

---

# Extraktion von Kontextinformationen zur Analyse von Nutzerströmen

Lorenz Schauer

---

Dissertation  
an der Fakultät für Mathematik, Informatik und Statistik  
der Ludwig-Maximilians-Universität München

vorgelegt von  
Lorenz Schauer

Tag der Einreichung: 08.11.2017



---

# Extraktion von Kontextinformationen zur Analyse von Nutzerströmen

Lorenz Schauer

---

Dissertation  
an der Fakultät für Mathematik, Informatik und Statistik  
der Ludwig-Maximilians-Universität München

vorgelegt von  
Lorenz Schauer

1. Berichterstatter:	Prof. Dr. Claudia Linnhoff-Popien
2. Berichterstatter:	Prof. Dr.-Ing. Georg Carle
Tag der Einreichung:	08.11.2017
Tag der Disputation:	23.03.2018



## **Eidesstattliche Versicherung**

(siehe Promotionsordnung vom 12.07.11, § 8, Abs. 2 Pkt. 5)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

Lorenz Schauer

München, 28.03.2018



# Zusammenfassung

Kontextadaptive Dienste ermitteln auf Basis von Sensordaten die aktuelle Situation des Nutzers. Dabei zählt neben Zeit, Identität und Aktivität vor allem der Aufenthaltsort zu den primären Kontextinformationen. Gerade Gebäude- und Infrastrukturbetreiber haben höchstes Interesse an solchen Daten, bekommen jedoch nur selten Zugriff darauf. Eine vielversprechende Alternative stellt das Aufzeichnen und Analysieren von Bluetooth- bzw. WLAN-Signalen dar, welches sich auch ohne Einverständnis der Nutzer realisieren lässt. Einige Startup-Unternehmen bieten mit dieser Strategie bereits Kundenanalysen an, jedoch fehlen wissenschaftliche Studien über die tatsächliche Güte solcher Verfahren. Zudem stellt die Extraktion primärer Kontextinformationen eine bisher ungelöste Herausforderung dar, da die aufgezeichneten Daten ein hohes Rauschen und eine geringe Frequenz aufweisen.

Die vorliegende Arbeit geht dieses Problem der Datenunsicherheit an. So wird im ersten Teil untersucht, wie und welche Informationen aus Kommunikationsdaten gewonnen werden können. Diese werden anschließend zur Berechnung von Nutzerströmen in einem realen Szenario verwendet. Mithilfe einer bekannten Grundwahrheit können erstmals belastbare Aussagen über die Güte der evaluierten Verfahren getroffen werden. Die Ergebnisse zeigen, dass sich gerade WLAN-Daten zur Nutzerstromanalyse eignen, weshalb anschließend Simulationsmodelle von WLAN-Beobachtungen entwickelt werden. Dabei wird erstmals neben der Zeit auch die Aktivität und das Endgerät berücksichtigt.

Der zweite Teil beschäftigt sich mit der Extraktion der Ortsinformation aus WLAN-Daten, die von mehreren Messknoten aufgezeichnet werden. Hierbei werden zunächst geometrische Ansätze zur Platzierung solcher Knoten vorgestellt. Auf Basis der Ergebnisse werden Messknoten installiert, um Bewegungspfade mobiler Nutzer im Gebäude zu ermitteln. Es zeigt sich, dass deterministische Methoden nicht die nötige Genauigkeit liefern und probabilistische Verfahren andere Nachteile aufweisen. Deshalb wird der State-Partikel-Filter als neuartige hybride Variante vorgestellt, die im Rahmen der Evaluation bessere Ergebnisse in Bezug auf anerkannte Metriken erreicht.

Im dritten Teil wird die Extraktion von Aktivitäts- und Identitätsinformationen untersucht. Hierzu wird ein modifiziertes Verfahren zur Bestimmung von Ruhe- und Bewegungsphasen vorgestellt, das im Vergleich zu verwandten Arbeiten genauer ist. Auf Basis aller verfügbaren Informationen wird schließlich zur Identitätserkennung eine Klassifizierung von Nutzerrollen durchgeführt. Anders als in bisherigen Arbeiten findet so eine komplette Extraktion primärer Kontextinformationen aus aufgezeichneten Kommunikationsdaten statt.

# Abstract

Context-aware services mainly estimate a user's current situation using sensor data. The well-known primary types of context information are time, identity, activity, and above all, location. Building managers and infrastructure providers are particularly interested in such data. However, they hardly get access to it. A promising alternative way without requiring users' permission for data access is to observe and analyse Bluetooth as well as Wi-Fi signals. Some start-ups already apply this technique in order to offer retail analytics. However, scientific studies providing information about the real performance of such approaches are still missing. Furthermore, extracting all of the primary types of context information is still challenging, due to the fact that the recorded data suffers from high noise and low sampling rates.

This work deals with the afore-mentioned problems. Hence, the first chapter investigates, how and which kind of information can be inferred from captured communication data. Subsequently, this information is used to estimate user flows in a realistic scenario. With access to a known ground truth it becomes possible for the first time to reveal the real performance of the evaluated approaches. The results indicate that Wi-Fi captures are suitable for user flow analysis. Hence, models for simulating Wi-Fi based observations are developed considering time and novel aspects such as the user's activity and the device.

The second chapter deals with the extraction of location information from Wi-Fi data which is captured by several distributed monitoring units. As a first step, geometric approaches for the placement of such units are presented. Based on the achieved results, Wi-Fi monitors are deployed in order to track and estimate trajectories of mobile users. In this connection, it is shown that deterministic methods do not reach the required accuracy and probabilistic approaches suffer from different drawbacks. Therefore, the so-called State-Particle-Filter is introduced as a novel hybrid variant returning better results in terms of well-adopted metrics within the performed evaluation.

In the third chapter, the extraction of activity and identity information is investigated. Therefore, a modified approach for recognition of dwell times and motion periods is introduced which is shown to be more accurate than existing techniques. Finally, on the basis of all the available information, a user role classification task is performed for identity discovery. Thus, in contrast to other works, a complete extraction of the well-known primary types of context information is realized using captured communication data and the proposed methods.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	2
1.2	Herausforderungen und Beiträge der Arbeit . . . . .	5
1.3	Veröffentlichte Vorarbeiten . . . . .	7
1.4	Aufbau der Arbeit . . . . .	10
<b>2</b>	<b>Grundlagen zur Erfassung und Analyse von Nutzerströmen</b>	<b>11</b>
2.1	Techniken zur passiven Lokalisierung . . . . .	12
2.1.1	Bluetooth . . . . .	14
2.1.2	WLAN . . . . .	15
2.2	Methoden zur passiven Lokalisierung . . . . .	18
2.2.1	Nahbereichserkennung . . . . .	18
2.2.2	Tirangulation . . . . .	18
2.2.3	Mustererkennung . . . . .	24
2.3	Probabilistische Verfahren zur Pfadermittlung . . . . .	26
2.3.1	Bayes-Filter . . . . .	26
2.3.2	Hidden Markov Model . . . . .	30
2.3.3	Viterbi-Algorithmus . . . . .	32
2.4	Analyse von Bewegungspfaden . . . . .	33
2.4.1	Ähnlichkeit . . . . .	34
2.4.2	Segmentierung . . . . .	36
2.4.3	Clustering . . . . .	38
2.4.4	Bewegungsmuster . . . . .	41
2.4.5	Interessante Orte . . . . .	41
2.5	Zusammenfassung . . . . .	42
<b>3</b>	<b>Datenerfassung zur Analyse von Nutzerströmen</b>	<b>45</b>
3.1	Datenerfassung unter realen Bedingungen . . . . .	45
3.1.1	Verwandte Arbeiten . . . . .	46
3.1.2	Ansätze zur Bestimmung von Personendichte und Nutzerströme	47
3.1.3	Experimenteller Aufbau und Implementierung des Flughafenszenarios . . . . .	49
3.1.4	Evaluation und Ergebnisse des Flughafenszenarios . . . . .	50
3.1.5	Zusammenfassung und Diskussion . . . . .	57
3.2	Simulationsumgebung zur Datenerfassung . . . . .	58
3.2.1	Umgebungsmodell für Gebäude . . . . .	59
3.2.2	Bewegungsmodell für Personen im Gebäude . . . . .	60

3.2.3	Aussendungsmodell für Probe Requests . . . . .	62
3.2.4	Modellvalidierung anhand realer Daten . . . . .	65
3.2.5	Zusammenfassung und Diskussion . . . . .	69
3.3	Zusammenfassung . . . . .	70
<b>4</b>	<b>Extraktion der Ortsinformation zur Nutzerstromanalyse</b>	<b>71</b>
4.1	Strategien zur Platzierung von Messknoten . . . . .	72
4.1.1	Problemstellung der Messknotenverteilung . . . . .	72
4.1.2	Existierende Ansätze von Platzierungsstrategien . . . . .	74
4.1.3	Beiträge zur Erkennung von Hotspots . . . . .	76
4.1.4	Evaluation und Ergebnisse der Hotspoterkennung . . . . .	79
4.1.5	Zusammenfassung und Diskussion . . . . .	85
4.2	Pfadermittlung auf Basis ungenauer Positionsdaten . . . . .	86
4.2.1	Verwandte Arbeiten . . . . .	87
4.2.2	Anwendung bestehender Verfahren . . . . .	88
4.2.3	Der State-Partikel-Filter als hybride Lösung . . . . .	95
4.2.4	Evaluation und Ergebnisse der Pfadermittlung . . . . .	97
4.2.5	Zusammenfassung und Diskussion . . . . .	104
4.3	Zusammenfassung . . . . .	106
<b>5</b>	<b>Extraktion der Aktivitäts- und Identitätsinformation</b>	<b>107</b>
5.1	Bestimmung von Ruhe- und Bewegungsphasen . . . . .	108
5.1.1	Verwandte Arbeiten . . . . .	109
5.1.2	Beiträge zur Bestimmung des Bewegungsstatus . . . . .	110
5.1.3	Evaluation und Ergebnisse der Aktivitätsbestimmung . . . . .	115
5.1.4	Zusammenfassung und Diskussion . . . . .	122
5.2	Klassifizierung von Nutzerrollen . . . . .	123
5.2.1	Verwandte Arbeiten . . . . .	124
5.2.2	Konzept zur Klassifizierung von Nutzerrollen . . . . .	125
5.2.3	Evaluation und Klassifizierungsergebnisse . . . . .	130
5.2.4	Diskussion anhand realer Daten . . . . .	132
5.3	Zusammenfassung . . . . .	135
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>137</b>
6.1	Ergebnisse der Arbeit . . . . .	137
6.2	Ausblick . . . . .	139
	<b>Literaturverzeichnis</b>	<b>141</b>

# Danksagung

An dieser Stelle möchte ich mich ganz herzlich bei einigen Personen bedanken, die mich auf dem langen Weg zur vorliegenden Dissertation unterstützt und motiviert haben.

Ein ganz besonderer Dank gilt dabei der inspirierenden Betreuung von Frau Prof. Dr. Claudia Linnhoff-Popien, die mir das Vertrauen schenkte, um an ihrem Lehrstuhl arbeiten und promovieren zu dürfen. Insbesondere möchte ich mich bei ihr für das vielfältige, angenehme, fruchtbare und vor allem freiheitliche Arbeitsklima am Lehrstuhl bedanken, welches mir ermöglicht hat, kreative Idee umzusetzen und meine Ziele zu verfolgen. Durch die regelmäßig stattfindenden Lehrstuhlworkshops konnten in einem angenehmen Rahmen eigene Forschungsansätze im Team diskutiert und verfeinert werden.

Mein Dank gilt deshalb auch allen meinen Kollegen, die mich nicht nur fachlich durch unzählige Diskussionen und kritische Fragestellungen vorangebracht haben, sondern mir auch im freundschaftlichen Sinne stets eine immense Unterstützung waren. Viele schöne Erinnerungen sind mit diesem starken Team verbunden. Namentlich hervorheben möchte ich an dieser Stelle Dr. Martin Werner für die anfängliche thematische Findungsphase, Dr. Philipp Marcus für zahlreiche inspirierende Gespräche und Programmierereinheiten, Dr. Chadly Marouane für die gegenseitige Unterstützung auch in schwierigen Zeiten und Carsten Hahn, der immer ein offenes Ohr und eine offene Bürotür für mich hatte. Des Weiteren ergeht mein Dank an Herrn Prof. Dr.-Ing. Georg Carle von der TU-München für sein Engagement als Zweitberichterstatter und an Herrn Prof. Dr. Christian Böhm für die Übernahme des Vorsitz der Prüfungskommission.

Ebenfalls möchte ich mich herzlich bei meinem *Burschen*-Freundeskreis bedanken, der mir seit vielen vielen Jahren zur Seite steht und ein fester Bestandteil und Stützpfeiler meines Lebens ist. Vor allem möchte ich mich auch bei meiner Familie und bei meinen liebevollen Eltern bedanken, die mich seit ich denken kann, bei all meinen zahlreichen Interessen und Vorhaben stets unterstützt haben. Mein größter Dank gilt dabei meiner wunderbaren und lieben Frau Maria, die mich in jedem Moment mit viel Liebe und Verständnis motiviert hat. Last but not least ergeht ein ganz besonderer Dank an meinen wundervollen Sohn Anian, der für mich das größte Glück auf Erden ist und der sich mit seiner Geburt Zeit gelassen hat, damit Papa die vorliegende Dissertation in Ruhe zu Ende schreiben konnte.



# 1 Einleitung

*„In the 21st century the technology revolution will move into the everyday, the small and the invisible.”*

— Mark Weiser, 1991

Bereits 1991 skizzierte Mark Weiser in seiner Arbeit *The Computer for the 21st Century* [212] die Vision eines technologischen und digitalen Wandels, den wir aktuell in nahezu allen Bereichen unseres täglichen Lebens erfahren. Schon heute ist unsere Welt durchzogen von Computersystemen, die gemäß dem obigen Zitat immer mehr mit dem Alltäglichen verschmelzen und dabei unsichtbar erscheinen. So werden beispielsweise Autos zu „Smartphones auf Rädern“ [231], unsere Häuser entwickeln sich zu *Smart Homes* [49] und Flughäfen zu *Smart Cities* [179]. Mit solchen Sammelbegriffen wird versucht, die komplette Vernetzung unseres Alltags mit digitalen Technologien zu beschreiben. Weiser bezeichnete diesen Umstand als „Ubiquitous Computing“, womit einhergeht, dass sich Nutzer von einem einzelnen, komplexen PC abwenden und stattdessen mehrere kleine und voll vernetzte Computer nutzen, deren Funktionalität allgegenwärtig zur Verfügung steht. Dabei tritt die Technik mehr und mehr in den Hintergrund und agiert eher automatisiert und proaktiv, als wie bisher reaktiv auf implizites Eingreifen des Nutzers [213].

Den ersten Teil dieser Vision sehen wir bereits durch die enorme Verbreitung moderner mobiler Endgeräte verwirklicht. Schon im Februar 2015 besaßen rund 45,6 Millionen Personen in Deutschland und damit mehr als die Hälfte der Bevölkerung ein *Smartphone*, Tendenz steigend [193]. Weltweit soll sich die Zahl der Nutzer von aktuell 2,1 auf 2,87 Milliarden bis zum Jahr 2020 erhöht haben [194]. Andere Prognosen gehen davon aus, dass dann 33 Milliarden internetfähige Geräte im Umlauf sein werden und somit jeder Mensch im Schnitt 4,3 Geräte besitzt. Dabei sollen mobile Geräte mit Internetanbindung einen immer größeren Anteil einnehmen und die klassischen PCs allmählich verdrängen, was schon heute zu beobachten ist [140].

Der zweite Teil von Weisers Vision setzt voraus, dass sich Computer ihrer Umgebung stets bewusst sind und die Situation bzw. die Bedürfnisse ihrer Nutzer kennen, um komplett automatisiert und damit im Verborgenen agieren zu können. Auch wenn dieser Stand noch nicht komplett erreicht ist, geht der Trend in die skizzierte Richtung. So unterstützen uns bereits heute digitale Assistenten wie Apples *Siri* [227], *Google Now* [223] oder *Cortana* von Microsoft [226] bei alltäglichen Anforderungen, wobei sie durch natürliche Sprache bedient werden können. Solche *kontextsensitiven Anwendungen* sammeln im

Hintergrund wichtige Informationen wie Aufenthalt, Interessen oder Vorlieben des Nutzers, um dessen Situation zu erfassen und die Dienstleistung entsprechend dem *Kontext* des Nutzers anzupassen. Letztendlich bilden dabei geeignete *Kontextinformationen* die Basis für eine adäquate und personalisierte Funktionalität. Neben dem Ort zählen nach Abowd et al. [2] die Zeit, Aktivität und Identität zu den primären und damit wichtigsten Typen von Kontextinformationen, um die Situation eines Nutzers zu charakterisieren. Der Ortsinformation kommt dabei eine Schlüsselrolle zu, da sie die Basis der sogenannten *ortsbezogenen Dienste* (engl.: *location based services*) darstellt. Diese finden heute in unzähligen Bereichen Anwendung, wie bspw. bei der klassischen Navigation, aber auch bei Suchanfragen, in digitalen Werbeanzeigen oder beim Entdecken von interessanten Orten in der näheren Umgebung.

Für die Extraktion von Kontextinformationen erfassen kontextbezogene Dienste typischerweise die physische Umgebung des Nutzers über Sensoren, wie bspw. dem GPS-Sensor für den Aufenthaltsort im Außenbereich oder dem Gyroskop und Beschleunigungssensor für die Bewegung. Gerade die weite Verbreitung mobiler Geräte, die häufig mit den entsprechenden Sensoren ausgestattet sind, hat hier eine wichtige Voraussetzung geschaffen, um kontextbezogene Anwendungen und Dienste für die breite Masse zu entwickeln und anzubieten. Aus diesem Grund stieg in den letzten Jahren neben dem wissenschaftlichen auch das wirtschaftliche Interesse an solchen Diensten, was die Entwicklung der genannten digitalen Assistenten und auch diverse Publikationen in dem Bereich beweisen [239]. Dabei stellt vor allem die Extraktion von Kontextinformationen aus Sensordaten sowie deren Aufbereitung für komplexe Situationen eine kontinuierliche und wesentliche Herausforderung dar, womit sich der Forschungsbereich der *kontextsensitiven und adaptiven Dienste* beschäftigt [44]. Dieser bildet die Ausgangslage zur Motivation der vorliegenden Arbeit.

## 1.1 Motivation

Wissenschaftliche Arbeiten im Bereich der kontextsensitiven und adaptiven Dienste nutzen klassischerweise die vorhandene Sensorik mobiler Endgeräte, um primäre Kontextinformationen wie den Ort [14], den Bewegungsstatus [98] oder andere Nutzeraktivitäten [117] zu extrahieren. Bei all diesen Arbeiten ist stets die Voraussetzung erfüllt, dass eine entsprechende mobile Applikation (App) auf den Geräten installiert ist, um überhaupt Sensordaten erfassen und verwenden zu können. In der Realität wird eine solche App aber nur dann akzeptiert, wenn Nutzer darin einen persönlichen Mehrwert erkennen. Praktische Beispiele hierfür sind etablierte Apps wie *Runtastic* [37] zur adaptiven Unterstützung sportlicher Aktivitäten oder der kontextsensitive Wecker *WakeApp* [38]. Anderenfalls geben Nutzer ihre Sensordaten nicht an Dritte weiter.

Gerade Betreiber von Shops, Infrastrukturen oder öffentlichen Gebäuden haben es häufig schwer, eine App bei ihren Kunden zu etablieren, da diese

meistens keinen ersichtlichen Mehrwert bieten [131]. Somit bleibt ihnen ein breiter Zugriff auf die Sensorik mobiler Endgeräte verwehrt, und klassische Verfahren zur Extraktion von Kontextinformationen können in diesem Fall nicht angewendet werden. Trotzdem besteht bei allen Beteiligten ein enormes Interesse, die gegenwärtige Situation der anwesenden Kunden zu bestimmen und deren Verhalten zu verstehen. So sucht der stationäre Handel seit längerem nach vergleichbaren Verfahren wie in Online-Shops, um das Kaufverhalten der Kunden im Laden zu analysieren. Dabei ist vor allem interessant, wann, wie lange und welche Produkte begutachtet werden, bis eine Kaufentscheidung getroffen wird [184]. Relevante Kontextinformationen sind somit Ort, Zeit und Aktivität.

