pedestrian navigation. *Measurement Science and Technology*, 19 (7), 075202.
- Golledge, R. G. (1991). Cognition of physical and built environments. In *Environment, Cognition, and Action: An Integrated Approach*. Oxford University Press.
- Google. (2021a). *Pixel – technische Daten zur Hardware*. Zugriff auf <https://support.google.com/pixelphone/answer/7158570>
- Google. (2021b). *Unterstützte Geräte*. Zugriff auf <https://support.google.com/googleplay/answer/1727131?hl=de>
- Google Developers. (2014). *Google I/O 2014 - A 3D tablet, an OSCAR, and a little cash. Tango, Spotlight, Ara. ATAP*. Zugriff auf <https://www.youtube.com/watch?v=cV8JDSO1NS8>
- Google Developers. (2021a). *Fused Location Provider API*. Zugriff auf <https://developers.google.com/location-context/fused-location-provider>
- Google Developers. (2021b). *Location*. Zugriff auf [https://developer.android.com/reference/android/location/Location.html#getAccuracy\(\)](https://developer.android.com/reference/android/location/Location.html#getAccuracy())
- Google Developers. (2021c). *SensorManager*. Zugriff auf <https://developer.android.com/reference/android/hardware/SensorManager>
- Google Developers. (2021d). *Sensors Overview*. Zugriff auf https://developer.android.com/guide/topics/sensors/sensors_overview
- Google Developers. (2021e). *Wi-Fi scanning overview*. Zugriff auf <https://developer.android.com/guide/topics/connectivity/wifi-scan>
- Google LLC & Open Handset Alliance. (2020). *Sensor types*. Zugriff auf <https://source.android.com/devices/sensors/sensor-types>
- Gordon, N. J., Salmond, D. J. & Smith, A. F. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140 (2), 107–113.
- Gower, J. C. (1971). A General Coefficient of Similarity and Some of Its Properties. *Biometrics*, 27 (4), 857.

- Gu, F., Hu, X., Ramezani, M., Acharya, D., Khoshelham, K., Valaee, S. & Shang, J. (2019). Indoor Localization Improved by Spatial Context — A Survey. *ACM Computing Surveys*, 52 (3), 1–35.
- Gu, F., Kealy, A., Khoshelham, K. & Shang, J. (2015). User-Independent Motion State Recognition Using Smartphone Sensors. *Sensors*, 15 (12), 30636–30652.
- Gu, F., Valaee, S., Khoshelham, K., Shang, J. & Zhang, R. (2020). Landmark Graph-Based Indoor Localization. *IEEE Internet of Things Journal*, 7 (9), 8343–8355.
- Guan, W., Chen, S., Wen, S., Tan, Z., Song, H. & Hou, W. (2020). High-Accuracy Robot Indoor Localization Scheme Based on Robot Operating System Using Visible Light Positioning. *IEEE Photonics Journal*, 12 (2), 1–16.
- Guan, Y. & Plötz, T. (2017). Ensembles of Deep LSTM Learners for Activity Recognition using Wearables. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1 (2), 1–28.
- Guo, H., Chen, L., Chen, G. & Lv, M. (2016). Smartphone-based activity recognition independent of device orientation and placement. *International Journal of Communication Systems*, 29 (16), 2403–2415.
- Haklay, M. & Weber, P. (2008). OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Computing*, 7 (4).
- Harle, R. (2013). A Survey of Indoor Inertial Positioning Systems for Pedestrians. *IEEE Communications Surveys & Tutorials*, 15 (3), 1281–1293.
- Hassan, M. M., Uddin, M. Z., Mohamed, A. & Almogren, A. (2018). A robust human activity recognition system using smartphone sensors and deep learning. *Future Generation Computer Systems*, 81, 307–313.
- Hastie, T., Tibshirani, R. & Friedman, J. H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2. Aufl.). Springer.
- He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (S. 770–778).
- He, Z. & Jin, L. (2009). Activity recognition from acceleration data based on discrete cosine transform and SVM. In *2009 IEEE International Conference on Systems, Man and Cybernetics* (S. 5041–5044).
- Hendrycks, D. & Gimpel, K. (2018). A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. *arXiv:1610.02136v3 [cs.NE]*.