Infrastrukturbetreiber und Organisatoren großer öffentlicher Gebäude wie Bahnhöfe oder Flughäfen verlangen hingegen nach technischen Lösungen zur Erfassung und Analyse von Personenansammlungen und Nutzerströmen, um ihre Ressourcen besser planen und einsetzen zu können. Dabei bezeichnen Nutzerströme im vorliegenden Fall eine Menge an Personen, die sich zeitlich und örtlich auf ähnliche Weise bewegen. Akkurate Kontextinformationen über Aufenthaltsorte und Aktivitäten der Personen können also zur Interpretation solcher Ströme verwendet werden, um auf aktuelle oder auch zukünftige Ereignisse im Betrieb zu reagieren. So lässt sich bspw. bei erkannten Engpässen Personal hinzuziehen und es können automatisch Schalter, Türen oder ganze Bereiche zusätzlich geöffnet werden. Außerdem besteht die Möglichkeit, Nutzer direkt über ihr mobiles Gerät zu informieren, welche Wege in einem Gebäude aktuell überlaufen sind. Dieses Konzept stellt wiederum die Basis für eine Dichte-basierte Navigation dar [200], die gerade in Evakuierungs- oder Notsituationen hilfreich sein kann, um Staus zu vermeiden. Da realistischere kaum eine Möglichkeit besteht, auf die Sensordaten mobiler Endgeräte im großem Stil zuzugreifen, müssen alternative Infrastruktur-basierte Techniken zur Nutzererfassung gefunden werden. Diese sollten ein ähnliches Potential aufweisen, um geeignete Kontextinformationen zu extrahieren, so dass die genannten Ziele erreicht werden können.

Zu dem Zweck wurden bereits vor der Verbreitung mobiler Endgeräte vor allem fest installierte Kameras und visuelle Verfahren eingesetzt. Damit konnte bspw. die Personendichte bestimmt werden, welche als Maß für die aktuelle Personenanzahl in einem bestimmten Bereich aufgefasst wird [134, 167, 235]. Durch die kontinuierliche Weiterentwicklung von Hard- und Software im Bereich der Computervision existiert heute eine Vielzahl an fortgeschrittenen Methoden, um aus vorhandenem Bildmaterial Kontextinformationen wie Aufenthaltsort [195], Bewegung [169] oder speziellere Nutzeraktivitäten [63] abzuleiten. Dabei erfolgt die Bestimmung des Kontextes komplett passiv, d.h. ohne aktiver Beteiligung oder gar dem Bewusstsein der betreffenden Nutzer. Zudem werden keine mobilen Geräte zur Datenerfassung benötigt. Allerdings ist eine flächendeckende Installation von Kameras nötig, um Personen und deren Aktivitäten in jedem Winkel eines Gebäudes überwachen zu können. Dies kann

vor allem bei großen und komplexen Gebäuden wie Einkaufszentren, Flughäfen oder Universitäten sehr kostspielig sein. Außerdem ist eine komplette Videoüberwachung aus technischen und insbesondere aus Datenschutzgründen nicht überall durchführbar. Gerade letzter Punkt wird in Hinblick auf immer weitreichendere Methoden in der Computervision stark diskutiert [158].

Um derartige Nachteile zu vermeiden, bietet sich mittlerweile eine einfachere und kostengünstigere Möglichkeit an, die auf der Datenaufzeichnung drahtloser Kommunikationstechnologien wie *Bluetooth* oder *WLAN* basiert. Beide Technologien sind heutzutage in nahezu allen Smartphones und Tablets als standardisierte Kommunikationsschnittstellen integriert. Sobald sie aktiviert sind, werden vor einem möglichen Verbindungsaufbau unverschlüsselte Managementpakete mit gerätespezifischen Informationen an die Umgebung versendet. Diese können ohne explizitem Einverständnis der Nutzer von einfachen Messknoten im Sendebereich aufgezeichnet und interpretiert werden. Vor allem anhand der übermittelten Hardware- bzw. MAC-Adresse kann ein Gerät eindeutig identifiziert und damit über weite Strecken verfolgt und wiedererkannt werden. Durch die ständig wachsende Verbreitung mobiler Geräte und der damit einhergehenden verstärkten Nutzung von Bluetooth und insbesondere WLAN sind solche Daten mittlerweile zur flächendeckenden Erfassung von Personen interessant. So verwenden bereits einige Arbeiten Bluetooth, um die unmittelbare Personendichte [215], menschliche Bewegungen [205] oder das Verhalten von Menschen, die sich in Empfangsreichweite befinden [118], zu untersuchen.

WLAN zeigt gegenüber Bluetooth einige Vorteile bei der automatischen Erfassung mobiler Geräte, vor allem durch schnellere und genauere Erkennungs-raten [1]. Zusätzlich folgt der WLAN-Standard *IEEE 802.11* [92] dem ubiquitären Prinzip von Mark Weiser. So suchen mobile Endgeräte mit aktiviertem WLAN selbständig und ohne Zutun des Nutzers immer wieder nach geeigneten Zugangspunkten (engl.: *access points*), wobei die erwähnten Managementpakete ausgesendet werden. Da die benötigten Infrastrukturen in sehr vielen öffentlichen Einrichtungen und Plätzen vorhanden sind, lassen die meisten Nutzer das WLAN ihres mobilen Gerätes während sie unterwegs sind aktiviert, um bequem und schnell eine kostengünstige Verbindung ins Internet aufbauen zu können. Prinzipiell lassen sich somit viele Geräte an unterschiedlichen Orten auf einfache Weise aufspüren.

Aus diesem Grund setzten aktuelle Forschungsarbeiten vermehrt auf WLAN-Signale, um die Personendichte [26] oder ganze Nutzerströme [45] in einem bestimmten Bereich zu erfassen. Zudem wurde bereits untersucht, in wie weit Kontextinformationen wie der Ort [232], Bewegungspfade [145], Aktivitäten [188] und Nutzerrollen [171] aber auch soziale Verbindungen [20] oder die Proximität zu anderen Nutzern [133] abgeleitet werden können. Neben dem enormen wissenschaftlichen Interesse hat auch die Wirtschaft das Potential von WLAN als Tracking-Technologie erkannt. So entstanden gerade in den letzten Jahren einige Startup-Unternehmen, welche diese Strategie nutzen, um

Kunden- und Nutzerstromanalysen für den Einzelhandel oder für Flughäfen anzubieten. Dabei überschneiden sich die verschiedenen Angebote und reichen von der Wiedererkennung von Personen, über die Bestimmung der Personendichte bis hin zur Lokalisierung und Klassifikation von Kunden [176].

Trotz der reichlich vorhandenen Vorarbeiten und Projekte auf diesem Gebiet bleiben wichtige Fragestellungen und Herausforderungen bestehen, die in bisherigen Arbeiten nicht hinreichend gelöst werden konnten. Diese werden im Folgenden vorgestellt und im Rahmen der vorliegenden Arbeit behandelt.

## 1.2 Herausforderungen und Beiträge der Arbeit

Für eine Nutzerstromanalyse auf Basis aufgezeichneter Bluetooth- bzw. WLAN-Daten ist es zunächst wichtig, beide Technologien in einem realen Szenario hinsichtlich ihrer Qualität zur Erfassung mobiler Endgeräte zu untersuchen. Meistens fehlt aber die dazu notwendige Grundwahrheit, weshalb viele Arbeiten bei der Evaluation lediglich auf Schätzungen zurückgreifen [26, 101, 164]. In der vorliegenden Arbeit hingegen werden die eingeführten Verfahren auf Basis einer realen Studie am Münchner Flughafen evaluiert. Dabei war der Versuchsaufbau so gewählt, dass die gemessenen Nutzerströme durch eine Sicherheitsschleuse verliefen, bei der die Bordkarten der Passagiere abgescannt wurden. Diese Daten standen dem Autor zur Verfügung, wodurch die einzelnen Verfahren erstmalig auf Basis einer bekannten Grundwahrheit bewertet werden können.

Da das Problem der fehlenden Grundwahrheit für reale Szenarien fortlaufend besteht, wird auf Basis der zuvor gewonnenen Erkenntnisse eine Simulationsumgebung entwickelt. Dabei wird vor allem das unregelmäßige Sendeverhalten der mobilen Endgeräte durch ein neuartiges Modell beschrieben, welches neben der zeitlichen Dimension auch die Aktivität und das Gerät des Nutzers berücksichtigt. Vergleichbare Arbeiten haben diese Aspekte so bisher nicht modelliert, sondern sind eher von konstanten Werten [70, 145] oder der generellen Erfassbarkeit von WLAN-aktiven Geräten ausgegangen [90]. Durch die hier entwickelte Simulationsumgebung können beliebige Szenarien mit entsprechender Grundwahrheit zur Evaluation weiterer Aspekte generiert werden.

Sowohl bei der Simulation als auch in realen Szenarien müssen entsprechende Messknoten, sogenannte *WLAN-Monitore*, im gewünschten Bereich installiert werden, um Signale von vorbeilaufenden Nutzern aufzeichnen zu können. Erst durch die Verteilung mehrerer Knoten ist es möglich, mobile Geräte über längere Strecken zu verfolgen und Bewegungsabläufe zu bestimmen. Obwohl die Anzahl und die Platzierung von Messknoten einen Einfluss auf Genauigkeit und Skalierbarkeit des Tracking-Systems haben, gehen verwandte Arbeiten nicht im Detail darauf ein [26, 86]. Daher soll diese Problematik im Rahmen der vorliegenden Arbeit näher untersucht werden. Konkret werden drei geometrische Ansätze vorgestellt, die auf Basis der oben erwähnten Simulationsumgebung beliebige Orte (engl.: *Hotspots*) in einem Gebäude identifizieren. Diese werden

dann als Grundlage zum Platzieren von Messknoten verwendet.

Auf Grundlage des zurückgelieferten Lösungsraums werden anschließend WLAN-Monitore in einem Universitätsgebäude platziert, um in einer realen Versuchsumgebung eine Standortverfolgung von Nutzern vorzunehmen. Allerdings ist eine kontinuierliche und exakte Lokalisierung auf Basis von WLAN-Beobachtungen schwer umzusetzen, da nicht eindeutig bestimmt werden kann, wann und ob eine Beobachtung gemacht wird. Außerdem kommt es innerhalb von Gebäuden durch Personen, Wände oder andere Hindernisse vermehrt zur Signaldämpfung oder Mehrwegeausbreitung, die zu veräuschten bzw. veränderten Messwerten führen [137,177]. Die dadurch entstehenden Positionsfehler erschweren die genaue Standortverfolgung. Im Rahmen der vorliegenden Arbeit werden bestehende probabilistische Verfahren untersucht, da diese auf Basis geeigneter Bewegungsmodelle die Position genauer abschätzen können. Allerdings zeigen diese unterschiedliche Nachteile in Bezug auf anerkannte Metriken aus der *Indoor-Positionierung*. Aus diesem Grund wird im weiteren Verlauf ein neuartiges hybrides Verfahren entwickelt, welches einen Partikel-Filter für Freiflächen auf einem vordefinierten Zustandsgraphen in Gebäuden nutzt. Dadurch soll die Standortverfolgung in Bezug auf die untersuchten Metriken im Vergleich zu bestehenden Verfahren verbessert werden.

Jedoch lassen sich Positionsfehler und Abweichungen bei der Standortverfolgung auch durch die genannten Verfahren kaum vermeiden. Durch die vorhandene Ungenauigkeit ist es nur begrenzt möglich, den exakten Bewegungsstatus einer Person wie *Stehen*, *Gehen* oder *Laufen* rein auf Basis der ermittelten Positionsdaten zu bestimmen. Der Bewegungsstatus liefert allerdings wertvolle Erkenntnisse über die Nutzeraktivität und zählt damit zu den primären Kontextinformationen. Daher wird im Rahmen der vorliegenden Arbeit ein modifizierter Ansatz untersucht, welcher den Unterschied im Signalverlauf nutzt, um zwischen Ruhe- und Bewegungsphasen eines Nutzers zu unterscheiden. Da verwandte Arbeiten häufig nur die Varianz des stärksten Signals betrachten [106,116,143], werden hier verschiedene Distanzfunktionen untersucht, mit deren Hilfe sich Wahrscheinlichkeiten für den aktuellen Bewegungsstatus ableiten lassen. Zusätzlich wird ein verändertes Modell eingesetzt, um die Genauigkeit und Generalisierbarkeit des Verfahrens zu erhöhen. Der vorgestellte Ansatz wird auf Basis simulierter Nutzer mit unterschiedlichem Verhalten evaluiert. Zudem wird untersucht, inwieweit die extrahierte Aktivitätsinformation das zugrunde liegende Bewegungsmodell des Freiflächen-Partikel-Filters adaptiv unterstützen kann, um genauere Bewegungspfade zu ermitteln.

Auf Basis der zuvor extrahierten Kontextinformationen wie Zeit, Ort und Aktivität lässt sich die Identität über eine Klassifizierung von Nutzerrollen herleiten. Dieses Ziel wird in der Literatur bereits diskutiert [147] und evaluiert [171]. Im Unterschied zu verwandten Arbeiten, werden hier die entwickelten Verfahren genutzt, um zunächst unterschiedliche Gruppen von statistischen Eigenschaften (engl.: *Features*) zu berechnen. Anschließend werden mithilfe anerkannter Methoden aus dem Bereich des *maschinellen Lernens* die errech-

neten Merkmale in vordefinierte Klassen eingeordnet und auf Basis simulierter Nutzergruppen evaluiert. Durch eine abschließende Diskussion auf Basis realer Daten lassen sich neue Erkenntnisse zur Einordnung von Nutzerströmen gewinnen.

Zusammenfassend wird ein Gesamtkonzept entwickelt, welches mithilfe neuartiger Verfahren eine vollständige Extraktion primärer Kontextinformationen aus aufgezeichneten Kommunikationsdaten erreicht. Ein solches Vorgehen konnte in der gesichteten Literatur bisher nicht entdeckt werden. Die hier gewonnenen Erkenntnisse sind vor allem für Gebäude-, Shop- und Infrastrukturbetreiber interessant, die versuchen Personenströme mit wenig Aufwand und ohne aktive Nutzerbeteiligung zu erfassen und zu analysieren.

## 1.3 Veröffentlichte Vorarbeiten

Die eben skizzierten Lösungsansätze bilden die Kerninhalte der vorliegenden Arbeit. Diese wurden bereits zum größten Teil als Vorarbeiten auf internationalen wissenschaftlichen Konferenzen bzw. in Journals veröffentlicht. Im Folgenden wird ein Überblick über diese Publikationen gegeben, wobei der jeweilige Anteil der beteiligten Autoren beschrieben wird. In den späteren Kapiteln wird inhaltlich auf die zugehörigen Vorveröffentlichungen verwiesen. Allgemein gilt, dass Frau Prof. Dr. Claudia Linnhoff-Popien als Doktormutter des Verfassers der vorliegenden Arbeit stets beratend zur Seite stand, vor allem wenn sie als Koautorin gelistet ist.

- **L. Schauer, M. Werner und P. Marcus: Estimating Crowd Densities and Pedestrian Flows Using Wi-Fi and Bluetooth, 11th international conference on Mobile and Ubiquitous Systems: computing, networking and services, ICST, 2014 [185]:** In dieser Arbeit werden grundlegende Ansätze vorgestellt, um generelle Informationen sowie Personendichte und Nutzerströme auf Basis von aufgezeichneten Bluetooth- und WLAN-Signalen zu erfassen. Die Idee entstand im Rahmen des Projekts „Seamless Travel“, das in Kooperation mit dem Flughafen München durchgeführt wurde. Dabei sollte unter anderem ein kostengünstiges und einfaches Verfahren zur Nutzerstromerfassung entwickelt und im Bereich des Flughafens evaluiert werden. Dr. Martin Werner hat dabei vor allem mit seinem Wissen über Korrelationsanalysen beigetragen. Philipp Marcus war als Kritikgeber und Diskussionspartner beteiligt. Die Idee und alle Hauptinhalte wie Konzeption und Implementierung sowie die Durchführung und die Evaluation stammen vom Autor der vorliegenden Arbeit. Diese veröffentlichten Inhalte finden sich zum größten Teil in Kapitel 3.1 wieder, in welchem die Güte der Datenerfassung unter realen Bedingungen evaluiert wird.

- **L. Schauer und M. Werner: Analyzing Pedestrian Flows based on Wi-Fi and Bluetooth Captures, EAI Endorsed Transactions on Ubiquitous Environments, ICST, 2015 [182]:** Diese Arbeit stellt eine erweiterte Version der vorherigen Publikation [185] dar und wurde als Journal-Beitrag veröffentlicht. Darin werden vor allem zusätzliche Aspekte zur Bestimmung von Wartezeiten an der Sicherheitsschleuse des Münchner Flughafens evaluiert und diskutiert. Hierbei stand Dr. Martin Werner beratend zur Seite, während die additiven Inhalte vom Autor der vorliegenden Arbeit stammen. Aus dieser Publikation wurde lediglich die Abbildung 3.6 in das Unterkapitel 3.1 für eine abschließende Bewertung der Evaluation übernommen.
  
- **L. Schauer und M. Werner: Clustering of Inertial Indoor Positioning Data, 1st KuVS expert talk on localization, Aachener Informatik-Berichte, 2015 [183]:** Dieser Beitrag beinhaltet eine Vorarbeit für die nachfolgende Publikation [175] und wurde im Rahmen des *KuVS* Fachgespräch für Lokalisierung veröffentlicht. Darin wird der *TRACCLUS*-Algorithmus [121] auf seine Eignung für Indoor-Trajektorien untersucht und einige Modifikationen zur Segmentierung und zum Clustering vorgeschlagen. Diese werden anschließend auf einem realen Datensatz evaluiert, welcher mittels Smartphone-Sensoren von Corina-Kim Schindhelm aufgenommen und von Dr. Martin Werner zur Verfügung gestellt wurde. Herr Werner hat außerdem den Großteil der schriftlichen Arbeit geleistet und stand für die Evaluation mit hilfreichen Diskussionsbeiträgen beratend zur Seite. Die Idee sowie die Konzeption und die Evaluation stammen vom Autor der vorliegenden Arbeit. In Unterkapitel 4.1.3 wird auf diese Veröffentlichung Bezug genommen und die vorgestellte Modifikation des *TRACCLUS*-Algorithmus verwendet.
  
- **L. Schauer: Discovering Hotspots: A Placement Strategy for Wi-Fi based Trajectory Monitoring within Buildings, SAI intelligent systems conference, IEEE, 2015 [175]:** Diese Arbeit greift die Idee des eben erwähnten *TRACCLUS*-Algorithmus auf, um eine effiziente Verteilung von Messknoten in einem gegebenen Gebäude zu erreichen. In der Arbeit werden noch zwei weitere Ansätze zur Bestimmung von stärker frequentierten Bereichen – sogenannter *Hotspots* – entwickelt und auf Basis simulierter Bewegungspfade für verschiedene Gebäudepläne evaluiert. Dazu wird eine Modifikation des *Pathway Mobility Models* [18] vorgestellt und verwendet. Alle Inhalte dieser Publikation stammen dabei vom Alleinautor, also dem Verfasser der vorliegenden Arbeit. Diese Inhalte fließen überwiegend in das Kapitel 4.1 ein, welches sich mit der optimalen Platzierung von Messknoten auseinandersetzt. Zudem wird das vorgestellte Umgebungsmodell zusammen mit dem modifizierten *Pathway Mobility Model* für die Simulationsumgebung in Kapitel 3.2 wiederverwendet.

- **L. Schauer, P. Marcus und C. Linnhoff-Popien: Towards Feasible Wi-Fi based Indoor Tracking Systems Using Probabilistic Methods, 7th international conference on Indoor Positioning and Indoor Navigation (IPIN), IEEE, 2016 [181]:** In dieser umfassenden Vorarbeit werden vor allem probabilistische Verfahren zur Positionierung und Standortverfolgung auf Basis von WLAN-Beobachtungen in Gebäuden untersucht. Zudem wird der sogenannte *State-Partikel-Filter* vorgestellt und anhand anerkannter Metriken evaluiert. Dieser stellt eine neuartige Kombination des Freiflächen-Partikel-Filters und des auf einem festen Zustandsgraphen arbeitenden Viterbi-Algorithmus [61] dar. Die Arbeit profitierte dabei besonders vom Vorwissen und den Ideen von Dr. Philipp Marcus, der sich in seiner eigenen Dissertation mit Positionsfehlern in standortbasierter Autorisierung beschäftigt hat [135]. Von ihm wurden auch große Teile der Implementierungsarbeit geleistet, insbesondere die Basisimplementierungen zum Partikel-Filter und dem Viterbi-Algorithmus sowie ein graphisches Tool zum Anzeigen und Vergleichen von Trajektorien. Außerdem war er bei der Konzeption des State-Partikel-Filters maßgeblich mitbeteiligt und lieferte in gemeinsamen Diskussionen wichtige Beiträge zu probabilistischen Verfahren. Die restlichen Hauptinhalte stammen hingegen vom Autor der vorliegenden Arbeit. Darunter fällt vor allem die komplette Evaluation, die Vorbereitung und Durchführung der notwendigen Experimente, die restliche Implementierungsarbeit sowie die grundsätzliche Idee zur Standortverfolgung von Nutzern auf Basis von WLAN-Beobachtungen. Diese Inhalte fließen fast vollständig in Kapitel 4.2 ein, in welchem die Pfadermittlung unter Verwendung unsicherer Positionsdaten untersucht wird.
  