- Herrera, J. C. A., Plöger, P. G., Hinkenjann, A., Maiero, J., Flores, M. & Ramos, A. (2014). Pedestrian indoor positioning using smartphone multi-sensing, radio beacons, user positions probability map and Indoor-OSM floor plan

- representation. In *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)* (S. 636–645).
- Hilsenbeck, S., Bobkov, D., Schroth, G., Huitl, R. & Steinbach, E. (2014). Graph-based Data Fusion of Pedometer and WiFi Measurements for Mobile Indoor Positioning. *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*.
- Hoffmann, E. J., Schauer, L., Schonfeld, M. & Kraus, M. (2017). Robust pedestrian dead reckoning using anchor point recalibration. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)* (S. 1–7). IEEE.
- Hsu, H.-H., Chang, J.-K., Peng, W.-J., Shih, T. K., Pai, T.-W. & Man, K. L. (2018). Indoor localization and navigation using smartphone sensory data. *Annals of Operations Research*, 265 (2), 187–204.
- Huang, H., Gartner, G., Krisp, J. M., Raubal, M. & Van de Weghe, N. (2018). Location based services: ongoing evolution and research agenda. *Journal of Location Based Services*, 12 (2), 63–93.
- Ingram, S., Harmer, D. & Quinlan, M. (2004). Ultrawideband indoor positioning systems and their use in emergencies. In *PLANS 2004. Position Location and Navigation Symposium* (S. 706–715).
- International Organization for Standardization. (2016). *Information technology — Real time locating systems — Test and evaluation of localization and tracking systems*. (ISO/IEC 18305:2016)
- Isard, M. & Blake, A. (1998). CONDENSATION — Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, 29, 5–28.
- Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D. F., Weber, J., ... Petitjean, F. (2020). InceptionTime: Finding AlexNet for time series classification. *Data Mining and Knowledge Discovery*, 34 (6), 1936–1962.
- Jackermeier, R. (2016). *Interaktive Indoor-Lokalisierung für Fußgänger* (Masterarbeit). Universität Regensburg.
- Jackermeier, R. & Ludwig, B. (2018a). Exploring the Limits of PDR-based Indoor Localisation Systems under Realistic Conditions. *Journal of Location Based Services*, 12 (3-4), 231–272.
- Jackermeier, R. & Ludwig, B. (2018b). Task-Oriented Evaluation of Indoor Positioning Systems. In P. Kiefer, H. Huang, N. Van de Weghe & M. Raubal (Hrsg.), *Progress in Location Based Services 2018* (S. 25–47). Springer International Publishing.
- Jackermeier, R. & Ludwig, B. (2019a). Door Transition Detection for Long-Term Stability in Pedestrian Indoor Positioning. In *2019 International Conference*

- on *Indoor Positioning and Indoor Navigation (IPIN)* (S. 1–8). IEEE.
- Jackermeier, R. & Ludwig, B. (2019b). User Behaviour in the Wild: Analysing Naturalistic Log Data of a Pedestrian Navigation App. *Advances in Cartography and GIScience of the ICA*, 2, 1–8.
- Jackermeier, R. & Ludwig, B. (2021). Smartphone-Based Activity Recognition in a Pedestrian Navigation Context. *Sensors*, 21 (9), 3243.
- Jansen, B. J., Taksa, I. & Spink, A. (2009). Research and Methodological Foundations of Transaction Log Analysis. In B. J. Jansen, I. Taksa & A. Spink (Hrsg.), *Handbook of Research on Web Log Analysis*. IGI Global.
- Jimenez, A. R., Seco, F., Zampella, F., Prieto, J. C. & Guevara, J. (2011). Improved Heuristic Drift Elimination (iHDE) for pedestrian navigation in complex buildings. In *2011 International Conference on Indoor Positioning and Indoor Navigation* (S. 1–8). IEEE.
- Jimenez Ruiz, A. R., Seco Granja, F., Prieto Honorato, J. C. & Guevara Rosas, J. I. (2012). Accurate Pedestrian Indoor Navigation by Tightly Coupling Foot-Mounted IMU and RFID Measurements. *IEEE Transactions on Instrumentation and Measurement*, 61 (1), 178–189.
- Ju, H., Park, S. Y. & Park, C. G. (2018). A Smartphone-Based Pedestrian Dead Reckoning System With Multiple Virtual Tracking for Indoor Navigation. *IEEE Sensors Journal*, 18 (16), 6756–6764.
- Kasebzadeh, P., Hendebay, G., Fritsche, C., Gunnarsson, F. & Gustafsson, F. (2017). IMU Dataset for Motion and Device Mode Classification. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)* (S. 1–8). IEEE.
- Kattenbeck, M. (2016). *Empirically Measuring Saliency of Objects for Use in Pedestrian Navigation* (Dissertation, Universität Regensburg). Zugriff auf <https://epub.uni-regensburg.de/34145/>
- Kattenbeck, M., Müller, M., Ohm, C. & Ludwig, B. (2015). Der Weg ist das Ziel – Fußgängernavigation ist Forschung zu Information Behavior. *Information – Wissenschaft & Praxis*, 66 (1).
- Kaufman, L. & Rousseeuw, P. J. (2005). *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons.
- Ke, S.-R., Thuc, H. L. U., Lee, Y.-J., Hwang, J.-N., Yoo, J.-H. & Choi, K.-H. (2013). A Review on Video-Based Human Activity Recognition. *Computers*, 2 (2), 88–131.
- Kim, S.-E., Kim, Y., Yoon, J. & Kim, E. S. (2012). Indoor positioning system using geomagnetic anomalies for smartphones. In *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)* (S. 1–5). IEEE.
- Kiritchenko, S., Matwin, S. & Famili, A. F. (2005). Functional Annotation

- of Genes Using Hierarchical Text Categorization. *Proceedings of the ACL Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics*, 7.
- Kiukkonen, N., Blom, J., Dousse, O., Gatica-Perez, D. & Laurila, J. (2010). Towards rich mobile phone datasets: Lausanne data collection campaign. In *Proceedings of the ACM International Conference on Pervasive Services*. ACM.
- Klein, I., Solaz, Y. & Ohayon, G. (2018). Pedestrian Dead Reckoning With Smartphone Mode Recognition. *IEEE Sensors Journal*, 18 (18), 7577–7584.
- Klepeis, N. E., Nelson, W. C., Ott, W. R., Robinson, J. P., Tsang, A. M., Switzer, P., ... Engelmann, W. H. (2001). The National Human Activity Pattern Survey (NHAPS): a resource for assessing exposure to environmental pollutants. *Journal of Exposure Science & Environmental Epidemiology*, 11 (3), 231–252.
- Kong, A., Liu, J. S. & Wong, W. H. (1994). Sequential Imputations and Bayesian Missing Data Problems. *Journal of the American Statistical Association*, 89 (425), 278–288.
- Kruskal, W. H. & Wallis, W. A. (1952). Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47 (260), 583–621.
- Kuhn, M. & Johnson, K. (2019). *Feature Engineering and Selection: A Practical Approach for Predictive Models*. Chapman and Hall/CRC.
- Kuo, Y.-S., Pannuto, P., Hsiao, K.-J. & Dutta, P. (2014). Luxapose: indoor positioning with mobile phones and visible light. In *Proceedings of the 20th annual international conference on Mobile computing and networking* (S. 447–458). ACM.
- Kwapisz, J. R., Weiss, G. M. & Moore, S. A. (2011). Activity recognition using cell phone accelerometers. *ACM SIGKDD Explorations Newsletter*, 12 (2), 74–82.
- Lara, O. D. & Labrador, M. A. (2012). A Survey on Human Activity Recognition using Wearable Sensors. *IEEE Communications Surveys & Tutorials*, 15 (3), 1192–1209.
- LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. *nature*, 521 (7553), 436–444.
- Lee, S.-M., Yoon, S. M. & Cho, H. (2017). Human activity recognition from accelerometer data using Convolutional Neural Network. In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)* (S. 131–134).
- Le Scornec, J., Ortiz, M. & Renaudin, V. (2017). Foot-mounted pedestrian navigation reference with tightly coupled GNSS carrier phases, inertial and magnetic data. In *2017 International Conference on Indoor Positioning and*

- Indoor Navigation (IPIN)* (S. 1–8). IEEE.
- Lin Liao, Fox, D., Hightower, J., Kautz, H. & Schulz, D. (2003). Voronoi tracking: location estimation using sparse and noisy sensor data. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)* (Bd. 1, S. 723–728). IEEE.
- Liu, J. S. & Chen, R. (1998). Sequential Monte Carlo Methods for Dynamic Systems. *Journal of the American Statistical Association*, 93 (443), 1032–1044.