- **L. Schauer und C. Linnhoff-Popien: Extracting Context Information from Wi-Fi Captures, 10th international conference on Pervasive Technologies Related to Assistive Environments, ACM, 2017 [180]:** In dieser Publikation werden Ruhe- und Bewegungsphasen über signifikante Schwankungen im Signalverlauf approximiert. Dabei werden verschiedene Distanzfunktionen verglichen und bestehende Ansätze erweitert, um solche Aktivitäten genauer als bisher erkennen zu können. Anschließend wird untersucht, ob auf Basis der extrahierten Aktivitätsinformationen einerseits das Bewegungsmodell des State-Partikel-Filters verbessert werden kann und ob sich andererseits Nutzerrollen genauer klassifizieren lassen. Auch hierbei stammen sämtliche Inhalte vom Autor der vorliegenden Arbeit. Diese Inhalte finden sich überwiegend in den beiden Kapiteln 5.1 und 5.2 wieder. Dabei behandelt Kapitel 5.1 die Aktivitätserkennung und Kapitel 5.2 die Klassifizierung von Nutzerrollen. Das in dieser Publikation vorgestellte Ausbreitungsmodell von Probe Requests wird hingegen in Kapitel 3.2 zur Entwicklung der Simulationsumgebung verwendet. Die zugehörige Modellvalidierung wurde in der vorliegenden Arbeit im Unterkapitel 3.2.4 neu durchgeführt und ist somit noch nicht veröffentlicht.

Neben den eben genannten Veröffentlichungen ist der Autor der vorliegenden Arbeit an zahlreichen weiteren Publikationen als Haupt- bzw. Koautor beteiligt [133, 176–179, 184, 219, 220]. Deren Inhalte fließen jedoch nicht unmittelbar in die Dissertation ein, sondern werden lediglich als wissenschaftliche Quellen an den entsprechenden Stellen zitiert.

### 1.4 Aufbau der Arbeit

In Kapitel 2 werden zunächst wichtige Grundlagen für das weitere Verständnis erörtert. Dabei werden sowohl die technischen Aspekte zur Standortverfolgung über Bluetooth und WLAN, als auch bestehende Verfahren zur Lokalisierung und Pfadermittlung mobiler Geräte beschrieben. Zudem wird auf anerkannte Methoden zur Analyse von Bewegungspfaden eingegangen, die für eine Bewertung von Nutzerströmen relevant sind.

Im anschließenden Kapitel 3 wird die Datenerfassung und die Eignung der vorliegenden Signalbeobachtungen zur Analyse von Nutzerströmen intensiv untersucht. Hierzu wird ein Experiment in einer realen Umgebung mit bekannter Grundwahrheit durchgeführt. Auf Basis der gewonnenen Erkenntnisse wird anschließend eine realitätsnahe Simulationsumgebung für WLAN-Beobachtungen entwickelt und anhand realer Daten validiert.

Darauf aufbauend beschäftigt sich Kapitel 4 mit der Extraktion von Ortsinformationen aus aufgezeichneten WLAN-Signalen. Zu diesem Zweck werden zunächst verschiedene Verfahren vorgestellt, die es ermöglichen, Messknoten in einer Umgebung geeignet zu platzieren. Auf Grundlage eines solchen Ergebnisses wird ein reales Experiment zur Pfadermittlung durchgeführt, wobei anerkannte Methoden und der State-Partikel-Filter als neuartiger hybrider Ansatz auf Basis bewährter Metriken evaluiert und verglichen werden.

In Kapitel 5 steht die Extraktion von Aktivitäts- und Identitätsinformationen im Fokus. Hierbei werden verschiedene Methoden zur Erkennung von Ruhe- und Bewegungsphasen auf Basis der Signalvarianzen untersucht und mit verwandten Arbeiten verglichen. Anschließend folgt auf Basis der verfügbaren Informationen eine Berechnung von statistischen Eigenschaften, mit deren Hilfe eine Klassifizierung von Nutzerrollen durchgeführt und evaluiert wird.

Abschließend fasst Kapitel 6 die Ergebnisse der vorliegenden Arbeit zusammen. Weiterführende Aspekte werden als Ausblick für zukünftige Forschungsinhalte diskutiert.

## 2 Grundlagen zur Erfassung und Analyse von Nutzerströmen

Zur Erfassung und Analyse von Nutzerströmen spielt die zeitlich kontinuierliche Bestimmung von akkuraten Nutzerpositionen eine wesentliche Rolle. Die resultierenden Zeitserien von Positionsdaten werden räumlich-zeitliche Trajektorien genannt. Mit diesen lassen sich einzelne Bewegungspfade beschreiben und ganze Personenströme ableiten [99]. Die Positionsbestimmung selbst hat bereits eine lange Tradition in der Forschung. So existieren heute zahlreiche Verfahren und Ansätze für den Außen- und insbesondere für den Innenbereich, um die eigene Position zu ermitteln oder Objekte zu orten [82,88,111,130]. Alle gängigen Positionierungssysteme lassen sich in *Endgerät-zentrische*, *Endgerät-unterstützte* und *Infrastruktur-basierte* Systeme aufteilen [13,103]. Bei Ersterem übernimmt das Endgerät sowohl die Messung der relevanten Sensordaten als auch die anschließende Positionsbestimmung. Bei Endgerät-unterstützten Ansätzen werden zwar die Daten vom Gerät gemessen, allerdings werden diese an eine leistungsstarke Infrastruktur gesendet, wo die eigentliche Positionsbestimmung stattfindet. Reine Infrastruktur-basierte Systeme brauchen hingegen keinen Zugriff auf das Endgerät. Sie fangen ausgesendete Signale auf und werten diese aus, um den Aufenthaltsort des entsprechenden Gerätes zu bestimmen. Wie in Kapitel 1.1 erwähnt, haben Betreiber von Shops oder größeren Einrichtungen nur selten Zugriff auf die mobilen Geräte der anwesenden Kunden. Daher bietet sich für sie nur eine Infrastruktur-basierte Lösung zur Erfassung von Nutzerströmen an. Solche Systeme werden generell für die Ortung bzw. Standortverfolgung bevorzugt, da die Soft- bzw. Hardware der beteiligten Endgeräte in der Regel nicht verändert werden muss. Häufig erfolgt die Positionsbestimmung dabei ohne aktive Beteiligung oder gar dem expliziten Einverständnis der Nutzer. Deshalb sprechen verwandte Arbeiten auch von einer *passiven Positionierung* bzw. von einer *passiven Lokalisierung* [107,129,187].

In diesem Kapitel werden die technischen und methodischen Grundlagen der passiven Lokalisierung beschrieben. Da die resultierenden Positionsdaten in der Regel fehlerbehaftet sind, d.h., die ermittelten Werte weichen von den tatsächlichen Aufenthaltsorten ab, bieten sich für eine genauere Pfad-ermittlung probabilistische Verfahren an. Diese schätzen den tatsächlichen Ort bzw. die Trajektorie des Nutzers mithilfe eines geeigneten Bewegungsmodells und einem Vorwissen über die bereits gemessenen Positionen. Hierbei gibt es mehrere Ansätze und anerkannte Algorithmen, die im Verlauf dieses Kapitels erörtert werden. Auf Basis der ermittelten Trajektorien von Einzelpersonen können

schließlich komplette Nutzerströme hinsichtlich zeitlicher und örtlicher Informationen analysiert werden. Dazu existieren unterschiedliche Methoden, mit deren Hilfe Bewegungspfade anhand bestimmter Eigenschaften verglichen werden können. Der letzte Abschnitt des vorliegenden Kapitels stellt diese Methoden vor, wodurch wichtige Grundlagen zum Umgang mit räumlich-zeitlichen Trajektorien aufgezeigt werden.

## 2.1 Techniken zur passiven Lokalisierung

Wie bereits erwähnt, werden für rein Infrastruktur-basierte Tracking-Systeme seit längerem fest installierte Kameras eingesetzt, die ein bestimmtes Gebiet oder ein komplettes Gebäude überwachen. Dabei ist weniger die Positionsbestimmung, sondern die Wiedererkennung und Verfolgung von Personen problematisch, die sich durch verschiedene Kamerabilder bewegen. Um diese Herausforderungen anzugehen, nutzen solche Systeme unterschiedliche Verfahren der Bildverarbeitung wie bspw. *Window-* oder *Feature-Tracking* [202]. Seit neuem werden auch vermehrt Ansätze aus dem Bereich des maschinellen Lernens verwendet. Dabei kommen vermehrt *convolutional neuronal networks* (CNNs) zum Einsatz, die zum Erlernen bestimmter Eigenschaften und Modelle verwendet werden und zu einer verbesserten optischen Identifizierung von sich bewegenden Objekten führen [47, 148, 196].

Zusätzlich zu Kameras werden häufig auch andere Sensoren genutzt, um die Performanz weiter zu steigern [138]. Bekannte Beispiele hierfür sind Licht-, Schall-, Druck- und vor allem Funk-basierte Sensoren. Diese lassen sich auch einzeln oder gebündelt zur passiven Lokalisierung von Personen einsetzen. So nutzen bspw. das *Active Badge Location System* [210] oder das *ParcTab-System* [3] an der Decke montierte Infrarotsensoren, die entsprechende Lichtsignale von Nutzern registrieren, um deren Position zu bestimmen. Im *Active Bat System* [211] werden hingegen Schallsensoren verwendet. Wegen der geringeren Ausbreitungsgeschwindigkeit von Schallwellen lässt sich über ein Mikrofonarray auch der Eingangswinkel und somit die Richtung eines Schallsignals bestimmen, was wiederum zur passiven Lokalisierung verwendet werden kann [25]. Das *Smart Floor System* [156] hingegen setzt Drucksensoren in Bodenplatten zur Identifizierung und Positionierung ein. Wie bei fest installierten Kameras ist hier nicht die Lokalisierung, sondern die Unterscheidung der verschiedenen Personen problematisch. Ein aktuelleres und bekanntes Beispiel zur Erfassung feingranularer und präziser Bewegungen auf Basis gebündelter Sensoren stellt die *Microsoft Kinect* [241] dar. Diese wurde zur Steuerung einer Spielekonsole entwickelt und hat daher nur eine begrenzte Reichweite. Sie verwendet unter anderem einen Tiefensensor, eine Farb- und Infrarotkamera sowie ein Array von vier Mikrofonen. Ein solches Zusammenspiel unterschiedlicher Sensoren wird allgemein auch als *Sensorfusion* bezeichnet [55], worauf diverse Tracking-Systeme basieren [23, 108, 141].

Seit einiger Zeit werden vermehrt Funktechnologien zur passiven Lokalisie-

rung und zur Standortverfolgung eingesetzt. Das liegt zum einen an der hohen Verfügbarkeit durch die weite Verbreitung entsprechender Geräte und Infrastrukturen, zum anderen an der hohen Genauigkeit, die von Funk-basierten Systemen erreicht werden kann [217]. Einige Arbeiten setzen dabei noch spezielle Transponder, sogenannte *Radio Frequency Identification Tags* (RFID) ein, welche einen eindeutigen Code zur Identifizierung an einen entsprechenden Empfänger übermitteln. Die Reichweite solcher Systeme variiert je nach Hardwareeinsatz extrem. So besitzen passive Tags, die ihre Stromversorgung mittels Induktion über das Empfangsgerät beziehen, lediglich einen Aktionsradius von wenigen Zentimetern. Aktive Tags mit eigener Energieversorgung können hingegen mehrere Kilometer überbrücken. Dadurch definieren sich auch die unterschiedlichen Einsatzszenarien. Die kostengünstigen passiven Tags sind meist in großer Anzahl an den zu überwachenden Objekten angebracht und werden direkt von strategisch platzierten Empfängern erfasst [162, 240]. Die teureren aktiven Tags kommen stattdessen in geringeren Mengen bei einer weitflächigen Überwachung zum Einsatz [5, 150].

Heutzutage nutzen immer mehr Funk-basierte Systeme die vorhandenen Signale mobiler Endgeräte, um diese zu orten. Der Hauptgrund ist, dass im Gegensatz zu den bisher vorgestellten Verfahren keine Spezialhardware benötigt wird, weder auf Seiten der Nutzer (wie Infrarotsender oder RFID-Tags) noch auf Infrastrukturebene (wie Kameras oder Drucksensoren). Die weite Verbreitung mobiler Geräte und der entsprechenden Funkzellen bzw. WLAN-Zugangspunkte hat bereits die technischen Voraussetzungen für eine kostengünstige und flächendeckende Nutzerüberwachung geschaffen. Allerdings ist hier eine akkurate und kontinuierliche Lokalisierung im besonderen Maße herausfordernd, da nicht auf präzise Sensorwerte, sondern nur auf vorhandene Signalbeobachtungen zurückgegriffen werden kann, die zum Teil einem hohen Rauschen unterliegen. Dies führt unter anderem dazu, dass ein mobiles Gerät nicht immer beobachtet bzw. lokalisiert werden kann, selbst dann nicht, wenn es sich in Reichweite einer Messstation befindet. Je nach Sende- und Nutzerverhalten entstehen so zum Teil sehr lückenhafte Zeitserien von Beobachtungen, mit denen ein adäquates Tracking-System umgehen muss [89]. Zudem werden die Signale vor allem innerhalb von Gebäuden durch Effekte wie Dämpfung, Reflektion oder Mehrwegeausbreitung verändert und führen so zu verfälschten Messwerten, wodurch eine exakte Positionsbestimmung erschwert wird [137].

Aufgrund dieser Eigenschaften stieg in den letzten Jahren auch das wissenschaftliche Interesse, mobile Endgeräte für die passive Lokalisierung und zur Standortverfolgung zu verwenden. Dabei nutzen die meisten Arbeiten sowohl im Innen- als auch im Außenbereich aufgezeichnete Bluetooth- bzw. WLAN-Signale [83, 171]. Beide Technologien stellen die technische Ausgangslage zur vorliegenden Arbeit dar und werden daher näher betrachtet.

### 2.1.1 Bluetooth

Bluetooth ist ein Industriestandard für WPANs (engl. *Wireless Personal Area Networks*) zur kabellosen Datenübertragung, der in IEEE 802.15.1 [91] spezifiziert ist. Es wurde für Punkt zu Punkt bzw. Ad-hoc Übertragungen zwischen Geräten über kurze Distanzen konzipiert und arbeitet wie WLAN auf den lizenzfreien ISM-Bändern mit ca. 2,4 GHz, wobei ein Frequenzsprungverfahren für eine robustere Kommunikation eingesetzt wird. Ein Bluetooth-Gerät kann dabei in drei Klassen gemäß seiner Ausgangsleistung (zwischen 0 und 20 dBm) eingeteilt werden. Damit ist auch seine Reichweite verknüpft, die je nach Hardwareeinsatz und Umweltfaktoren zwischen einem und 100 Meter betragen kann [1, 224]. Mobile Endgeräte wie Smartphones oder Tablets nutzen in der Regel eine Bluetooth-Schnittstelle der Klasse zwei mit einer Reichweite von ca. zehn Metern [225].

Zur Kommunikation wird ein sogenanntes *Piconet* von einem *Master* aufgebaut, der bis zu sieben Teilnehmer (engl.: *Slaves*) über 3-Bit-Adressen ansprechen kann. Zudem gibt es die Möglichkeit, dass einzelne Geräte in mehreren solcher Piconets aktiv sind, wodurch insgesamt ein größeres *Scatternetz* geformt wird. Das kleinste Netz wiederum besteht zwischen einem Master- und einem Slave-Gerät. Um ein solches aufzubauen definiert der IEEE 802.15.1 Standard zwei sequentielle Prozesse: Den *Inquiry*- und den anschließenden *Paging*-Prozess. Während der Erste für das Scannen der Umgebung zum Auffinden möglicher Teilnehmer verantwortlich ist, kümmert sich Letzterer um die Aushandlung von Kommunikationsparameter, um schlussendlich eine Verbindung herzustellen. Zur passiven Nutzererfassung ist jedoch nur der Inquiry-Prozess interessant, da es hier rein um die Erkennung eines mobilen Gerätes geht, ohne dass mit diesem eine Verbindung hergestellt wird. Dabei kommen zwei Protokolle zum Einsatz, die eine aktive und eine passive Rolle definieren. Das *Inquiry-Protokoll* übernimmt den aktiven Teil und wird vom zukünftigen Master realisiert, der aktiv auf allen zur Verfügung stehenden Kanälen ein Inquiry-Anfrage sendet und für eine bestimmte Zeit auf die entsprechende Antwort wartet. Im Gegenzug setzt das *Inquiry-Scan-Protokoll* die passiv Rolle um, indem ein Teilnehmer passiv im sogenannten *Inquiry-Scan-Modus* auf mögliche Inquiry-Anfragen lauscht. Wird eine solche empfangen, sendet der Teilnehmer eine entsprechende Inquiry-Antwort, wodurch er für den Initiator sichtbar wird [33]. Diese enthält unter anderem die individuelle Hardware- bzw. MAC-Adresse des jeweiligen Bluetooth-Geräts, womit es sich eindeutig identifizieren lässt. Zudem werden seit der Protokollarchitektur von Bluetooth 2.1 zusätzliche Informationen wie Name, Sendeleistung, unterstützte Dienste oder herstellerspezifische Werte mitgeschickt [24].

All diese Informationen können unbemerkt und ohne großen Aufwand von Dritten gesammelt werden, indem eine Messstation fortlaufend Inquiry-Anfragen an die Umgebung verschickt und die automatisch zurückgelieferten Antworten abspeichert. Durch den Einsatz mehrerer solcher Stationen in einem Gebiet kann theoretisch eine flächendeckende passive Nutzererfassung umge-

setzt werden [118]. Allerdings gelingt dies nur unter der Voraussetzung, dass angefragte Geräte sich in Reichweite und im Inquiry-Scan-Modus befinden und damit für eine Messtation erfassbar sind. Es darf bezweifelt werden, ob dies in der Realität für eine hinreichend große Anzahl an Geräten zutrifft, da moderne mobile Geräte per Default nicht sichtbar sind und somit unentdeckt bleiben, selbst wenn Bluetooth aktiviert ist [7]. Diese Annahme wird im späteren Verlauf der Arbeit auf Basis realer Daten untersucht. Als vielversprechende Alternative bietet sich daher WLAN an, was im Folgenden näher beschrieben wird.

### 2.1.2 WLAN

WLAN ist ähnlich wie Bluetooth ein Standard für drahtlose lokale Netzwerke (engl.: *Wireless Local Area Networks*) und in IEEE 802.11 spezifiziert [92]. Im Vergleich zu WPANs werden hier höhere Datenübertragungsraten auf größere Entfernungen mit mehr Sendeleistung erreicht. So ergeben sich je nach Umgebung, Ausgangsleistung und der verwendeten 802.11 Protokollerweiterung Reichweiten von ca. 35 m innerhalb von Gebäuden bis hin zu 100 m im Außenbereich [1,229]. Für eine stabile Kommunikation definiert der Standard drei unterschiedliche Typen an Datenrahmen (engl.: *Frames*):

- *Kontroll-Frames* für die Mediumszugriffskontrolle, um einen störungsfreien Versand von Nutzdaten vorzubereiten. Darunter zählen typischerweise *RTS*- und *CTS*-Frames zur Vermeidung von Kollisionen sowie *ACK*-Frames zur Empfangsbestätigung bei Datenpaketen.
- *Daten-Frames* zur Übertragung der eigentlichen Nutzdaten, die für höhere Schichten wie bspw. Anwendungen bestimmt sind.
- *Management-Frames* für den Aufbau und die Verwaltung von Verbindungen. Hierzu zählen vor allem die *Beacon*- bzw. *Probe*-Frames, um Netzwerkteilnehmer zu finden sowie *Authentication*- und *Association*-Frames, um sich zu authentifizieren und eine Verbindung einzugehen.