- Luo, J., Fan, L. & Li, H. (2017). Indoor Positioning Systems Based on Visible Light Communication: State of the Art. *IEEE Communications Surveys & Tutorials*, 19 (4), 2871–2893.
- Lymberopoulos, D., Liu, J., Yang, X., Choudhury, R. R., Handziski, V. & Sen, S. (2015). A Realistic Evaluation and Comparison of Indoor Location Technologies: Experiences and Lessons Learned. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks* (S. 178–189).
- Lynch, K. (1960). *The image of the city* (Bd. 11). MIT press.
- Ma, Y., Zhou, G. & Wang, S. (2019). WiFi Sensing with Channel State Information: A Survey. *ACM Computing Surveys*, 52 (3), 1–36.
- Malmi, E., Do, T. M. T. & Gatica-Perez, D. (2012). Checking in or checked in: comparing large-scale manual and automatic location disclosure patterns. In *Proceedings of the 11th international conference on mobile and ubiquitous multimedia* (S. 1–10).
- Mendoza-Silva, G. M., Torres-Sospedra, J. & Huerta, J. (2017). A more realistic error distance calculation for indoor positioning systems accuracy evaluation. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)* (S. 1–8). IEEE.
- Mendoza-Silva, G. M., Torres-Sospedra, J., Potorti, F., Moreira, A., Knauth, S., Berkvens, R. & Huerta, J. (2020). Beyond Euclidean Distance for Error Measurement in Pedestrian Indoor Location. *IEEE Transactions on Instrumentation and Measurement*, 1–11.
- Menke, J. & Zakhor, A. (2015). Multi-Modal Indoor Positioning of Mobile Devices. (October), 13–16.
- Montello, D. R. (2005). Navigation. In P. Shah & A. Miyake (Hrsg.), *The Cambridge Handbook of Visuospatial Thinking* (1. Aufl., S. 257–294). Cambridge University Press.
- Montoliu, R., Blom, J. & Gatica-Perez, D. (2013). Discovering places of interest in everyday life from smartphone data. *Multimedia Tools and Applications*, 62 (1), 179–207.
- Moreira, D., Barandas, M., Rocha, T., Alves, P., Santos, R., Leonardo, R., ... Gamboa, H. (2021). Human Activity Recognition for Indoor Localization Using

- Smartphone Inertial Sensors. *Sensors*, 21 (18), 6316.
- Munoz Diaz, E. (2015). Inertial Pocket Navigation System: Unaided 3D Positioning. *Sensors*, 15 (4), 9156–9178.
- Murad, A. & Pyun, J.-Y. (2017). Deep Recurrent Neural Networks for Human Activity Recognition. *Sensors*, 17 (11), 2556.
- Murata, Y., Hiroi, K., Kaji, K. & Kawaguchi, N. (2014). Pedestrian dead reckoning based on human activity sensing knowledge. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct Publication - UbiComp '14 Adjunct* (S. 797–806). ACM Press.
- Norman, D. A. (1986). *User centered system design: New perspectives on human-computer interaction*. CRC Press.
- Nurminen, H., Koivisto, M., Ali-Loytty, S. & Piche, R. (2014). Motion model for positioning with graph-based indoor map. In *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)* (S. 646–655). IEEE.
- Nurminen, H., Ristimäki, A., Ali-Loytty, S. & Piche, R. (2013). Particle filter and smoother for indoor localization. In *International Conference on Indoor Positioning and Indoor Navigation* (S. 1–10). IEEE.
- Öberg, T., Karsznia, A. & Öberg, K. (1993). Basic gait parameters: Reference data for normal subjects, 10-79 years of age. *Journal of Rehabilitation Research and Development*, 30 (2), 210–23.
- Ordóñez, F. & Roggen, D. (2016). Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors*, 16 (1).
- Panko, R. (2018). *The Popularity of Google Maps: Trends in Navigation Apps in 2018*. Zugriff auf <https://themanifest.com/app-development/trends-navigation-apps>
- Perttula, A., Leppakoski, H., Kirkko-Jaakkola, M., Davidson, P., Collin, J. & Takala, J. (2014). Distributed Indoor Positioning System With Inertial Measurements and Map Matching. *IEEE Transactions on Instrumentation and Measurement*, 63 (11), 2682–2695.
- Pielot, M., Poppinga, B., Heuten, W. & Boll, S. (2012). PocketNavigator: studying tactile navigation systems in-situ. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12* (S. 3131). ACM Press.
- Potorti, F., Barsocchi, P., Girolami, M., Torres-Sospedra, J. & Montoliu, R. (2015). Evaluating Indoor Localization Solutions in Large Environments Through Competitive Benchmarking: The EvAAL-ETRI Competition. In *Indoor Positioning and Indoor Navigation (IPIN), 2015 International Conference on* (S. 1–10).

- Potorti, F., Crivello, A., Barsocchi, P. & Palumbo, F. (2018). Evaluation of Indoor Localisation Systems: Comments on the ISO/IEC 18305 Standard. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)* (S. 1–7). IEEE.
- Potorti, F., Park, S., Crivello, A., Palumbo, F., Girolami, M., Barsocchi, P., ... Tsao, Y. (2020). The IPIN 2019 Indoor Localisation Competition — Description and Results. *IEEE Access*, 8.
- Potorti, F., Torres-Sospedra, J., Quezada-Gaibor, D., Jimenez, A. R., Seco, F., Perez-Navarro, A., ... Oh, H. L. (2021). Off-line Evaluation of Indoor Positioning Systems in Different Scenarios: The Experiences from IPIN 2020 Competition. *IEEE Sensors Journal*.
- Pulkkinen, T. & Verwijnen, J. (2015). Evaluating indoor positioning errors. In *2015 International Conference on Information and Communication Technology Convergence (ICTC)* (S. 167–169). IEEE.
- Radu, V. & Marina, M. K. (2013). HiMLoc: Indoor Smartphone Localization via Activity Aware Pedestrian Dead Reckoning with Selective Crowdsourced WiFi Fingerprinting. In *International Conference on Indoor Positioning and Indoor Navigation* (S. 1–10). IEEE.
- Rai, A., Chintalapudi, K. K., Padmanabhan, V. N. & Sen, R. (2012). Zee: zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th annual international conference on Mobile computing and networking - Mobicom '12* (S. 293). ACM Press.
- Ramasamy Ramamurthy, S. & Roy, N. (2018). Recent trends in machine learning for human activity recognition—a survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8 (4), e1254.
- Rantakokko, J., Rydell, J., Strömbäck, P., Händel, P., Callmer, J., Törnqvist, D., ... Grudén, M. (2011). Accurate and reliable soldier and first responder indoor positioning: multisensor systems and cooperative localization. *IEEE Wireless Communications*, 18 (2), 10–18.
- Ravi, D., Wong, C., Lo, B. & Yang, G.-Z. (2016). A deep learning approach to on-node sensor data analytics for mobile or wearable devices. *IEEE Journal of Biomedical and Health Informatics*, 21 (1), 56–64.
- Rehrl, K., Häusler, E., Leitinger, S. & Bell, D. (2014). Pedestrian navigation with augmented reality, voice and digital map: final results from an in situ field study assessing performance and user experience. *Journal of Location Based Services*, 8 (2).
- Renaudin, V., Ortiz, M., Perul, J., Torres-Sospedra, J., Jimenez, A. R., Perez-Navarro, A., ... Park, Y. (2019). Evaluating Indoor Positioning Systems in a Shopping Mall: The Lessons Learned From the IPIN 2018 Competition.

- IEEE Access*, 7, 148594–148628.
- Reyes-Ortiz, J.-L., Oneto, L., Ghio, A., Samá, A., Anguita, D. & Parra, X. (2014). Human Activity Recognition on Smartphones with Awareness of Basic Activities and Postural Transitions. In S. Wermter et al. (Hrsg.), *Artificial Neural Networks and Machine Learning – ICANN 2014* (Bd. 8681, S. 177–184). Springer International Publishing.
- Richter, D., Winter, S., Richter, K.-F. & Stirling, L. (2013). Granularity of locations referred to by place descriptions. *Computers, Environment and Urban Systems*, 41, 88–99.
- Robinson, S., Jones, M., Eslambolchilar, P., Murray-Smith, R. & Lindborg, M. (2010). »I did it my way«: moving away from the tyranny of turn-by-turn pedestrian navigation. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services - MobileHCI '10* (S. 341–344). ACM Press.