Für eine passive Nutzererfassung kommen erneut nur diejenigen Mechanismen in Frage, die zur Erkennung eines Geräts in Reichweite vorgesehen sind, ohne dass eine Verbindung aufgebaut oder gar bestehen muss. Somit liegt hier der Fokus auf den Management-Frames, da sie diese Aufgabe im Rahmen des IEEE 802.11 Netzwerkerkennungs- und Assoziierungsprozesses übernehmen. Abbildung 2.1 stellt den typischen Verlauf eines solchen Prozesses zwischen einem mobilen Gerät und einem Access Point schematisch dar. Dabei kann die Netzwerksuche aus der Sicht eines mobilen Geräts passiv (1) oder aktiv (2) verlaufen. Bei der passiven Suche (engl.: *passive scan*) hört ein Teilnehmer zunächst auf *Beacon-Frames*, die in regelmäßigen zeitlichen Abständen (ca. alle 100 ms) von einem Access Point auf dem eingestellten Kanal versendet werden, um dessen Präsenz mitzuteilen. Für eine erfolgreiche Erkennung muss

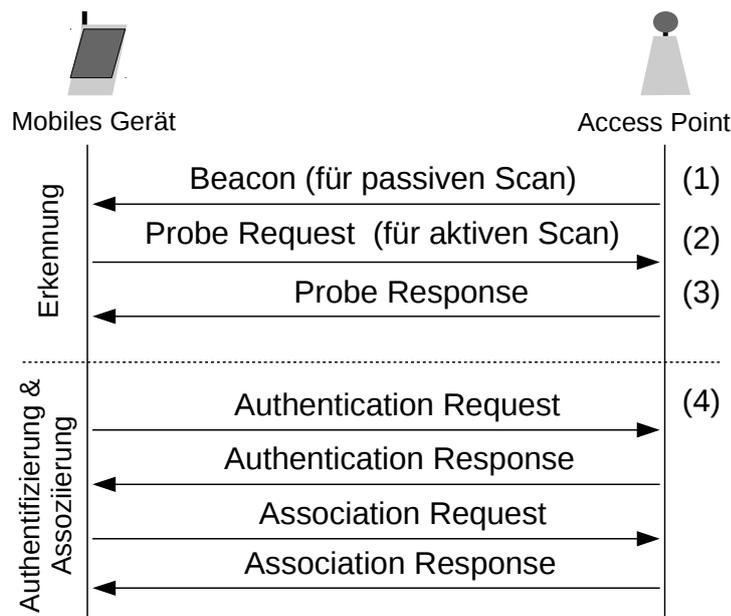


Abbildung 2.1: Typischer Verlauf des IEEE 802.11 Netzwerkerkennungs- und Assoziierungsprozesses.

die Netzwerkschnittstelle des Teilnehmers zum Zeitpunkt des Empfangs auf den Frequenzbereich des entsprechenden Kanals eingestellt sein. Ist dies gerade nicht der Fall, kann sich der Prozess verzögern, wodurch mehr Strom und Zeit verbraucht wird.

Um dies zu vermeiden, nutzen vor allem mobile Geräte die Möglichkeit der aktiven Netzwerksuche (engl.: *active scan*) [122]. Dabei versenden sie sogenannte *Probe Request Frames* und lauschen für eine gewisse Zeit auf entsprechende Antworten, den *Probe Responses* (3). Diese Prozedur wird bei jedem aktiven Scan für alle verfügbaren Kanäle iterativ ausgeführt. Ein Probe Response fungiert dabei ähnlich wie ein Beacon-Frame. Wird ein solcher auf einem Kanal empfangen, kann über die mitgelieferten Informationen ein Verbindungsaufbau initiiert werden (4), wie in Abbildung 2.1 zu sehen ist. Dieser soll hier nicht weiter thematisiert werden, da zur Nutzererfassung wie erwähnt die bloße Erkennung eines Geräts ausreicht. Mobile Geräte bevorzugen im Allgemeinen den aktiven Scan, um schneller Access Points in der Umgebung zu entdecken, mit denen sie sich ggf. automatisch verbinden können. Durch dieses Vorgehen wird eine ubiquitäre Nutzung von WLAN-Infrastrukturen erreicht, die gerade im mobile Bereich für eine schnelle, bequeme und kostengünstige Verbindung ins Internet von großem Vorteil ist.

Ebenso führen mobile Geräte mit aktivierter WLAN-Schnittstelle selbstständig in unregelmäßigen Abständen aktive Scans durch, ohne dass der Nutzer dies initiieren muss. Das Scan-Intervall hängt dabei von diversen Faktoren wie der verwendeten Hardware oder der Nutzeraktivität ab. Empirische Unter-

suchungen mit mehreren mobilen Geräten legen nahe, dass unabhängig vom aktuellen Verbindungsstatus im Durchschnitt mindestens alle zwei Minuten ein aktiver Scan durchgeführt wird [26]. Zudem konnte eine durchschnittliche Zeitspanne von 45 Sekunden zwischen zwei aufeinanderfolgenden Anfragen ermittelt werden [70]. Eigene Untersuchungen mit einem Android-Smartphone und einem iPhone konnten diese Ergebnisse bestätigen, wobei in einer zehnstündigen Testphase die längste Zeitspanne zwischen zwei Anfragen ca. fünf Minuten betrug [186]. Eine Anfrage besteht dabei aus einer Abfolge mehrerer Probe Requests, die engl. *Bursts* genannt werden.

Selbst wenn ein Gerät bereits mit einem Access Point verbunden ist, sendet es zur Verbesserung oder dem Erhalt einer bestehenden Verbindung weiter Probe Requests aus. Zudem sieht der Standard einen gezielten aktiven Scan vor, wenn nach dem Erhalt eines Beacon-Frames weitere Informationen vom entsprechenden Access Point benötigt werden. Probe Requests selbst enthalten neben der gerätespezifischen MAC-Adresse des Senders und Empfängers weitere Managementinformationen wie bspw. Angaben zu unterstützten Übertragungsraten und den Namen des Zielnetzwerks, die SSID (engl.: *Service Set Identifier*). Falls das SSID-Feld bei der Übertragung leer ist, wird der entsprechende Probe Request als Broadcast verstanden. Ein solcher adressiert damit alle Access Points, die sich in Reichweite befinden und dementsprechend antworten können. Ist das Feld hingegen mit einem Namen belegt, spricht man von einem gerichteten Probe Request, auf den nur derjenige Access Point antworten sollte, der ein Netzwerk mit gleichem Namen anbietet. Dieser Mechanismus ist das Resultat von versteckten Netzwerken (engl.: *hidden networks*), die keine SSID in ihren Beacons mitliefern und somit von anderen Teilnehmern nur mittels gerichteter Anfragen entdeckt und angesprochen werden können. In der Realität versenden viele mobile Geräte auch dann gerichtete Probe-Anfragen mit eindeutigem Namen, wenn sie die entsprechende SSID in ihrer Liste der präferierten Netzwerke, der sogenannten *Preferred Network List* (PNL), gespeichert haben. In Kombination mit den anderen Informationen, die aus regelmäßigen aktiven Scans gewonnen werden können wie bspw. der MAC-Adresse des Geräts, stellt dieser Vorgang ein großes Risiko für die Privatsphäre der jeweiligen Nutzer dar [128].

Probe Requests werden zudem standardmäßig unverschlüsselt übertragen und können daher auf einfache Weise abgehört und interpretiert werden. Ein beliebiges Gerät in Reichweite, ausgestattet mit einer WLAN-Schnittstelle, die sich im sogenannten Monitor-Modus befindet, ist dafür ausreichend. Mit dem Einsatz mehrerer solcher WLAN-Monitore lassen sich ganze Personengruppen in einem bestimmten Gebiet erfassen, ohne dass die Zustimmung der jeweiligen Nutzer verlangt wird. Um aus diesen Daten die notwendigen Ortsinformationen zur Analyse von Nutzerströmen ableiten zu können, bedarf es geeigneter Methoden zur passiven Lokalisierung. Diese werden im Folgenden vorgestellt.

## 2.2 Methoden zur passiven Lokalisierung

Generell bieten sich nach Hightower und Boriello [85] drei prinzipielle Methoden zur Positionsbestimmung an: Die Nahbereichserkennung (engl.: *Proximity Detection*), die Triangulation und die Szenenanalyse, auch Mustererkennung genannt. Für eine passive Lokalisierung kommen nur Infrastruktur-basierte Lösungen in Frage, da sowohl die Initiierung als auch die Durchführung der Positionsbestimmung rein durch die vorhandene Infrastruktur bewerkstelligt werden muss. Daher werden im Folgenden die drei genannten Methoden in Hinblick auf eine Infrastruktur-basierte Umsetzung betrachtet, wobei der Fokus auf den empfangenen Signalen von mobilen Geräten liegt.

### 2.2.1 Nahbereichserkennung

Bei der Nahbereichserkennung geht es lediglich um die grobe Erfassung der räumlichen Nähe eines mobilen Geräts, das sich zum Zeitpunkt seiner Entdeckung im Empfangsbereich der Messstation aufhalten muss. Die Position der Station ist dabei genau bekannt, was der prinzipiellen Ausgangslage eines Infrastruktur-basierten Tracking-Systems entspricht. Demnach ist die Nahbereichserkennung sehr leicht umzusetzen. Allerdings können hier, je nach Sendeleistung des Geräts, nur grobe Positionsangaben über den wahren Aufenthalt des mobilen Teilnehmers ermittelt werden. Dabei gilt, je größer die maximale Reichweite einer Signalübertragung, desto größer der entsprechende Signalaradius  $r$  und damit auch die Kreisfläche, auf der sich das jeweilige Gerät um die Messstation herum befinden kann. Durch Verringerung dieser Fläche lässt sich die Genauigkeit des Verfahrens steigern. Dies kann zum einen durch die Verwendung von Technologien erreicht werden, die Signale mit niedrigerer Reichweite einsetzen, wie bspw. Bluetooth. Zum anderen kann die Messstation ihren Radius selbst begrenzen, indem sie nur Geräte erfasst, die eine bestimmte Empfangssignalstärke überschreiten. Als weitere Möglichkeit kann, wie in Abbildung 2.2 dargestellt, ein Teilnehmer von mehreren Messstationen gleichzeitig beobachtet werden. Durch die Schnittfläche der einzelnen Kreise entsteht ein eingeschränkter Bereich, in dem sich das Gerät befinden muss. Dieser kann, je nach Dichte der Stationen, sehr klein werden, wodurch auch höhere Genauigkeiten im Meter- und Zentimeterbereich erreicht werden [217].

### 2.2.2 Triangulation

Bei der Triangulation erfolgt die Positionsbestimmung über gemessene oder errechnete geometrische Dreieckseigenschaften wie Winkel oder Entfernung. Bei einer genaueren Klassifizierung wird noch zwischen der bloßen *Lateralation* und der reinen *Angulation* unterschieden, bei der entweder die Entfernung oder der Winkel eines eintreffenden Signals bestimmt wird, um auf den Ort des Senders zu schließen [85, 130]. Beide Verfahren sollen im Folgenden gesondert betrachtet werden.

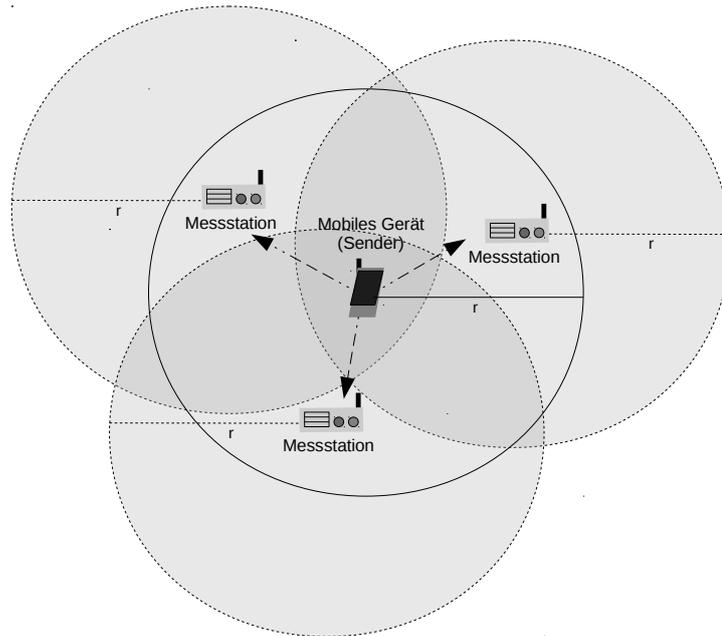


Abbildung 2.2: Beispiel der Nahbereichserkennung unter Einsatz mehrerer Messstationen. Die Schnittfläche der Kreise stellt den eingeschränkten möglichen Aufenthaltsbereich des Geräts dar.

### Lateration

Bei der Lateration werden mindestens drei Stationen benötigt, die auf Basis der errechneten Entfernungen zum Sender einen eindeutigen zweidimensionalen Schnittpunkt bestimmen, der die exakte Position des Senders in der Ebene widerspiegelt. Für eine dreidimensionale Lokalisierung sind vier linear unabhängige Stationen nötig. Allerdings sind Signalmessungen in der Realität fehlerbehaftet, weshalb kein eindeutiger Schnittpunkt, sondern vielmehr eine Fläche bzw. ein geometrischer Körper als möglicher Aufenthaltsort bestimmt werden kann. Dies wird in Abbildung 2.3 durch den schraffierten Bereich beispielhaft dargestellt. Mittels Ausgleichsrechnung lässt sich eine solche Unsicherheit auf die wahrscheinlichste Position reduzieren [103, 217].

Um die Entfernung zwischen Sender und Empfänger auf Basis von Funksignalen bestimmen zu können, bieten sich prinzipiell zwei Möglichkeiten an: Entweder wird die Laufzeit oder der Energieverlust des Signals gemessen. Bei Laufzeit-basierten Verfahren wird die zurückgelegte Distanz des Signals  $d$  über die einfache Formel  $d = v \cdot t$  berechnet. Hierzu müssen die Ausbreitungsgeschwindigkeit  $v$  und die Laufzeit  $t$  des Signals bekannt sein. Bluetooth- und WLAN-Signale zählen zu Radiowellen, die sich im freien Raum mit Lichtgeschwindigkeit  $c$  fortbewegen. Daher gilt hier für die Ausbreitungsgeschwindigkeit  $v = c \approx 3,0 \cdot 10^8 \frac{m}{s}$ . Bei dieser hohen Geschwindigkeit benötigt man eine exakte Laufzeitmessung im Nanosekundenbereich. Weicht dabei der gemessene Wert für  $t$  nur um eine Nanosekunde von der echten Signallaufzeit ab, führt

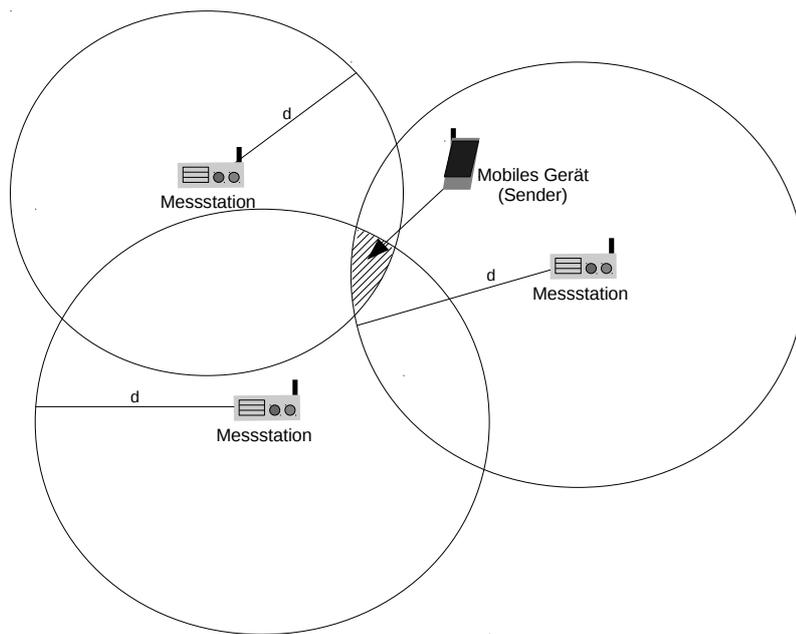


Abbildung 2.3: Darstellung der Lateration zur Positionsbestimmung. Durch ungenaue Entfernungsmessungen kann häufig kein eindeutiger Schnittpunkt errechnet werden.

dies bereits zu einem Distanzfehler von 0,3 Metern. Neben ungenauen oder nicht synchronisierten Uhren führt auch die Mehrwegeausbreitung (engl.: *multipath propagation*) zu Abweichungen bei der Zeitmessung. Ein solcher Effekt tritt vor allem innerhalb von Gebäuden auf. Dabei werden Signale durch Hindernisse oder Wände von ihrem direkten Weg abgelenkt und benötigen somit mehr Zeit als angenommen, was sich negativ auf die Distanzberechnung auswirkt [68]. Generell werden in der Literatur drei unterschiedliche Arten der Laufzeitmessung diskutiert:

- **Time-Difference-of-Arrival (TDOA):** Beim TDOA-Verfahren wird der Zeitpunkt eines ankommenden Signals an mehreren synchronisierten Stationen erfasst. Durch die bekannten Positionen der einzelnen Stationen und der errechneten Differenz der Ankunftszeiten kann über eine *hyperbolische* Lateration der Ort des Senders ermittelt werden. Dafür müssen mindestens zwei Hyperbeln bestimmt werden, die sich im besten Fall in einem Punkt, nämlich der gesuchten Position, schneiden. Mathematisch muss ein nicht-lineares und überbestimmtes Gleichungssystem gelöst werden. Zudem sind die Eingabewerte hier meist fehlerbehaftet. Ansätze zu diesem nicht trivialen Problem werden bspw. von Gustafsson und Gunnarsson [77] diskutiert. Wichtig bei TDOA-Verfahren ist die exakte Synchronisation der einzelnen Messstationen. Zum einen müssen sie in der selben Frequenz auf Signale hören und zum anderen müssen ihre Uhren exakt aufeinander abgestimmt sein. Bei Radiowellen ist dazu wieder Nanosekundengenauigkeit erforderlich. Außerdem bedarf

es einer zentralen Instanz, welche die aufgenommenen Zeitstempel vergleicht und die hyperbolische Lateration durchführt. Aufgrund dieser Schwierigkeiten gibt es vergleichsweise wenige Arbeiten, die ein TDOA-Verfahren für die Bluetooth- bzw. WLAN-basierte Ortung verwenden: Li et al. [126] untersuchen bspw. einen Ansatz zur Vermeidung der notwendigen Synchronisation der Empfänger durch die Messung der Verzögerung von Empfangszeiten. Yamasaki et al. [233] hingegen evaluieren verschiedene Ansätze zur Synchronisation von Access Points und zur Vermeidung von Multipath-Effekten. Dabei erreichen die Autoren mit ihrem TDOA-basierten System Genauigkeiten von bis zu 2,4 Metern. Oft wird TDOA auch zusammen mit anderen Techniken wie FDOA (engl.: *Frequency-Difference-of-Arrival*) als hybrides Verfahren zur Lokalisierung von Objekten im Außenbereich verwendet [146].