- Roggen, D., Calatroni, A., Rossi, M., Holleczeck, T., Forster, K., Troster, G., ... Millan, J. d. R. (2010). Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh International Conference on Networked Sensing Systems (INSS)* (S. 233–240). IEEE.
- Rossi, M., Seiter, J., Amft, O., Buchmeier, S. & Tröster, G. (2013). RoomSense: an indoor positioning system for smartphones using active sound probing. In *Proceedings of the 4th Augmented Human International Conference* (S. 89–95). ACM Press.
- Roy, N., Wang, H. & Roy Choudhury, R. (2014). I am a smartphone and i can tell my user's walking direction. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services* (S. 329–342).
- Ruggiero, L., Charitha, D., Xiang, S. & Lucia, B. (2018). Investigating Pedestrian Navigation in Indoor Open Space Environments using Big Data. *Applied Mathematical Modelling*, 62, 499–509.
- Saeedi, S. & El-Sheimy, N. (2015). Activity Recognition Using Fusion of Low-Cost Sensors on a Smartphone for Mobile Navigation Application. *Micro-machines*, 6 (8), 1100–1134.
- Savino, G.-L., Sturdee, M., Rundé, S., Lohmeier, C., Hecht, B., Prandi, C., ... Schöning, J. (2020). MapRecorder: Analysing Real World Usage of Mobile Map Applications. *Behaviour & Information Technology*, 40 (7), 646-662.
- Shang, J., Gu, F., Hu, X. & Kealy, A. (2015). APFiLoc: An Infrastructure-Free Indoor Localization Method Fusing Smartphone Inertial Sensors, Landmarks and Map Information. *Sensors*, 15 (10), 27251–27272.
- Shavit, Y. & Klein, I. (2021). Boosting Inertial-Based Human Activity Recognition With Transformers. *IEEE Access*, 9, 53540–53547.

- Shepperd, S. W. (1978). Quaternion from Rotation Matrix. *Journal of Guidance and Control*, 1 (3), 223–224.
- Shin, S. H., Park, C. G., Kim, J. W., Hong, H. S. & Lee, J. M. (2007). Adaptive Step Length Estimation Algorithm Using Low-Cost MEMS Inertial Sensors. In *2007 IEEE Sensors Applications Symposium* (S. 1–5). IEEE.
- Shoab, M., Bosch, S., Incel, O. D., Scholten, H. & Havinga, P. J. (2014). Fusion of smartphone motion sensors for physical activity recognition. *Sensors*, 14 (6), 10146–10176.
- Song, X., Fan, X., Xiang, C., Ye, Q., Liu, L., Wang, Z., ... Fang, G. (2019). A Novel Convolutional Neural Network Based Indoor Localization Framework With WiFi Fingerprinting. *IEEE Access*, 7, 110698–110709.
- Sousa Lima, W., Souto, E., El-Khatib, K., Jalali, R. & Gama, J. (2019). Human Activity Recognition Using Inertial Sensors in a Smartphone: An Overview. *Sensors*, 19 (14), 3213.
- StatCounter. (2021). *Mobile Operating System Market Share Worldwide*. Zugriff auf <https://gs.statcounter.com/os-market-share/mobile/worldwide>
- Subbu, K. P., Gozick, B. & Dantu, R. (2013). LocateMe: Magnetic-fields-based Indoor Localization Using Smartphones. *ACM Transactions on Intelligent Systems and Technology*, 4 (4), 73:1–73:27.
- Sun, Y. & Wang, B. (2017). Indoor Corner Recognition from Crowdsourced Trajectories using Smartphone Sensors. *Expert Systems with Applications*, 82, 266–277.
- Susi, M., Renaudin, V. & Lachapelle, G. (2013). Motion Mode Recognition and Step Detection Algorithms for Mobile Phone Users. *Sensors*, 13 (2), 1539–1562.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (S. 1–9).
- Tenzer, F. (2021a). *Anteil der Smartphone-Nutzer in Deutschland in den Jahren 2012 bis 2020*. Zugriff auf <https://de.statista.com/statistik/daten/studie/585883/umfrage/anteil-der-smartphone-nutzer-in-deutschland/>
- Tenzer, F. (2021b). *Anteil der Smartphone-Nutzer in Deutschland nach Altersgruppe im Jahr 2020*. Zugriff auf <https://de.statista.com/statistik/daten/studie/459963/umfrage/anteil-der-smartphone-nutzer-in-deutschland-nach-altersgruppe/>
- Thrun, S., Burgard, W. & Fox, D. (2006). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press.
- Tian, Q., Salcic, Z., Wang, K. I.-K. & Pan, Y. (2016). A Multi-Mode Dead Recko-

- ning System for Pedestrian Tracking Using Smartphones. *IEEE Sensors Journal*, 16 (7), 2079–2093.
- Tomczak, M. & Tomczak, E. (2014). The need to report effect size estimates revisited. An overview of some recommended measures of effect size. *Trends in Sport Sciences*, 1 (21), 19–25.
- Torok, A., Nagy, A., Kovats, L. & Pach, P. (2014). DREAR - Towards Infrastructure-Free Indoor Localization via Dead-Reckoning Enhanced with Activity Recognition. In *2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies* (S. 106–111). IEEE.
- Torres-Sospedra, J., Jiménez, A., Knauth, S., Moreira, A., Beer, Y., Fetzer, T., ... Castelli, E. (2017). The Smartphone-Based Offline Indoor Location Competition at IPIN 2016: Analysis and Future Work. *Sensors*, 17 (3), 557.
- Torres-Sospedra, J., Jiménez, A., Moreira, A., Lungenstrass, T., Lu, W.-C., Knauth, S., ... Tsao, Y. (2018). Off-Line Evaluation of Mobile-Centric Indoor Positioning Systems: The Experiences from the 2017 IPIN Competition. *Sensors*, 18 (2).
- Ullmann, M. (2020). *Datengetriebene Optimierung präferenzadaptiver Fußwegrou-ten durch Gebäudekomplexe* (Dissertation, Universität Regensburg). Zugriff auf <https://epub.uni-regensburg.de/43697/>
- Universität Regensburg. (2021). *Lagepläne der Universität Regensburg*. Zugriff auf <https://www.uni-regensburg.de/kontakt/lageplan/index.html>
- van der Maaten, L. & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9 (11).
- van de Sanden, S., Willems, K. & Brengman, M. (2019). In-store location-based marketing with beacons: from inflated expectations to smart use in retailing. *Journal of Marketing Management*, 35 (15-16), 1514–1541.
- van Kuppevelt, D., Meijer, C., Huber, F., van der Ploeg, A., Georgievska, S. & van Hees, V. (2020). Mcfly: Automated deep learning on time series. *SoftwareX*, 12, 100548.
- van Opdenbosch, D., Schroth, G., Huitl, R., Hilsenbeck, S., Garcea, A. & Steinbach, E. (2014). Camera-based indoor positioning using scalable streaming of compressed binary image signatures. In *2014 IEEE International Conference on Image Processing (ICIP)* (S. 2804–2808). IEEE.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (S. 5998–6008).
- Wang, H., Elgohary, A. & Choudhury, R. R. (2012). No Need to War-Drive: Unsupervised Indoor Localization. *Proceedings of the 10th international conference on Mobile systems, applications, and services (MobiSys '12)*, 197–210.

- Wang, J., Chen, Y., Hao, S., Peng, X. & Hu, L. (2019). Deep Learning for Sensor-based Activity Recognition: A Survey. *Pattern Recognition Letters*, 119.
- Wang, Y., Xiu, C., Zhang, X. & Yang, D. (2018). WiFi indoor localization with CSI fingerprinting-based random forest. *Sensors*, 18 (9), 2869.
- Wang, Z., Yan, W. & Oates, T. (2017). Time Series Classification from Scratch with Deep Neural Networks: A Strong Baseline. In *2017 International Joint Conference on Neural Networks (IJCNN)* (S. 1578–1585). IEEE.
- Weinberg, H. (2002). *Using the ADXL202 in Pedometer and Personal Navigation Applications*. Zugriff auf <https://www.analog.com/media/en/technical-documentation/application-notes/513772624AN602.pdf>
- Winter, S., Tomko, M., Vasardani, M., Richter, K.-F., Khoshelham, K. & Kalantari, M. (2019). Infrastructure-Independent Indoor Localization and Navigation. *ACM Computing Surveys*, 52 (3), 1–24.
- Wu, Y., Chen, P., Gu, F., Zheng, X. & Shang, J. (2019). HTrack: An Efficient Heading-aided Map Matching for Indoor Localization and Tracking. *IEEE Sensors Journal*, 3100 - 3110.