- **Time-of-Arrival (TOA)**: Dies ist der klassische Ansatz, bei dem die genaue Zeit erfasst wird, die ein Signal auf seinem direkten Weg zwischen Sender und Empfänger benötigt. Allerdings treten dabei erhebliche Schwierigkeiten auf, die ein TOA-Verfahren für eine passive Nutzerverfolgung praktisch unmöglich machen. Zum einen wird wieder eine nanosekundengenaue Synchronisation von Sender und Empfänger vorausgesetzt, die bei relativ günstiger Hardware von mobilen Geräten aber nicht erreicht wird. Zum anderen braucht man die exakten Sende- und Empfangszeitpunkte, um die Differenz zu bilden und die Laufzeit bestimmen zu können. Bei einer passiven Lokalisierung besitzt eine Messstation jedoch keine Kenntnis darüber, wann genau ein Signal von einem mobilen Gerät ausgesendet wurde. Außerdem kann sowohl der Sende- als auch der Empfangszeitpunkt auf Software-Ebene nicht exakt bestimmt werden, da bereits die hardwareseitige Verarbeitung des Signals eine nicht-deterministische Zeit in Anspruch nimmt [177]. Dem Autor der vorliegenden Arbeit sind demnach keine Publikationen bekannt, die ein reines TOA-Verfahren zur Nutzerverfolgung einsetzen.
- **Roundtrip-Time-of-Flight (RTOF)**: Um die angesprochenen Probleme teilweise zu umgehen, wird beim RTOF-Verfahren ein *Ping-Pong*-Ansatz realisiert. Dabei wird die Laufzeit gemessen, die ein Signal vom Sender zum Empfänger und wieder zurück benötigt. Man spricht hier deshalb von der *Round Trip Time* (RTT), mit deren Hilfe sich die Distanz  $d$  zwischen den beiden Teilnehmern wie folgt berechnen lässt:

$$d = \frac{\text{RTT} - t_{\text{Proc}}}{2} c$$

Dabei stellt  $t_{\text{Proc}}$  die Prozesszeit dar, die bei der Verarbeitung des Signals auf Empfängerseite benötigt wird, bis das Signal zurückgeschickt wird. Da die Zeitmessung komplett vom eigentlichen Sender (im vorliegenden Fall ist das die Infrastruktur) übernommen wird, benötigt das RTOF-Verfahren keine Zeitsynchronisation zwischen den beteiligten Geräten. Allerdings muss eine Station über eine hinreichend genaue Uhr im Nanosekundenbereich verfü-

gen und die exakte Prozesszeit  $t_{\text{Proc}}$  kennen. Diese ist aber in Anbetracht der unterschiedlichen mobilen Geräte, die aktuell im Umlauf sind, nicht einheitlich zu bestimmen. Weiterhin besteht das Problem die exakten Sende- und Empfangszeitpunkte zu ermitteln, was ohne Spezialhardware problematisch ist [40]. Bei der Umsetzung einer passiven Lokalisierung kommt erschwerend hinzu, dass eine zusammenhängende, bilaterale Kommunikation nur selten möglich ist. Während Endgerät-zentrische RTOF-Verfahren eine atomare *DATA-ACK*-Kommunikation mit dem Access Point realisieren können [41, 67, 177], ist dies auf umgekehrten Weg nicht möglich, da mobile Geräte in der Regel nicht auf WLAN-Pakete von Netzen reagieren, mit denen sie nicht verbunden sind. So zeigen eigene Experimente, dass Smartphones weder auf eingeschleuste RTS-, noch auf Daten-Frames gemäß dem IEEE 802.11 Standard mit den entsprechenden CTS- bzw. ACK-Frames antworten. Auch Mussa und Eriksson [145] konnten durch gezielt versendete RTS-Frames nur wenige Smartphones aufspüren. Daher kann in diesem Fall kein *Ping-Pong*-Ansatz realisiert werden.

Aufgrund der genannten Schwierigkeiten sind Laufzeit-basierte Systeme zur passiven Lokalisierung nicht realistisch. Somit kommt für die Lateration nur eine Distanzbestimmung auf Basis der gemessenen Empfangssignalstärke (RSS für engl. *received signal strength*) in Frage. Dabei wird die Tatsache ausgenutzt, dass ein Signal auf dem Weg zum Empfänger kontinuierlich an Energie verliert und damit ein direkter Zusammenhang mit der zurückgelegten Distanz besteht. Dieser lässt sich durch das folgende Pfadverlustmodell beschreiben [154]:

$$P_{\text{rx}}(d) = A - 10n \log_{10}(d) - l \cdot \text{WAF} \quad (2.1)$$

Ist also die Ausgangsleistung  $A$  bekannt, kann der Empfänger über die empfangene Signalstärke  $P_{\text{rx}}$  die zurückgelegte Distanz  $d$  in Abhängigkeit des errechneten Energieverlusts bestimmen. Dazu wird ein Exponent  $n$  für den Pfadverlust benötigt, der auf freier Fläche  $n = 2$  beträgt [204]. Für den Innenbereich kann kein derart generischer Wert angegeben werden, da die Beschaffenheit und Struktur des jeweiligen Gebäudes einen signifikanten Einfluss auf das Modell hat. Der Pfadverlust ist jedoch höher als im Freien, weshalb  $n > 2$  innerhalb von Gebäuden gilt. Empirische Untersuchungen legen dabei einen Wert für  $n \approx 3,5$  nahe [27]. Zusätzlich wird ein Signal, das durch Wände hindurch verläuft, abgeschwächt, was sich durch den *wall attenuation factor* (WAF) im obigen Modell beschreiben lässt. Dieser muss entsprechend der Wandanzahl  $l$ -mal vom natürlichen Pfadverlust subtrahiert werden. Die Modellparameter  $A$ ,  $n$  und WAF können auf Basis einer bekannten Referenzdistanz  $d_0$  für ein gegebenes Szenario empirisch ermittelt werden. Somit lässt sich das Modell in einer beliebigen Umgebung zur Infrastruktur-basierten Lateration anwenden und wird deshalb auch von verschiedenen Tracking-Systemen im Außenbereich [232] und innerhalb von Gebäuden genutzt [234]. Dabei besteht die größte Schwierigkeit darin, mit Signalschwankungen adäquat umzugehen, die vor

allem im Innenbereich auftreten und zu einer ungenaueren Entfernungsmessung führen. Mittels linearer Regression bzw. durch die Methode der kleinsten Quadrate lassen sich solche Probleme minimieren.

### Angulation

Bei der zweidimensionalen Angulation werden zwei Stationen benötigt, die den Empfangswinkel eines eintreffenden Signals messen. Befinden sich die drei Teilnehmer nicht auf einer Geraden, kann auf Basis der Winkel ein eindeutiger Schnittpunkt bestimmt werden, der die gesuchte Position des Geräts widerspiegelt. Da die Winkelmessung in der Praxis wieder fehlerbehaftet ist, werden anstelle von Geraden Kegel angenommen, um die Signalausbreitung zu beschreiben. Statt eines Schnittpunkts entsteht somit eine Fläche, die den möglichen Aufenthaltsort angibt. Dies wird beispielhaft durch den überlappenden Bereich in Abbildung 2.4 dargestellt.

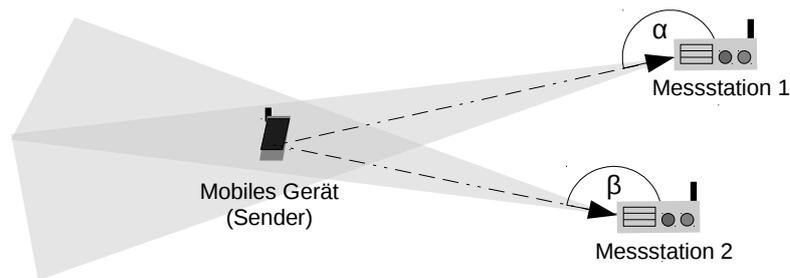


Abbildung 2.4: Darstellung der Angulation zur Positionsbestimmung. Messungenauigkeiten führen zu einem größeren Aufenthaltsbereich.

Liegen mehr als zwei Winkelmessungen vor, kann das überbestimmte Gleichungssystem durch die Methode der kleinsten Quadrate näherungsweise gelöst und somit die wahrscheinlichste Position errechnet werden. Für eine dreidimensionale Lokalisierung wird eine weitere Winkelangabe bzgl. der Höhe benötigt. Die Winkelmessung erfolgt über größere Antennenarrays, die herkömmliche Access Points nicht besitzen. Daher ist eine Angulation nur mit spezieller Hardware realisierbar. Das größte Problem sind jedoch reflektierte Signale, die aus einer anderen als der ursprünglichen Richtung auf die Messstation treffen, wodurch die Positionierung verfälscht wird. Vor allem innerhalb von Gebäuden ist damit eine reine Angulation nicht praktikabel und kann höchstens ergänzend zu anderen Verfahren als hybrider Ansatz verwendet werden [159]. Beispielsweise ist eine Kombination mit der Mustererkennung denkbar, welche nachfolgend beschrieben wird.

### 2.2.3 Mustererkennung

Methoden der Mustererkennung gleichen gemessene Merkmale mit einem entsprechenden Datensatz ab, um Rückschlüsse auf die Position eines Teilnehmers zu ziehen. Anders als bei der RSS-basierten Lateration werden hierbei auftretende Signalschwankungen bewusst in Kauf genommen, um einen eindeutigen Vergleich zwischen gespeicherten und aktuell gemessenen Empfangssignalstärken zu ermöglichen. Bei der Mustererkennung werden zwei Phasen unterschieden.

In einer anfänglichen *Trainings-* bzw. *Offline-*Phase werden mehrdimensionale Vektoren aus gemessenen RSS-Werten gebildet und zusammen mit einer bekannten Ortsinformation als Referenz für die spätere Positionierung in eine Datenbank, der sogenannten *Radiomap*, gespeichert. Die Messungen werden in der Regel vom Infrastruktur-Betreiber für sämtliche Referenzpunkte mehrmalig und unter verschiedenen Blickrichtungen durchgeführt, um möglichst eindeutige Vektoren bilden zu können, die hier *Fingerprints* genannt werden [104]. Zusätzlich gilt, je mehr unterschiedliche Referenzpunkte und damit Einträge in der Radiomap gespeichert sind, desto genauer kann eine anschließende Positionsbestimmung durchgeführt werden. Für größere Bereiche kann die Trainingsphase sehr aufwendig und zeitintensiv sein. Zudem muss sie bei veränderten Bedingungen teilweise oder sogar komplett wiederholt werden. Deshalb beschäftigen sich viele Arbeiten damit, Ansätze zu entwickeln, um den Aufwand zu reduzieren [76, 110, 125]. Eine Möglichkeit, die auch in der vorliegenden Arbeit genutzt wird, ist eine mathematische Berechnung der Radiomap über das in Formel 2.1 beschriebene Pfadverlustmodell. Auf diese Weise lassen sich Fingerprints ohne großen Aufwand für beliebige Referenzpunkte in einem gegebenen Szenario approximieren. Allerdings ist solch ein Modell-basiertes Verfahren in der Regel ungenauer, als wenn empirisch ermittelte Fingerprints verwendet werden [16].

In der späteren *Online-*Phase findet die eigentliche Positionierung des mobilen Geräts auf Basis der erstellten Radiomap statt. Bei einer passiven Lokalisierung muss die Infrastruktur einen RSS-Vektor aus den empfangenen Signalen eines Geräts bilden, der mit den gespeicherten Fingerprints in der Datenbank verglichen werden kann. Dabei müssen wie in der Offline-Phase alle vorhandenen Messstationen berücksichtigt werden, also auch diejenigen, welche im entsprechenden Zeitraum keine zugehörige Signal-Beobachtung machen konnten. Der Grund ist, dass auch die Abwesenheit, also die Nicht-Erreichbarkeit eines mobilen Geräts, wichtige Erkenntnisse über den Aufenthaltsort des Nutzers liefert [145]. Dies erfordert allerdings eine Strategie zum Umgang mit fehlenden RSS-Werten, die in manchen Arbeiten mit der Einführung einer Konstanten gelöst wird [104].

Der RSS-Vektor aus der Online-Phase wird mit allen Einträgen in der Radiomap verglichen. Zur Positionsbestimmung werden schließlich die Referenzpositionen der geeignetsten Kandidaten zurückgeliefert. Zur Festlegung der Eignung eines Kandidaten existieren sowohl deterministische als auch proba-

bilistische Strategien, die in verschiedenen Arbeiten verfolgt werden. Bei einem deterministischen Ansatz werden anhand einer Distanzfunktion die  $k$  nächsten Nachbarn (engl.: *k-Nearest-Neighbors*) bestimmt und der Durchschnitt der entsprechenden, zum Teil gewichteten Referenzpositionen als Aufenthaltsort errechnet [16]. Im Fall von probabilistischen Ansätzen werden hingegen diskrete Wahrscheinlichkeitsverteilungen über vorhandene Einträge bestimmt, um den wahrscheinlichsten Ort bzw. Bereich für das mobile Gerät abzuschätzen [178]. Solche Verteilungen werden häufig im Zusammenhang mit dem Satz von Bayes verwendet. Zudem werden sie in probabilistischen Modellen wie bspw. in *Markov-Ketten* verarbeitet, die aus mehreren ungenauen Beobachtungen ein Wissen über den tatsächlichen Aufenthalt oder den wahren Bewegungspfad herleiten [103]. Auf solche Verfahren wird im späteren Verlauf eingegangen. Prinzipiell eignen sich für die Mustererkennung auch alle gängigen Methoden des maschinellen Lernens, bei denen ein Modell mittels geeigneter Referenzdaten trainiert wird, um konkrete Entscheidungen für die Ortsbestimmung treffen zu können. Bekannte Vertreter sind bspw. *Neuronale Netze* und die *Support Vector Machine* [57].

Die Mustererkennung gilt als eine der verbreitetsten und meist erforschten Methoden zur Lokalisierung mobiler Geräte, da sie auch innerhalb von Gebäuden praktisch einsetzbar ist und höhere Genauigkeiten als andere Verfahren erzielen kann [39]. Im Zusammenhang mit WLAN-Signalen wird häufig auch vom sogenannten *WLAN-Fingerprinting* gesprochen, welches erstmalig im Jahr 2000 von Bahl und Padmanabhan [16] in Form des *RADAR-Systems* vorgestellt wurde. Seitdem haben sich unzählige Arbeiten mit verschiedenen Aspekten zur Verbesserung oder Erweiterung des ursprünglichen Ansatzes beschäftigt, die teilweise bereits erwähnt wurden. Eine komplette Auflistung der relevanten Literatur würde den Rahmen sprengen, weshalb an dieser Stelle auf einschlägige Übersichtsarbeiten (engl.: *Surveys*) verwiesen wird [73, 87, 130].

In Hinblick auf eine passive Lokalisierung scheint die Mustererkennung unter den vorgestellten Methoden am geeignetsten zu sein. Die genannten Gründe hierfür können wie folgt zusammengefasst werden: Da Bluetooth- bzw. WLAN-Signale aufgrund diverser Umwelteinflüsse hohen Schwankungen unterliegen, ist eine genaue Distanzmessung und damit eine Lateration nur schwer umzusetzen. Die Angulation scheidet ebenfalls wegen reflektierter Signale und fehlender Spezialhardware aus. Eine einfache Nahbereichserkennung ist aufgrund der großen Reichweite von WLAN zu ungenau. Es sei angemerkt, dass Signalschwankungen auch beim Fingerprinting zu einer fehlerhaften Positionsbestimmung und zu einer abweichenden Pfadermittlung führen. Die Aufgabe des Tracking-Systems ist es, trotz ungenauer Messungen möglichst akkurate Bewegungspfade zu ermitteln. Hierfür rücken probabilistische Ansätze in den Fokus, da diese die vorhandene Unsicherheit in den Messungen mittels geeigneter Modelle ausgleichen können. Dadurch lässt sich der tatsächliche Bewegungsablauf eines Nutzers genauer abschätzen, wie im Folgenden erläutert wird.

## 2.3 Probabilistische Verfahren zur Pfadermittlung

Probabilistische Verfahren zur Pfadermittlung basieren auf stochastischen Modellen, welche mittels unsicherer Beobachtungen die Position einer Entität abschätzen. Dabei wird der gesuchte Ort über eine Wahrscheinlichkeitsverteilung bestimmt, die angibt, wie wahrscheinlich die Beobachtung für eine Positions-schätzung zutrifft. Die Position wird also nicht wie bei deterministischen Verfahren direkt festgelegt, sondern wahrscheinlichkeitstheoretisch approximiert.

Die folgenden Methoden und Konzepte sind hauptsächlich aus dem Buch *Probabilistic Robotics* von Thrun et al. [198] entnommen, das einen umfassenden Einblick in die Thematik der probabilistischen Positionsbestimmung bietet. Dabei basieren die meisten mathematischen Modelle auf dem Satz von Bayes, der die Berechnung bedingter Wahrscheinlichkeiten beschreibt:

$$P(X|B) = \frac{P(B|X) \cdot P(X)}{P(B)} \quad (2.2)$$

Im vorliegenden Fall sei  $P(X|B)$  die Wahrscheinlichkeit, dass sich ein Gerät an der Position  $X$  befindet, unter der Bedingung, dass eine Beobachtung  $B$  gemacht wurde. Dies entspricht der gesuchten Größe für die Positionsbestimmung. Es wird also entweder der wahrscheinlichste Aufenthaltsort oder die resultierende Wahrscheinlichkeitsverteilung über alle möglichen Aufenthaltsorte auf Basis einer Beobachtung zurückgeliefert. Diese sogenannte *A-posteriori-Verteilung* lässt sich durch den Satz von Bayes zurückführen auf ein Vorwissen über mögliche Positionen, nämlich der *A-priori-Verteilung*  $P(X)$  sowie der bedingten Wahrscheinlichkeit  $P(B|X)$  an der Position  $X$  eine Beobachtung  $B$  gemacht zu haben.  $P(B)$  dient hier als Normalisierungsfaktor mit  $P(B) > 0$ .

Der Satz von Bayes kommt auch beim probabilistischen Fingerprinting zur Anwendung, da  $P(B|X)$  bereits beim Erstellen der Radiomap für jeden Referenzort  $X$  als Wahrscheinlichkeitsdichtefunktion (WDF) der Beobachtungen direkt berechnet und abgespeichert werden kann [135]. Vor allem aber basieren die sogenannten *Bayes-Filter* auf der beschriebenen Beziehung zwischen dem Vorwissen  $P(X)$ , dem Einbeziehen der Messung über  $P(B|X)$  und der daraus berechenbaren A-posteriori-Verteilung. Bayes-Filter gehören zu den gängigsten probabilistischen Algorithmen für die kontinuierliche Positionsbestimmung [103].

### 2.3.1 Bayes-Filter

Die Welt, in der sich ein mobiler Teilnehmer bewegt, lässt sich auch als dynamisches System auffassen, welches sich zu jedem Zeitpunkt  $t$  in einem bestimmten Zustand  $x_t$  befindet. Dabei kann dieser Zustand die Position, aber auch die aktuelle Geschwindigkeit, die Bewegungsrichtung oder andere Eigenschaften des

dynamischen Ziels beschreiben. Durch Beobachtungen in Form von Messungen lassen sich Zustandsänderungen an diskreten Zeitpunkten  $k$  erfassen, die in einen kontinuierlichen stochastischen Prozess eingebunden werden. Hängt die zukünftige Entwicklung dieses Prozesses zu einem Zeitpunkt  $t$  rein vom aktuell bekannten Zustand  $x_t$  ab, so wird die Markov-Bedingung erfüllt und der zeitliche Prozess als Markov-Kette (erster Ordnung) bezeichnet. Da Zustandsänderungen durch die fortlaufende Bewegung eines Nutzers jedoch auch zwischen zwei Beobachtungen kontinuierlich hervorgerufen werden können, benötigt man für den Prozess ein Vorhersage- bzw. in diesem Fall ein Bewegungsmodell, das die Wahrscheinlichkeit  $p(x_k|x_{k-1})$  eines Übergangs zwischen den zwei Folgezuständen  $x_{k-1}$  und  $x_k$  angibt.

Bayes-Filter kombinieren ein solches Vorhersagemodell mit einem Messmodell, welches eine unsichere Messung  $z_k$  zum diskreten Zeitpunkt  $k$  hinsichtlich eines vermuteten Zustands  $x_k$  durch die bedingte Wahrscheinlichkeit  $p(z_k|x_k)$  bewertet. Das Filter-Problem besteht nun darin, die A-Posteriori-Wahrscheinlichkeitsverteilung über den vermuteten Zustand (engl.: *belief*) mit  $bel(x_k) = p(x_k|z_{1:k})$  unter Berücksichtigung aller bisherigen Messungen  $z_{1:k}$  zu bestimmen. Dabei kann  $bel(x_k)$  über den Satz von Bayes mittels eines Vorwissens  $\overline{bel}(x_k)$  und dem Messmodell  $p(z_k|x_k)$  errechnet werden, wobei  $\overline{bel}(x_k) = p(x_k|z_{1:k-1})$  die A-Priori-Verteilung über den Zustand ohne Einbezug der aktuellen Messung  $z_k$  beschreibt:

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{k-1})} \quad (2.3)$$

$\overline{bel}(x_k)$  wird deshalb auch als Vorhersage bezeichnet, womit sich der Zusammenhang aus dem Satz von Bayes verkürzt darstellen lässt als:

$$bel(x_k) = \eta p(z_k|x_k) \overline{bel}(x_k) \quad (2.4)$$

Dabei beschreibt  $\eta = p(z_k|z_{k-1})^{-1}$  den Normalisierungsfaktor der Messung auf Basis der vergangenen Messungen. Der Filter-Algorithmus folgt demnach einem Vorhersage- und einem Korrektur- bzw. Update-Schritt, in welchem die Vorhersage mithilfe der Messung und dem Messmodell korrigiert wird (siehe Gleichung 2.4). Zur Berechnung der aktuellen Vorhersage  $\overline{bel}(x_k)$  wird das Vorhersagemodell und der Einfluss der bisherigen Messungen auf den Vorzustand  $x_{k-1}$  mittels der Chapman-Kolmogorov Gleichung wie folgt berücksichtigt:

$$\overline{bel}(x_k) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1} \quad (2.5)$$

Durch Annahme der Markov-Bedingung hängt der aktuelle Zustand  $x_k$  jedoch nur vom Vorzustand  $x_{k-1}$  ab und somit gilt:

$$\overline{bel}(x_k) = \int p(x_k|x_{k-1})bel(x_{k-1})dx_{k-1} \quad (2.6)$$

Das grundlegende Konzept eines Bayes-Filter ist also rekursiv, in welchem die A-Posteriori-Verteilung  $bel(x_k)$  über den Zustand  $x_k$  zum diskreten Zeitpunkt  $k$  auf Basis der zuvor ermittelten Verteilung  $bel(x_{k-1})$  berechnet wird. Dabei setzt sich ein kompletter Filter-Schritt aus der Vorhersage (s. Gleichung 2.6) und der Korrektur (s. Gleichung 2.4) zusammen. Nach Durchlaufen aller diskreten Zustände wird schließlich eine Abschätzung  $bel(x_k)$  zurückgeliefert.

Dieses Filter-Prinzip lässt sich auf das Problem der Standortverfolgung übertragen. Dabei sei  $x_k$  die Position eines mobilen Geräts zu einem diskreten Zeitpunkt  $k$ , an dem eine Messung  $z_k$  vorliegt. Die Messung kann bspw. eine ungenaue Positionsangabe oder auch der bloße RSS-Vektor sein, der von der Infrastruktur über die zugehörigen Signalbeobachtungen zum Zeitpunkt  $k$  aufgenommen wurde. Für die Vorhersage bis zur nächsten Beobachtung wird ein möglichst realistisches Modell der menschlichen Bewegung benötigt. Da gerade Probe Requests Bursts in größeren zeitlichen Abständen ausgesendet werden, kommt diesem Schritt eine großen Bedeutung zu. Ein umfassender Überblick über verschiedene Bewegungsmodelle wird bspw. in [18] gegeben.

Durch die Verknüpfung der Messung mit dem Wissen über menschliche Bewegungen lassen sich auch unregelmäßige und verrauschte Signalbeobachtungen für eine genaue kontinuierliche Standortverfolgung verwenden. Hierfür werden zwei spezielle Ausprägungen des Bayes-Filters genutzt, nämlich der Kalman- und der Partikel-Filter [103]. Da der Kalman-Filter jedoch Normalverteilung und Linearität für sein Vorhersage- und Messmodell voraussetzt, ist dieser zum Tracken von Personen in der Realität weniger geeignet. Es gibt zwar auch Erweiterungen, wie den *Extended Kalman-Filter*, welcher die Linearität durch die Verwendung von Taylor-Reihen auf nicht-lineare Funktionen herstellt, allerdings geht dadurch die bei klassischen Kalman-Filtern bewiesene Optimalität verloren [198]. Zudem wird weiterhin eine vorhandene Gauß-Verteilung der Zufallsgröße vorausgesetzt, was die Genauigkeit der Lokalisierung in realen Szenarien verschlechtern kann [78]. Um solche Probleme zu vermeiden, bietet sich der Partikel-Filter an, der auf diskretisierte Verteilungen setzt und somit flexibler einsetzbar ist.

## Partikel-Filter

Partikel-Filter stellen eine parameterfreie Variante des rekursiven Bayes-Filters dar. Das bedeutet, dass die zugrunde liegende Wahrscheinlichkeitsverteilung nicht a-priori festgelegt ist, sondern durch Ziehung einer Menge von  $N$  zufälligen Stichproben aus einer Verteilung repräsentiert wird. Dadurch wird zwar die wahre Verteilung häufig nur angenähert, jedoch bildet diese ein breiteres Spektrum ab (als bspw. eine Gauß-Verteilung) und kann somit auch nicht-lineare Zustandsübergänge modellieren. Für eine ausreichend gute Annäherung darf  $N$  nicht zu klein gewählt werden.

Die Stichproben, welche die A-Posteriori-Verteilung  $bel(x_k)$  repräsentieren, werden Partikel genannt und sind definiert als  $X_k := x_k^{[1]}, x_k^{[2]}, \dots, x_k^{[N]}$ . Jeder

**Algorithmus 1** Basisalgorithmus des Partikel-Filters [198]

---

```

1: procedure PARTICLE_FILTER( $X_{k-1}, z_k$ )
2:    $\bar{X}_k = X_k = \emptyset$ 
3:   for  $i \leftarrow 1, N$  do
4:     sample  $x_k^{[i]} \sim p(x_k | x_{k-1}^{[i]})$ 
5:      $w_k^{[i]} = p(z_k | x_k^{[i]})$ 
6:      $\bar{X}_k = \bar{X}_k + \langle x_k^{[i]}, w_k^{[i]} \rangle$ 
7:   for  $i \leftarrow 1, N$  do
8:     ziehe  $i$  mit Wahrscheinlichkeit  $\propto w_k^{[i]}$ 
9:     füge  $x_k^{[i]}$  der Menge  $X_k$  hinzu
10:  return  $X_k$ 

```

---

Partikel  $x_k^{[i]}$  mit  $1 \leq i \leq N$  stellt eine konkrete Instanziierung des Zustands zum diskreten Zeitpunkt  $k$  dar und kann somit mehrere Eigenschaften beschreiben. Im vorliegenden Fall sind dies Bewegungseigenschaften, wie aktuelle Position, aber auch Geschwindigkeit und Richtung eines mobilen Teilnehmers. Idealerweise sollte die Existenz eines Partikels zur A-Posteriori-Verteilung des Bayes-Filters proportional sein, also:  $x_k^{[i]} \sim p(x_k | z_{1:k})$ . Das bedeutet als Konsequenz, dass über die Partikeldichte der wahre Zustand abgeschätzt werden kann. Mit anderen Worten, je dichter die Partikel um einen bestimmten Punkt kreisen, desto wahrscheinlicher ist es, dass sich dort auch die gesuchte Person aufhält. Der grundlegende Ablauf eines Partikel-Filters ist in Algorithmus 1 als Pseudocode dargestellt und wird nun näher erläutert.

Als Eingabewerte erhält der rekursive Filter-Algorithmus eine Partikelmenge  $X_{k-1}$ , welche die A-Posteriori-Verteilung des Vorzustands  $bel(x_{k-1})$  repräsentiert, sowie die aktuelle Messung  $z_k$ . Bei der Initialisierung des Filters zum Zeitpunkt  $k = 1$  muss daher eine Anfangsmenge  $\bar{X}_0$  vorhanden sein, die als primäre Eingabe verwendet wird. Diese kann z.B. durch eine Verteilung über einen bekannten Anfangszustand oder einer angenommenen Gleichverteilung in einer Umgebung realisiert werden. Der Algorithmus erzeugt daraufhin eine temporäre Menge  $\bar{X}$ , welche die Vorhersage  $\overline{bel}(x_k)$  beschreibt. Im eigentlichen Vorhersage-Schritt (vgl. Zeile 4) wird auf Grundlage eines Modells  $p(x_k | x_{k-1}^{[i]})$  der Zustandsübergang für jedes Partikel  $i$  berechnet. Konkret bedeutet dies, dass auf Basis eines zugrunde liegenden Bewegungsmodells die letzte Position jedes Partikels um die angenommene Geschwindigkeit und Richtung verändert wird. Anschließend (vgl. Zeile 5) folgt die Berechnung des sogenannten *importance factor*, der die Wahrscheinlichkeit der aktuellen Messung  $z_k$  für den jeweiligen Partikel  $i$  als Gewicht  $w_k^{[i]}$  beschreibt. Dies stellt den klassischen Korrektur-Schritt in Analogie zum Bayes-Filter dar. Dabei muss die Gesamtsumme aller Gewichte 1 betragen, weshalb häufig eine Normalisierung notwendig ist.

Insgesamt repräsentiert die Menge aller gewichteten Partikel bereits die ge-

suchte A-Posteriori-Verteilung  $bel(x_k)$ . Jedoch ist für den weiteren Verlauf wieder eine ungewichtete Menge nötig, die durch den sogenannten *Resampling*-Schritt (vgl. Zeile 8-9) gewonnen wird. Darin werden ebenfalls  $N$  Partikel aus der zuvor berechneten gewichteten Menge  $\overline{X}_k$  mit Zurücklegen gezogen, wobei die Gewichtung eines Partikels als Wahrscheinlichkeit für dessen Ziehung verwendet wird. So entsteht eine neue Partikelmenge  $X_k$  mit gleicher Mächtigkeit aber mit einer neuen Verteilung, die jetzt auch  $bel(x_k)$  approximiert. Zuvor höher gewichtete Partikel sind darin häufiger vertreten, während alte Partikel mit einer niedrigen Gewichtung nicht mehr vorhanden sind. Durch den Resampling-Schritt, der Charles Darwins Idee vom Überleben der Stärkeren nachempfunden ist, werden Partikel letztendlich in einer A-Posteriori-Verteilung gehalten. Dies stellt sicher, dass Berechnungen nur für die interessanten Bereiche durchgeführt werden müssen, im vorliegenden Fall also dort, wo der Nutzer mit hoher Wahrscheinlichkeit vermutet wird. Der Rechenaufwand wird im Vergleich zu Filtern ohne Resampling, den sogenannten *Sequential Importance Sampling* Filtern (SIS), deutlich reduziert, da diese häufig mehr Partikel benötigen. Aus diesem Grund verwenden auch die meisten Arbeiten den eben beschriebenen *Sequential Importance Resampling* Ansatz (SIR), um Personen oder Objekte über die Zeit hinweg zu verfolgen [35, 65, 153, 208].

Anhand der vielen Referenzen wird bereits deutlich, dass Partikel-Filter ein beliebtes und weit verbreitetes Verfahren in Tracking-Systemen darstellen. Durch ihre Flexibilität lassen sie sich zudem in diversen Szenarien einsetzen. So werden sie bspw. bei der Standortverfolgung innerhalb von Gebäuden bevorzugt verwendet, da vorhandene Raumpläne für eine verbesserte Abschätzung herangezogen werden können. Allerdings spielt die benötigte Anzahl an Partikeln für die Genauigkeit und die Skalierbarkeit solcher Systeme eine wesentliche Rolle. Wenn zeitgleich eine große Menge an Nutzern verfolgt werden soll, kann der Rechenaufwand stark ansteigen. Zur Verbesserung der Skalierbarkeit muss also die Partikelmenge möglichst gering gehalten werden. Ein entsprechendes Verfahren, um dieses Ziel bei gleichbleibender Genauigkeit zu erreichen, wird im späteren Verlauf dieser Arbeit entwickelt. Dabei liegt die Idee zugrunde, statt der kompletten Freifläche ein diskretes Zustandsmodell zu verwenden, welches sich für die probabilistische Pfadermittlung einsetzen lässt. Ein bekanntes Beispiel für ein solches Modell soll im Folgenden vorgestellt werden.

### 2.3.2 Hidden Markov Model

Ein *Hidden Markov Model* (HMM) beschreibt ein Modell für die Markov-Kette mit verborgenen Zuständen [22]. Dabei ist der Gesamtzustand des Systems  $x_t$  durch die Markov-Bedingung wieder rein von seinem Vorzustand  $x_{t-1}$  abhängig, wobei der Zustandsübergang mit der Wahrscheinlichkeitsverteilung  $p(x_t|x_{t-1})$  angegeben wird. Im vorliegenden Fall wird mit dem Index  $t$  der zeitlich kontinuierliche Prozess charakterisiert. Es sei jedoch angemerkt, dass die

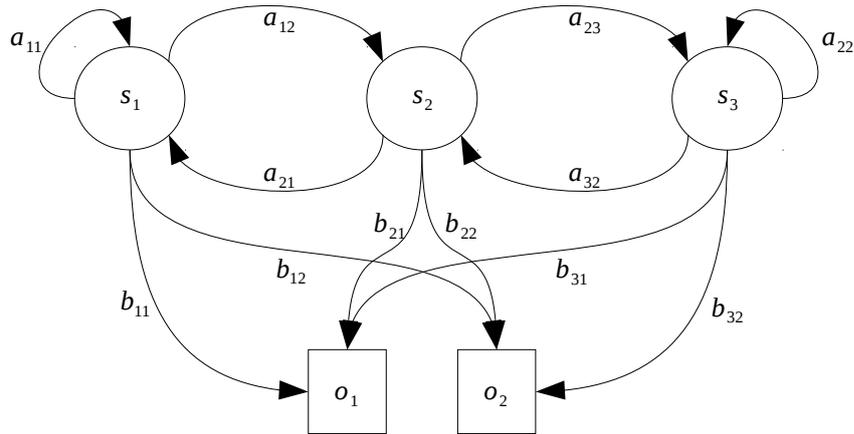


Abbildung 2.5: Beispielhafte Darstellung eines HMMs mit 3 Zuständen, 2 Beobachtungen und den jeweiligen Übergängen.

Übergangswahrscheinlichkeit nicht zwangsläufig von der Zeit abhängen muss. Da die Zustände in einem solchen Modell verborgen und somit nicht beobachtbar sind, muss die Messung  $z_k$  Rückschlüsse auf den aktuellen Zustand  $x_k$  zu einem diskreten Zeitpunkt  $k$  erlauben. Da  $z_k$  als stochastisch abhängig von  $x_k$  angenommen wird, gilt die Messung bzw. die Beobachtung als verrauschte Projektion des jeweiligen Zustands mit  $p(z_k|x_k)$ . Ein HMM kann damit prinzipiell auch als dynamisches bayessches Netz verstanden werden [198].

Formal ist es definiert als das Fünfer-Tupel  $\lambda = (S; O; A; B; \pi)$ , wobei  $S$  die Menge aller  $N$  diskreten Zustände mit  $S = \{s_1, s_2, \dots, s_n\}$  umfasst. Durch die Verteilung über  $S$  kann der Zustand des Systems  $x_k$  zu einem diskreten Zeitpunkt  $k$  abgeschätzt werden.  $O$  hingegen beinhaltet alle möglichen Beobachtungen mit  $O = \{o_1, o_2, \dots, o_m\}$ .  $A \in \mathbb{R}^{n \times n}$  und  $B \in \mathbb{R}^{n \times m}$  definieren die Wahrscheinlichkeitsmatrizen für die Zustandsübergänge und die Beobachtungen. Dabei steht ein Eintrag  $a_{ij} \in A$  für die Wahrscheinlichkeit von einem Zustand  $s_i$  in den Zustand  $s_j$  zu gelangen, während  $b_{iv} \in B$  die Beobachtungswahrscheinlichkeit  $p(o_v|s_i)$  beschreibt. Zur Initialisierung des Modells muss noch eine Anfangs-Verteilung  $\pi \in \mathbb{R}^n$  festgelegt werden, welche die Wahrscheinlichkeit  $\pi_i = p(x_0 = s_i)$  für den Startzustand  $x_0$  angibt. Somit lässt sich ein HMM über  $\lambda$  komplett beschreiben. Zur besseren Veranschaulichung stellt Abbildung 2.5 die Zusammenhänge eines exemplarischen HMMs mit drei diskreten Zuständen ( $s_1$  bis  $s_3$ ) und zwei möglichen Beobachtungen ( $o_1$  und  $o_2$ ) noch einmal graphisch dar.

HMMs werden häufig bei der Mustererkennung eingesetzt. Dabei reichen die Anwendungsszenarien von der Erkennung einer gesprochenen Sprache, über Textanalysen, bis hin zur Aufbereitung von Proteinsequenzen [60]. Aber auch zur Lokalisierung und Pfadermittlung werden diese Modelle bevorzugt verwendet [116, 145]. Hierbei beschreibt  $S$  die Menge der möglichen Aufenthaltsorte,

die für einen gegebenen Bereich (wie bspw. einem Gebäude) definiert werden müssen. Die Wahrscheinlichkeiten für den Zustandsübergang  $A$  können dabei über die theoretische Erreichbarkeit eines anderen Ortes innerhalb der Umgebung bestimmt werden. Im vorliegenden Fall beinhaltet  $O$  die Signalbeobachtungen der mobilen Geräte und  $B$  die Wahrscheinlichkeiten, eine solche Beobachtung an einem bestimmten Ort zu tätigen. Damit lässt sich also ein diskretes Zustandsmodell für die Positionsbestimmung definieren. Für einen kontinuierlichen Prozess, der bspw. durch den vollständigen Bewegungsablauf eines Nutzers beschrieben wird, besteht jedoch weiterhin das Problem, die wahrscheinlichste Sequenz von verborgenen Zuständen über die Zeit zu bestimmen. Dies ist für eine akkurate Pfadermittlung und somit zur Erfassung von Nutzerströmen essentiell und kann mithilfe des Viterbi-Algorithmus gelöst werden, der im Folgenden beschrieben wird.

### 2.3.3 Viterbi-Algorithmus

Der Viterbi-Algorithmus, welcher in [206] vorgestellt wurde, stellt eine optimale Lösung zur Bestimmung der wahrscheinlichsten Sequenz in einem zeitlich diskreten Markov-Prozess mit endlichen Zuständen dar [61]. Konkret bedeutet dies, dass für ein gegebenes HMM  $\lambda$  und einer zeitlich geordneten Menge an Beobachtungen  $o = o_1 o_2 \dots o_T \in O^*$ , die wahrscheinlichste Folge an verborgenen Zuständen  $s^* = s_1^* s_2^* \dots s_T^* \in S^*$  zurückgeliefert wird. Diese Folge wird auch als *Viterbi-Pfad* mit Länge  $T$  bezeichnet. Mathematisch muss folgendes Problem gelöst werden:

$$p(o; s^* | \lambda) = \max_s p(o; s | \lambda) \quad (2.7)$$

Hierbei stellt  $s \in S^T$  alle möglichen Folgen und  $s^* \in S^T$  diejenige Sequenz mit Länge  $T$  dar, die den Wert der Verteilung  $p(s|o; \lambda)$  maximiert. Zur Lösung des Problems wird zunächst die Variable  $\delta_t(i)$  definiert, welche die zum Zeitpunkt  $t$  wahrscheinlichste Sequenz speichert, die im Zustand  $s_i$  aufhört. Durch Induktion erhält man die mathematische Formulierung für die wahrscheinlichste Folgesequenz, die im Zustand  $s_j$  endet:

$$\delta_{t+1}(j) = [\max_s \delta_t(i) a_{ij}] \cdot b_j(o_{t+1}) \quad (2.8)$$

Gemäß dem definierten HMM steht  $a_{ij}$  wieder für die Wahrscheinlichkeit des Zustandsübergangs von  $s_i$  nach  $s_j$  und  $b_j(o_{t+1})$  bildet die Beobachtungswahrscheinlichkeit  $p(o_{t+1}|s_j)$  ab. Es lässt sich also bereits erkennen, dass der Viterbi-Algorithmus rekursiv vorgeht. Dabei werden zur Berechnung des Viterbi-Pfads vier Phasen durchlaufen, die wie folgt definiert sind [166]:

1. **Initialisierung:** Neben der Anfangsbelegung  $\delta_1(i) = \pi_i b_i(o_1)$  mit  $1 \leq i \leq |S|$  wird eine Variable  $\psi_t(j)$  mit  $\psi_t(i) = 0$  initialisiert. Diese speichert für jeden Zeitpunkt  $t$  und jeden Zustand  $s_j$  das Argument, welches die Gleichung 2.8 maximiert, sprich welcher Vorgängerzustand  $s_i$  letztendlich

an der Maximumsbildung beteiligt war [228].