- Xiao, Z., Wen, H., Markham, A. & Trigoni, N. (2015). Robust Indoor Positioning With Lifelong Learning. *IEEE Journal on Selected Areas in Communications*, 33 (11), 2287–2301.
- Xie, H., Gu, T., Tao, X., Ye, H. & Lu, J. (2016). A Reliability-Augmented Particle Filter for Magnetic Fingerprinting Based Indoor Localization on Smartphone. *IEEE Transactions on Mobile Computing*, 15 (8), 1877–1892.
- Xin-Yu Lin, Te-Wei Ho, Cheng-Chung Fang, Zui-Shen Yen, Bey-Jing Yang & Feipei Lai. (2015). A mobile indoor positioning system based on iBeacon technology. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (S. 4970–4973). IEEE.
- Xu, Z., Wei, J., Zhu, J. & Yang, W. (2017). A robust floor localization method using inertial and barometer measurements. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE.
- Yan, T., Chu, D., Ganesan, D., Kansal, A. & Liu, J. (2012). Fast app launching for mobile devices using predictive user context. In *Proceedings of the 10th international conference on Mobile systems, applications, and services - MobiSys '12* (S. 113). ACM Press.
- Yang, J., Cheng, K., Chen, J., Zhou, B. & Li, Q. (2018). Smartphones based Online Activity Recognition for Indoor Localization using Deep Convolutional Neural Network. In *2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)* (S. 1–7). IEEE.
- Yang, L., Li, Y. & Rizos, C. (2014). Walking Status Detection for Pedestrian Navigation. In J. Sun, W. Jiao, H. Wu & M. Lu (Hrsg.), *China Satellite Naviga-*

- tion Conference (CSNC) 2014 Proceedings: Volume III* (Bd. 305, S. 485–497). Springer Berlin Heidelberg.
- Yao, S., Hu, S., Zhao, Y., Zhang, A. & Abdelzaher, T. (2017). DeepSense: A Unified Deep Learning Framework for Time-Series Mobile Sensing Data Processing. In *Proceedings of the 26th International Conference on World Wide Web - WWW '17* (S. 351–360). ACM Press.
- Yazici, A., Yayan, U. & Yücel, H. (2011). An ultrasonic based indoor positioning system. In *2011 International Symposium on Innovations in Intelligent Systems and Applications* (S. 585–589).
- Ye, H., Gu, T., Tao, X. & Lu, J. (2016). Scalable floor localization using barometer on smartphone. *Wireless Communications and Mobile Computing*, 16 (16), 2557–2571.
- Ye, J., Li, X., Zhang, X., Zhang, Q. & Chen, W. (2020). Deep Learning-Based Human Activity Real-Time Recognition for Pedestrian Navigation. *Sensors*, 20 (9), 2574.
- Youssef, M. & Agrawala, A. (2005). The Horus WLAN location determination system. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services* (S. 205–218).
- Zafari, F., Gkelias, A. & Leung, K. (2019). A Survey of Indoor Localization Systems and Technologies. *IEEE Communications Surveys & Tutorials*, 21 (3), 2568–2599.
- Zeng, M., Nguyen, L. T., Yu, B., Mengshoel, O. J., Zhu, J., Wu, P. & Zhang, J. (2014). Convolutional Neural Networks for human activity recognition using mobile sensors. In *6th International Conference on Mobile Computing, Applications and Services* (S. 197–205).
- Zhang, S., McCullagh, P., Nugent, C. & Zheng, H. (2010). Activity Monitoring Using a Smart Phone's Accelerometer with Hierarchical Classification. In *2010 Sixth International Conference on Intelligent Environments* (S. 158–163). IEEE.
- Zhou, B., Li, Q., Mao, Q., Tu, W. & Zhang, X. (2015). Activity Sequence-Based Indoor Pedestrian Localization Using Smartphones. *IEEE Transactions on Human-Machine Systems*, 45 (5), 562–574.
- Zhou, B., Yang, J. & Li, Q. (2019). Smartphone-Based Activity Recognition for Indoor Localization Using a Convolutional Neural Network. *Sensors*, 19 (3), 621.
- Zhou, Y., Zheng, X., Xiong, H. & Chen, R. (2017). Robust Indoor Mobile Localization with a Semantic Augmented Route Network Graph. *ISPRS International Journal of Geo-Information*, 6 (7), 221.