2. **Rekursion:** Hier findet der Rekursionsschritt zur Berechnung der beiden Variablen  $\delta_t(j)$  und  $\psi_t(j)$  über die Zeit  $t$  statt, wobei für  $2 \leq t \leq T$  und  $1 \leq j \leq |S|$  gilt:

$$\delta_t(j) = \max_{1 \leq i \leq |S|} [\delta_{t-1}(i) a_{ij}] b_j(o_t)$$

$$\psi_t(j) = \arg \max_{1 \leq i \leq |S|} [\delta_{t-1}(i) a_{ij}]$$

3. **Terminierung:** Am Ende der Rekursion wird das Ergebnis von Gleichung 2.7 anhand  $\delta_T(i)$  sowie die wahrscheinlichste Sequenz  $s_T^*$  für den letzten Zeitpunkt  $t = T$  bestimmt:

$$p(o; s^* | \lambda) = \max_{1 \leq i \leq |S|} [\delta_T(i)]$$

$$s_T^* = \arg \max_{1 \leq i \leq |S|} [\delta_T(i)]$$

4. **Pfadermittlung:** Mithilfe der  $\psi_t(j)$  Variablen und den berechneten wahrscheinlichsten Sequenzen  $s_t^*$  für jeden Zeitpunkt  $t$  lässt sich nun der wahrscheinlichste Pfadverlauf über die Gesamtzeit mittels *Backtracking* von hinten nach vorne rekursiv bestimmen. Somit gilt für  $t = T - 1, T - 2, \dots, 1$  folgender Zusammenhang für die Pfadermittlung:

$$s_t^* = \psi_{t+1}(s_{t+1}^*)$$

Der Viterbi-Algorithmus ist zwar optimal, hat aber aufgrund der durchgeführten Optimierungen über  $|S| \times |S|$  für  $T$  Schritte eine quadratische Laufzeit von  $O(|S|^2 T)$ . Die Komplexität wird also maßgeblich von der Anzahl der versteckten Zustände bestimmt. Das bedeutet, dass ein Tracking-System möglichst wenige ortsbezogene Zustände definieren sollte, um für eine größere Menschenmenge skalierbar zu bleiben. Dies wiederum kann sich negativ auf die Genauigkeit der Pfadermittlung auswirken, was im späteren Verlauf der Arbeit genauer evaluiert wird.

Es wurden nun die Grundlagen vermittelt, wie kontinuierliche Positionsschätzungen und somit ganze Bewegungsabläufe auf Basis verrauschter Signalbeobachtungen bestimmt werden können. Im folgenden Abschnitt steht die Analyse solcher Bewegungspfade im Fokus. Hierzu werden bekannte Verfahren vorgestellt, die für eine Untersuchung von Nutzerströmen relevant sind.

## 2.4 Analyse von Bewegungspfaden

Der Bewegungspfad eines Nutzers wird in der vorliegenden Arbeit als Sequenz zeitlich aufeinanderfolgender Positionsschätzungen  $x_k$  aufgefasst, die zu diskreten Zeitpunkten  $1 \leq k \leq T$  vorliegen. Eine solche spatiale Zeitreihe wird im

Folgenden als Trajektorie  $\tau$  (engl.: *trajectory*) mit  $\tau = x_1, x_2, \dots, x_T$  bezeichnet. Im Rahmen des Forschungsgebiets *Trajectory Computing*, dessen Bedeutung durch die immense Verbreitung mobiler Geräte und der damit verbundenen Verfügbarkeit von Positionsdaten stark zugenommen hat, werden Methoden zur Verarbeitung und Analyse großer Trajektorien-Datensätze untersucht [244]. Bestehende Analyse-Verfahren lassen sich dabei nach Gudmundsson et al. [74] in fünf thematische Bereiche gliedern, die nachfolgend vorgestellt werden.

### 2.4.1 Ähnlichkeit

Die Ähnlichkeit zwischen zwei oder mehreren Trajektorien lässt sich anhand einer Funktion bestimmen, die einen Skalar als Ähnlichkeitswert zurückliefern kann. Dabei gilt in der Regel, je höher dieser Wert, desto höher auch die Ähnlichkeit der Trajektorien, was der Inversen einer Distanzfunktion entspricht. Dadurch lässt sich der Abstand zwischen zwei Pfaden anhand bestimmter Metriken errechnen. Hierzu können bspw. spatiale Kriterien wie Lage, Richtung oder Länge, aber auch temporale oder spatio-temporale Eigenschaften, wie Abtastezeit, Sequenz oder Geschwindigkeit herangezogen werden. Als eines der einfachsten Distanzmaße kann bspw. die  $p$ -norm  $L_p(\vec{x}, \vec{y})$  auf zwei Vektoren  $\vec{x}$  und  $\vec{y}$  angewendet werden [207].  $L_1$  entspricht dabei der Manhattan- und  $L_2$  der Euklidischen Distanz. Allerdings kann die topologische Beziehung zwischen Trajektorien beliebig komplex werden, was solche linearen Distanzmaße nicht vernünftig abbilden können. Daher werden im Bereich des Trajectory Computing meist weiterführende Funktionen zur Bestimmung der Ähnlichkeit verwendet. Darunter zählen bspw. die *Hausdorff*- und die *Fréchet-Distanz* sowie das *Dynamic Time Warping* [74, 220].

#### Hausdorff-Distanz

Die Hausdorff-Distanz beschreibt ein Maß für die Diskrepanz zwischen zwei Punktmenge  $A$  und  $B$  im Raum. Im Fall von Trajektorien werden dabei alle Punkte berücksichtigt, die auf den Pfaden liegen. Bei größerer Überdeckung der beiden Punktmenge ergibt sich ein geringerer Hausdorff-Abstand. Dieser lässt sich auf Basis von  $\delta(a, b)$ , was die Euklidische Distanz zwischen den Punkten  $a \in A$  und  $b \in B$  beschreibt, wie folgt bestimmen [220]:

$$\delta_{\text{HD}}(A, B) = \max\left\{ \sup_{a \in \text{Im}(A)} \inf_{b \in \text{Im}(B)} \delta(a, b), \sup_{b \in \text{Im}(B)} \inf_{a \in \text{Im}(A)} \delta(a, b) \right\}$$

Suprema und Infima sind mitunter schwer zu berechnen, können aber für Trajektorien durch Betrachtung der entscheidenden Punkte auf dem Pfad effizient bestimmt werden. Obwohl die Hausdorff-Distanz in vielen Fällen gute Ergebnisse liefert, kann es beim Vergleichen von Pfaden zu Problemen kommen, da

die Zeit-Domäne vernachlässigt wird. Dies ist exemplarisch in Abbildung 2.6 dargestellt. Dabei besitzen die beiden Punktmengen  $A$  und  $B$  einen geringen Hausdorff-Abstand, obwohl ihre Verläufe als Trajektorien betrachtet sehr unterschiedlich sind [6]. Abhilfe schafft hier die *Fréchet-Distanz*, welche zwar auch die Distanz zwischen Punkten berechnet, aber gleichzeitig deren Reihenfolge mitberücksichtigt.

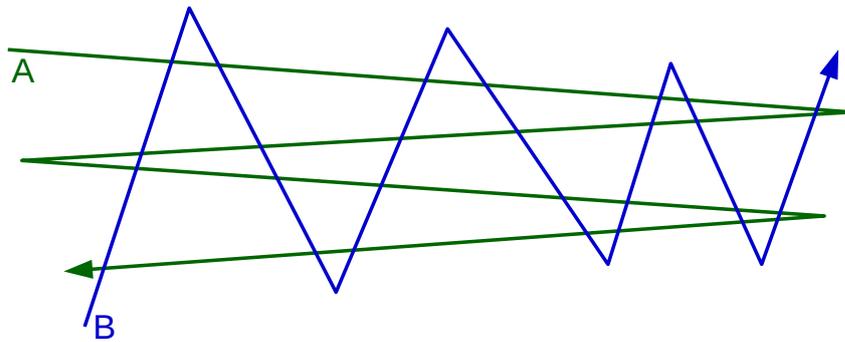


Abbildung 2.6: Probleme der Hausdorff-Distanz beim Trajektorien-Vergleich.

### Fréchet-Distanz

Die Fréchet-Distanz kann informell als die minimale Länge beschrieben werden, die eine Hundeleine haben muss, so dass Hund und Besitzer auf zwei unterschiedlichen Pfaden miteinander kontinuierlich spazieren gehen können. Dabei dürfen sie ihre Geschwindigkeit variieren, aber nicht zurücklaufen. Mathematisch lässt sich dieses Distanzmaß für zwei Trajektorien  $\alpha$  und  $\beta$  wie folgt formulieren [81]:

$$\delta_{\text{FD}}(\alpha, \beta) = \min_{f:[0,1] \rightarrow \alpha, g:[0,1] \rightarrow \beta} \max_t \delta(f(t), g(t))$$

Dabei bezeichnet  $\delta(a, b)$  wieder die Euklidische Distanz, während  $f$  und  $g$  jeweils eine kontinuierliche, monoton steigende Funktion definieren. Durch diese wird eine Parametrisierung der jeweiligen Trajektorie erreicht, welche die Position des Hundes bzw. des Besitzers auf ihrem Pfad für jede Instanz beschreibt. Zur Berechnung besitzt der ursprüngliche Algorithmus eine Komplexität von  $O(pq \log(pq))$ , wobei  $p$  und  $q$  die Anzahl der Kanten der jeweiligen Trajektorien repräsentieren [6]. Mithilfe einer Diskretisierung kann die Laufzeit auf  $O(pq)$  reduziert werden. Diese sogenannte *diskrete* Fréchet-Distanz approximiert die Lösung des ursprünglichen Algorithmus hinreichend genau [54]. Ein Nachteil des Distanzmaßes ist allerdings, dass einzelne Abweichungen zweier sonst recht ähnlicher Trajektorien das Ergebnis stark beeinflussen, wie Abbildung 2.7 exemplarisch veranschaulicht.



Abbildung 2.7: Nachteil der Fréchet-Distanz beim Trajektorien-Vergleich.

### Dynamic Time Warping (DTW)

DTW stellt einen punktweisen Vergleich zwischen zwei Trajektorien  $\alpha$  und  $\beta$  auf Basis einer Kostenmetrik  $\delta$  an. Dabei wird die Zeitkomponente adaptiv normiert, um auch diejenigen Pfade als ähnlich identifizieren zu können, die zwar eine zeitliche Diskrepanz aufweisen, sonst aber strukturell vergleichbar sind. Ein konkretes Beispiel für die Funktionsweise von DTW ist in Abbildung 2.8 gegeben. Wie man sieht, würde hier ein einfaches lineares Distanzmaß wie bspw. die Euklidische Distanz (vgl. Abbildung 2.8a) zu einer geringeren Ähnlichkeit führen, obwohl die beiden Trajektorien relativ gleich sind. Bei DTW hingegen (vgl. Abbildung 2.8b), werden die vorhandenen Messpunkte mehrfach betrachtet. Dies ermöglicht durch Faltung der Zeitachse (engl.: *time warping*) einen genaueren Strukturvergleich der beiden Linien. Formal lässt sich DTW durch Rekursion beschreiben [220]:

$$\delta_{DTW}(a_{1..n}, b_{1..m}) = \delta(a_n, b_m) + \min \begin{cases} \delta_{DTW}(a_{1..n-1}, b_{1..m-1}) \\ \delta_{DTW}(a_{1..n-1}, b_{1..m}) \\ \delta_{DTW}(a_{1..n}, b_{1..m-1}) \end{cases}$$

Dadurch werden letztendlich die minimalen Kosten über alle paarweisen Kosten  $\delta(a_i, b_j)$  mit  $a_i \in \alpha$  und  $b_j \in \beta$  berechnet. Durch diesen paarweisen Punktvergleich liegt die generelle Laufzeit bei  $O(mn)$  [170]. Man beachte, dass DTW die Dreiecksungleichung nicht zwangsläufig erfüllt und somit kein Distanzmaß im eigentlichen Sinn darstellt [220].

### 2.4.2 Segmentierung

Bei der Segmentierung wird eine Trajektorie aufgrund vorhandener Eigenschaften in ihrem Verlauf an entsprechenden charakteristischen Punkten geteilt, so dass letztendlich die Gesamttrajektorie  $\tau$  als Menge von Subtrajektorien  $v \subseteq \tau$  vorliegt. Dies kann Vorteile für Analyseverfahren wie bspw. dem Clustering bringen [121]. Welche Charakteristika zur Segmentierung herangezogen werden, hängt vom verwendeten Algorithmus bzw. der zugrunde liegenden Strategie ab. In der Literatur finden sich hierfür zahlreiche Ansätze [10, 30, 236].

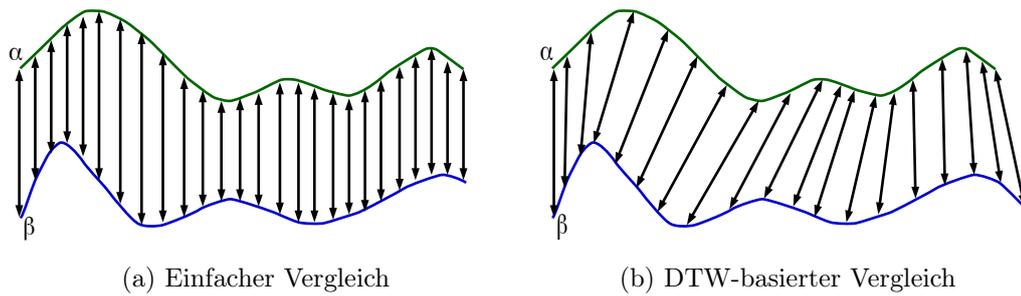


Abbildung 2.8: Veranschaulichung des Trajektorien-Vergleichs mit DTW [151].

Im Folgenden werden zwei bekannte Algorithmen vorgestellt, die im weiteren Verlauf zur Segmentierung von Trajektorien verwendet werden.

### Douglas-Peucker-Algorithmus

Der Douglas-Peucker-Algorithmus dient eigentlich zur Kompression und Glättung von Linien [52]. Wie die meisten Algorithmen zur Simplifizierung von Trajektorien versucht er eine optimale Abstimmung zwischen den konkurrierenden Zielen von Präzision und Prägnanz zu erreichen. Dabei soll die Originaltrajektorie  $\tau_o$  mit möglichst wenig Punkten hinreichend gut approximiert werden. Im vorliegenden Fall wird dazu ein  $\varepsilon$ -Parameter eingesetzt, der den Toleranzbereich der Approximation angibt. Der Algorithmus geht dann wie folgt vor: Zunächst wird der Start- und Endpunkt von  $\tau_o$  mit einer direkten Linie verbunden. Diese wird in zwei Segmente geteilt, indem der Punkt  $c \in \tau_o$ , der den längsten senkrechten Abstand mit  $d > \varepsilon$  zur eben erzeugten Linie hat, als Segmentierungspunkt gewählt wird. Der Algorithmus läuft rekursiv für beide Segmente weiter, bis er keinen solchen Segmentierungspunkt mehr findet.

Dabei wirkt sich die Wahl des  $\varepsilon$ -Parameters unmittelbar auf das Ergebnis aus. Wird  $\varepsilon$  sehr klein gewählt, ist die Präzision hoch, sprich  $\tau_o$  wird komplett oder annähernd wiedergegeben. In diesem Fall leidet jedoch die Prägnanz, da nur wenige oder keine Punkte eingespart werden. Wird  $\varepsilon$  hingegen sehr groß gewählt, kann es sein, dass der Algorithmus sofort terminiert und die Originaltrajektorie auf die direkte Linie zwischen Start- und Endpunkt reduziert. Dies entspricht der maximalen Prägnanz, aber häufig auch der geringsten Präzision.

Da der Douglas-Peucker-Algorithmus charakteristische Punkte zur Trennung von Trajektorien bestimmt, eignet sich dieser ebenso für das klassische Problem der Segmentierung. Er besitzt dabei eine obere Laufzeitkomplexität von  $O(n^2)$ , da im schlechtesten Fall alle  $n$  Punkte von  $\tau_o$  auch  $n$  mal überprüft werden müssen [84].

## TRACCLUS-Algorithmus

Der TRACCLUS-Algorithmus beschreibt ein komplettes Framework zur Partitionierung und Gruppierung von geometrischen Linien. Die gefundenen Gruppen lassen sich dabei über repräsentative Trajektorien zusammenfassen [121]. Die Segmentierung wird durch den Partitionierungsschritt realisiert. Dieser ist für die spätere Gruppierung, dem sogenannten *Clustering*, notwendig, da die Zusammenfassung kompletter Trajektorien häufig ein verzerrtes Bild über deren wirkliche Ähnlichkeit liefert. Mittels Partitionierung lassen sich hingegen Liniensegmente bilden, die für ein feingranulareres Clustering herangezogen werden, wodurch Aussagen über Gemeinsamkeiten von Subtrajektorien getroffen werden können. Zur Segmentierung wird das Prinzip der *minimum description length* (MDL) angewendet [72], um wie beim Douglas-Peucker-Algorithmus eine optimale Abstimmung zwischen Präzision und Prägnanz zu erreichen. Dabei wird gemäß der MDL eine Kostenabwägung getroffen, so dass die Summe der Hypothesenlänge und der Länge der durch diese Hypothese komprimierten Daten minimiert wird. Konkret bedeutet dies, dass der TRACCLUS-Algorithmus die Originaltrajektorie  $\tau_o$  an denjenigen Punkten segmentiert, an denen es gemäß Kostenmetrik teurer wird  $\tau_o$  beizubehalten, als zu trennen. Die Kosten bestehen im vorliegenden Fall aus einer Kombination geometrischer Distanzen, wobei die logarithmischen Summen aus Lotdistanz  $d_{\perp}$  und Winkeldistanz  $d_{\Theta}$  genutzt werden. Die gefundenen Segmentierungspunkte werden als sogenannte *charakteristische Punkte*  $p_c$  bezeichnet und bilden die Start- bzw. Endpunkte der resultierenden Liniensegmente. Diese werden dann für den Clustering-Schritt herangezogen, welcher mitunter im folgenden Abschnitt beschrieben wird.

### 2.4.3 Clustering

Bei der Cluster-Analyse (kurz: *Clustering*) wird über ein Ähnlichkeitsmaß nach zusammenhängenden Gruppen in vorhandenen Datenbeständen gesucht. Dabei sollen Daten, die demselben Cluster zugeteilt werden, möglichst ähnlich zueinander sein, während Elemente aus unterschiedlichen Clustern möglichst verschieden sein sollten [74]. Die Ergebnisbewertung erfolgt daher meist subjektiv und verlangt ein Vorwissen über die Datengrundlage. Es gibt allerdings auch Verfahren, mit der sich die Cluster-Tendenzen in einem Datensatz objektiv beschreiben lassen. So kann bspw. mittels der *Hopkins-Statistik* [19] ein numerischer Wert  $H$  angegeben werden, der sich über den Vergleich der nächsten Nachbar-Distanzen berechnen lässt. Dabei gilt:  $0 \leq H \leq 1$ , wobei  $H = 0$  die schwächste und  $H = 1$  die stärkste Cluster-Tendenz repräsentiert.

Clustering wird im Bereich des *Data Mining* angesiedelt und ist in der Literatur weit verbreitet. Einen umfassenden Überblick über bestehende Verfahren für allgemeine Datensätze liefert beispielsweise das Review von Jain et al. [95]. Zudem wurden Clustering-Algorithmen speziell auch für den Bereich des Trajectory Computing intensiv erforscht [64, 94, 172]. Nach Han et

al. [80] lassen sich die vorhandenen Ansätze unter anderem in partitionierende und hierarchische sowie Dichte-, Gitter- und Modell-basierte Methoden einteilen. Einer der wohl bekanntesten Clustering-Algorithmen ist *k-means*, der ein partitionierendes Verfahren beschreibt und einen gegebenen Datensatz in  $k$  Partitionen unterteilt. Zur Gruppierung von Trajektorien, die als geometrische Linien vorliegen, eignen sich insbesondere Dichte-basierte Verfahren, welche im Folgenden näher erläutert werden.

### Dichte-basiertes Clustering

Beim Dichte-basierten Clustering wird die Gruppierung des Datensatzes auf Basis der vorliegenden Dichte der Objekte in einem  $n$ -dimensionalen Raum vorgenommen. Zur Bestimmung der Dichte und der Dichte-verbundenen Regionen wird ein Distanzmaß und häufig auch ein Toleranzwert benötigt. Zweidimensionale Linien werden bspw. dann als Gruppe zusammengefasst, wenn sie in der Ebene eine Dichte erreichen, die innerhalb des definierten Toleranzbereichs liegt. Die Cluster werden durch weniger dichte Regionen voneinander abgegrenzt. Dies entspricht somit am ehesten der augenscheinlichen Zusammengehörigkeit menschlicher Bewegungspfade.

Ein sehr bekannter und vielseitig angewandter Algorithmus in diesem Bereich ist *DBSCAN* [56], der Dichte-verbundene Punkte als Cluster bestimmt und diese von verrauschten Datenobjekten trennt. Dazu definiert der Algorithmus zunächst drei verschiedene Arten von Punkten, nämlich Kern- bzw. Dichte-Punkte, Dichte-erreichbare Punkte und Rauschpunkte. Als Eingabeparameter wird neben dem Datensatz ein Toleranzwert  $\varepsilon$  und die Angabe einer Mindestanzahl an Punkten *minPts* verlangt. Durch diese lassen sich die genannten Punktarten wie folgt bestimmen:

- $\varepsilon$ -Nachbarschaft: Ein Punkt  $p_1$  befinde sich in der  $\varepsilon$ -Nachbarschaft zu Punkt  $p_2$ , wenn für den Abstand  $\delta$  gilt:  $\delta(p_1, p_2) < \varepsilon$
- Kernpunkt: Ein Punkt ist dann ein Kernpunkt, wenn innerhalb seiner  $\varepsilon$ -Nachbarschaft mindestens *minPts* andere Punkte liegen.
- Dichte-erreichbarer Punkt: Ein Punkt ist Dichte-erreichbar, wenn er innerhalb der  $\varepsilon$ -Nachbarschaft eines Kernpunkts liegt, selbst aber keinen Kernpunkt darstellt. Dies gilt vor allem für Randpunkte innerhalb eines Clusters.
- Rauschpunkt: Ein Punkt, der weder ein Kernpunkt noch Dichte-erreichbar ist, wird als Rauschpunkt bezeichnet.

Alle Punkte, die über eine Kette von Kernpunkten Dichte-erreichbar sind bilden letztendlich ein Cluster von Dichte-verbundenen Punkten. Dabei kann der Abstand oder die  $\varepsilon$ -Nachbarschaft über beliebige Distanzfunktionen bzw. Ähnlichkeitsmaße bestimmt werden. Aufgrund dieser Flexibilität wird *DBSCAN* auch häufig als Ausgangslage für weitere Clustering-Algorithmen verwendet.

So greift bspw. der OPTICS-Algorithmus [9] das Grundkonzept von DBSCAN auf. Dieser kann Cluster von unterschiedlicher Dichte erkennen, indem zunächst auf Basis des *minPts*-Parameters eine Kerndistanz  $\delta_\varepsilon(p)$  des Punktes  $p$  ermittelt wird. Diese entspricht dem  $\varepsilon$ -Wert von DBSCAN, ab dem  $p$  zum Kernpunkt wird. Somit wird zur Bestimmung von  $\delta_\varepsilon(p)$  geprüft, für welches  $\varepsilon$  mindestens *minPts* Punkte in der  $\varepsilon$ -Nachbarschaft von  $p$  liegen. Durch die Modifikation wird der  $\varepsilon$ -Parameter des DBSCAN weitestgehend unnötig. Er wird lediglich noch als obere Schranke und zur Begrenzung der Komplexität des Algorithmus verwendet. Mithilfe von  $\delta_\varepsilon(p)$  errechnet OPTICS eine Erreichbarkeitsdistanz für jeden Punkt. Ausgehend von einem beliebigen Datenelement fügt der Algorithmus die Punkte anhand der jeweils geringsten Erreichbarkeitsdistanz in eine Ordnung ein, wodurch gefundene Cluster nacheinander abgearbeitet werden und schließlich als Erreichbarkeitsdiagramm visualisiert werden können [222]. Sowohl OPTICS als auch der DBSCAN besitzen in ihrer ursprünglichen Form, also ohne Verbesserung oder der Angabe einer oberen Schranke, eine Laufzeitkomplexität von  $O(n^2)$  bei  $n$  Datenpunkten.

Auch der bereits erwähnte TRACCLUS-Algorithmus [121] verwendet zur Gruppierung von Trajektorien die Konzepte des DBSCAN. Dabei werden die ursprünglichen Begrifflichkeiten der Dichte-Verbundenheit von Punkten auf Liniensegmente angewendet. So gilt die  $\varepsilon$ -Nachbarschaft für Linien statt für Punkte, und ein Liniensegment wird als Kernlinie definiert, wenn sich innerhalb ihrer  $\varepsilon$ -Nachbarschaft mindestens *minLns* Liniensegmente befinden. Auch die Definitionen für Dichte-Erreichbarkeit und Dichte-Verbundenheit werden analog auf Linien übertragen. Somit verlangt der TRACCLUS-Algorithmus ebenfalls zwei Eingabeparameter, nämlich  $\varepsilon$  und *minLns*, um Cluster aus Dichte-verbundenen Liniensegmenten zu ermitteln. Die hierbei verwendete Distanzfunktion setzt sich zusammen aus der gewichteten Summe von drei Distanzen, nämlich der Paralleldistanz  $d_{\parallel}$  und den bereits zur Segmentierung verwendeten Distanzen  $d_{\perp}$  und  $d_{\Theta}$ . Wie die beiden anderen Algorithmen besitzt auch TRACCLUS in seiner Ursprungsform zum clustern von  $n$  Liniensegmenten eine Laufzeitkomplexität von  $O(n^2)$ .

Im Anschluss an diesen Schritt erzeugt TRACCLUS auf Basis des *minLns*-Parameters eine repräsentative Trajektorie  $\tau_r$  für jedes gefundene Cluster. Hierbei wird zunächst ein Durchschnittsvektor  $\vec{V}$  für das jeweilige Cluster errechnet. Danach wird augenscheinlich in Richtung  $\vec{V}$  mit einer senkrechten Linie zu  $\vec{V}$  über alle Start- und Endpunkte der Segmente im Cluster iteriert. Dabei erzeugt der Algorithmus immer einen neuen Punkt für  $\tau_r$ , wenn mindestens *minLns* Segmente die zu  $\vec{V}$  senkrechte Linie schneiden. Die Lage eines solchen Punktes wird über den Durchschnitt der jeweiligen Schnittpunkte bestimmt. Durch die sequentielle Verbindung der neuen Punkte entsteht  $\tau_r$ . Diese repräsentiert die durchschnittliche Bewegungsinformation eines gefundenen Clusters und eignet sich daher insbesondere auch zur Nutzerstromanalyse.

### 2.4.4 Bewegungsmuster

Im Rahmen des Trajectory Computing modellieren Bewegungsmuster einen in natürlicher Sprache beschreibbaren Zusammenhang von Subtrajektorien. Dabei beziehen diese Muster meist mehrere Entitäten mit ein, die innerhalb eines zeitlich und örtlich begrenzten Bereichs in Interaktion treten [75]. Bspw. kann die Bewegung einer Tierherde auf der Weide oder ein bestimmter Personenstrom im Bahnhof als ein Bewegungsmuster aufgefasst werden. Die Schwierigkeit besteht letztendlich darin, örtliche und/oder zeitliche Zusammenhänge in den vorliegenden Trajektorien zu identifizieren und schließlich zu klassifizieren. Hierzu existieren in der Literatur bereits zahlreiche Ansätze und Algorithmen. So ist bspw. das Verfahren der bewegten Cluster von Kalnis et al. [100] zu nennen, welches die klassische Cluster-Analyse um die zeitliche Komponente erweitert. Laube et al. [119] definieren hingegen eine Reihe typischer Muster, wie bspw. Schwarm-, Führungs-, Stau- oder Kampfverhalten und zeigen Algorithmen auf, um diese effizient zu berechnen. Nach Dodge et al. [51] lassen sich die existierenden Bewegungsmuster in generische und verhaltensspezifische Muster einteilen. Diese können beliebig komplex werden, vor allem wenn neben individuellen auch gruppenabhängige Bewegungsmuster betrachtet werden.

Zur Analyse von Nutzerströmen sind vor allem die verhaltensspezifischen Muster interessant, da diese Aufschluss über das individuelle Verhalten einer Person geben. Dadurch lassen sich auch Ziele und weiterführende Gemeinsamkeiten ganzer Gruppen ableiten, die ebenfalls zur Vorhersage von Nutzerbewegungen verwendet werden können. Während generische Muster typischerweise über einen Clustering- bzw. Join-Ansatz gewonnen werden [97], umfassen verhaltensspezifische Muster meist komplexere Zusammenhänge, die aus den generischen Mustern abgeleitet werden [51]. Zu diesem Zweck bieten sich Methoden der Mustererkennung und des überwachten Lernens (engl.: *supervised learning*) an. Dabei kann ein Klassifikator auf generische Muster eines bekannten Verhaltens trainiert werden, um anschließend Bewegungsabläufe von unbekanntem Verhalten zu klassifizieren. Ein entsprechender Ansatz wird im späteren Verlauf dieser Arbeit vorgestellt und evaluiert.

In manchen Verfahren spielen zur Ermittlung von typischen und wiederkehrenden Bewegungsmustern auch sogenannte *interessante Orte* eine große Rolle, die von mehreren Personen auf ähnliche Weise häufig frequentiert werden [66]. Die Bestimmung solcher Orte ist ein weiterer wichtiger Bereich des Trajectory Computing, welcher im Folgenden vorgestellt wird.

### 2.4.5 Interessante Orte

Interessante Orte beschreiben im Rahmen des Trajectory Computings bestimmte Bereiche, die entweder besonders oft oder ungewöhnlich lange besucht werden, bzw. an denen sich das Bewegungsverhalten der meisten Trajektorien signifikant verändert [74]. Solche Orte werden je nach Szenario auch als *Hotspots*, *stationäre Plätze* oder als *Steh-* bzw. *Stop-Punkte* bezeichnet. Zur

Identifizierung von interessanten Orten aus vorhandenen Trajektorien-Daten stehen wieder verschiedene Ansätze in der Literatur bereit. Moreno et al. [142] folgen bspw. dem Schwerpunkt der vorliegenden Arbeit und extrahieren Kontextinformationen für eine semantische Analyse von Trajektorien, um letztendlich Stop-Punkte besser erkennen und interpretieren zu können. Tiwari und Kaushik [199] nutzen ebenfalls semantische Informationen über stark frequentierte Orte (wie bspw. Sehenswürdigkeiten), um diese als interessante Plätze in GPS-Trajektorien zu markieren. Palma et al. [160] hingegen verwenden einen Geschwindigkeits-basierten Clustering-Ansatz nach den Konzepten von DBSCAN, um Stop-Bereiche aus Einzelpfaden zu extrahieren. Thiagarajan et al. [197] bezeichnen viel befahrene Straßensegmente als Hotspots, die durch eine auffallend niedrige Reisegeschwindigkeit von Fahrzeugen identifiziert werden.

Insgesamt versuchen die genannten Arbeiten interessante Orte vor allem über temporale Eigenschaften der vorhandenen Trajektorien zu bestimmen, indem sie die Änderung der Pfadgeschwindigkeit oder die Aufenthaltsdauer betrachten. In anderen Untersuchungen stehen eher spatiale Merkmale im Fokus mit dem Ziel, dichte bzw. überfüllte Regionen zu erkennen, wie bspw. innerhalb von Gebäuden [4] oder im Mobilfunknetz [89]. In diesem Zusammenhang werden interessante Orte auch oft als *Hotspots* bezeichnet, da sie bspw. mittels Heatmap als heiße Stellen visualisiert werden können. In Anlehnung daran ist das *Hot-Route-Discovery-Problem* zu nennen, bei dem versucht wird, die am häufigsten genutzten Routen in einem Trajektorien-Datensatz zu finden [36]. Im späteren Verlauf der vorliegenden Arbeit wird u.a. auf Hotspots und den beschriebenen Stop-Punkten als interessante Orte einer Trajektorie noch näher eingegangen.

## 2.5 Zusammenfassung

In diesem Kapitel wurden wichtige Grundlagen zur Erfassung und Analyse von Nutzerströmen vorgestellt. Es hat sich gezeigt, dass Infrastruktur-basierte Tracking-Systeme prinzipiell durch verschiedene Technologien, wie Kameras, Druck-, oder Schallsensoren realisiert werden können. Allerdings bietet die Aufzeichnung von Bluetooth- bzw. WLAN-Signalen durch die weite Verbreitung moderner mobiler Geräte eine vielversprechende Alternative an, um Personen mit minimalem Einsatz flächendeckend zu verfolgen. Hierbei ist weder die aktive Beteiligung der Nutzer, noch deren explizite Einwilligung von Nöten. Allerdings unterliegen die entsprechenden Signalbeobachtungen gerade im Innenbereich einem hohen Rauschen und im Fall von WLAN einer niedrigen Abtastrate, die durch das jeweilige Sendeverhalten von Probe Requests bestimmt wird. Daher bietet sich neben der groben Nahbereichserkennung vor allem ein Fingerprinting-Ansatz zur genaueren Lokalisierung und ein probabilistisches Verfahren zur kontinuierlichen Standortverfolgung an. Hierzu wurde als Grundlage der Bayes-Filter und im Speziellen der Partikel-Filter vorge-

stellt, welcher in Verbindung mit den beschriebenen Konzepten des Viterbi-Algorithmus im späteren Verlauf der vorliegenden Arbeit Verwendung findet. Zur Analyse von Zeitreihen kontinuierlicher Positionsschätzungen wurden im letzten Abschnitt relevante Verfahren aus fünf wichtigen Bereichen des Trajectory Computing vorgestellt. Neben den genannten Ähnlichkeitsmaßen spielt hier vor allem Segmentierung und Clustering von Trajektorien eine große Rolle, was bspw. durch den DBSCAN-basierten TRACCLUS-Algorithmus realisiert wird. Zudem konnten in diesem Kapitel technische Details sowie wichtige Begrifflichkeiten für die weitere Verwendung geklärt werden.

Die Publikationen, die im Zusammenhang mit den erwähnten Verfahren genannten wurden, bilden bereits ein breites wissenschaftliches Spektrum ab, welches das hier behandelte Themenfeld in groben Zügen umfasst. Ein genauere Bezug zwischen verwandten Arbeiten und den jeweiligen Problemstellungen wird in den entsprechenden Hauptkapiteln gegeben, die nachfolgend beschrieben sind. Dabei steht zunächst die Datenerfassung im Fokus, ehe auf Basis der vorgestellten Grundlagen weiterführende Methoden für die Extraktion von Kontextinformationen zur Analyse von Nutzerströmen erarbeitet werden.



## 3 Datenerfassung zur Analyse von Nutzerströmen

Wie in den vorherigen Kapiteln deutlich wurde, stellt die Aufzeichnung drahtloser Kommunikationsdaten von mobilen Endgeräten eine einfache, kostengünstige und skalierbare Technik zur Erfassung von Personenansammlungen und Nutzerströmen dar. Dabei wird weder eine Veränderung der vorhandenen Soft- bzw. Hardware noch eine aktive Nutzerbeteiligung vorausgesetzt. Die bloße Aktivität der Kommunikationsschnittstelle mobiler Geräte reicht aus, um Signale im großen Stil unbemerkt abzuhören und interpretieren zu können. Durch die anwachsende Verbreitung von Smartphones bzw. Tablets und die damit verbundene Nutzung von Bluetooth oder WLAN nimmt auch das Verhältnis von potentiell erfassbaren Personen in alltäglichen Szenarien zu.

Daher wird diese Technik bereits in vielen wissenschaftlichen und wirtschaftlichen Bereichen zur Personenerfassung eingesetzt, wie einleitend erläutert wurde. Trotz der diversen Anwendungen konnte bisher nicht abschließend geklärt werden, wie genau abgehörte Signale mobiler Geräte die tatsächliche Nutzerdichte abbilden, da häufig die dafür nötige Grundwahrheit fehlt. Um solche Fragen zu klären, wird zunächst die Datenerfassung unter realen Bedingungen und mit einer bekannten Grundwahrheit näher untersucht.

### 3.1 Datenerfassung unter realen Bedingungen

In einem ersten Schritt gilt es herauszufinden, wie und welche Informationen von aufgezeichneten Bluetooth- bzw. WLAN-Signalen realistischerweise verwendet werden können. Anschließend soll evaluiert werden, wie genau diese Daten die Realität, also die echte Anzahl an anwesenden Personen widerspiegeln. Zu diesem Zweck wird im vorliegenden Teil des Kapitels ein konkreter Versuch unter realen Bedingungen durchgeführt und eine Korrelationsanalyse zwischen ermittelten und tatsächlichen Nutzerströmen angestrebt. Hierdurch sollen erstmalig belastbare Aussagen über die Güte der angewendeten Verfahren getroffen werden können. Die hier vorgestellten Kerninhalte wurden bereits in [185] und in erweiterter Form als Journal-Beitrag in [182] veröffentlicht. Darin enthalten sind die Abbildungen 3.2, 3.3, 3.4, 3.5 und 3.6. Wie in Kapitel 1.3 beschrieben, stammen die Kerninhalte wie Idee, Konzeption, Implementierung und Durchführung der Experimente sowie die Evaluation vom Autor der vorliegenden Arbeit. Im Folgenden wird zunächst ein konkreter Überblick

über verwandte Arbeiten gegeben, um das Thema in den wissenschaftlichen Kontext einzuordnen.

#### 3.1.1 Verwandte Arbeiten

Die Nutzererfassung auf Basis von Signalbeobachtungen hat eine lange Tradition in der Wissenschaft und wird spätestens seit der Vorstellung des RADAR-Systems [16] im Jahr 2000 intensiv erforscht. Durch die starke Verbreitung mobiler Geräte wird das Thema mehr und mehr auch im praktischen Einsatz evaluiert [173, 214]. Hierbei kommen vor allem Bluetooth und WLAN als Tracking-Technologien in Betracht, da diese in nahezu allen Smartphones integriert sind.

So wurde bereits die Anzahl und Dichte von Personen auf Massenveranstaltungen mithilfe von Bluetooth-Beobachtungen kollaborativer Smartphones bestimmt [205, 215]. Larsen et al. [118] gehen einen Schritt weiter und untersuchen mittels erfassbarer Bluetooth-Geräte ganze Nutzerbewegungen und das Verhalten von Personen auf einem großen Musikfestival. Nishide und Takada [152] hingegen analysieren sowohl die Personendichte als auch deren Bewegung durch das Mitschneiden von Bluetooth-Aktivitäten in mobilen Ad-hoc-Netzen. Allerdings zeigen solche Bluetooth-basierten Ansätze einige Nachteile, die bereits in Kapitel 2.1.1 erwähnt wurden. Zum einen fordert die geringe Reichweite eine größere Dichte an Sensoren, zum anderen befinden sich moderne Geräte nur selten im Inquiry-Scan-Modus und sind demnach für Messstationen kaum sichtbar. Diese Erkenntnisse werden auch von Abedi et al. [1] beschrieben, die Bluetooth und WLAN in Bezug auf Architektur, Entdeckungszeit, Signalstärken sowie die Nutzungshäufigkeit vergleichen. Dabei diskutieren die Autoren beide Technologien auf ihre Eignung zur passiven Nutzererfassung und kommen zu dem Schluss, dass WLAN aufgrund kürzerer Entdeckungszeiten und höherer Erkennungsraten deutliche Vorteile gegenüber Bluetooth aufweist. In den durchgeführten Experimenten konnte zudem gezeigt werden, dass von allen MAC-Adressen, die an sechs verschiedenen Orten erfasst wurden, nur 5% zu Bluetooth- und über 90% zu WLAN-Geräten gehörten.

Wissenschaftler sind deshalb verstärkt dazu übergegangen, WLAN als Tracking-Technologie zu verwenden. Dabei wurde u.a. untersucht, wie extrahierte Daten aus WLAN-Management-Frames genutzt werden können, um eine Standortverfolgung durchzuführen [232], Trajektorien von Nutzern auf Straßenkarten zu berechnen [145], soziale Beziehungen zu bestimmen [20], Wartezeiten in Anstehschlangen zu messen [209] oder die Personendichte in einem Gebiet zu approximieren [26]. All diese und auch weitere Arbeiten wie bspw. [101, 164] greifen bei der Evaluation ihrer Ansätze auf Schätzungen bzw. Hochrechnungen zurück, da eine genaue Grundwahrheit in realen Umgebungen häufig fehlt oder nur schwer zu bestimmen ist.

So gibt es nach heutigem Kenntnisstand des Autors keine wissenschaftliche Studie, in der Dichte und Ströme von Personen mittels Bluetooth bzw. WLAN

erfasst und auf Basis einer bekannten Grundwahrheit in einem öffentlichen Gebäude evaluiert wurden. Diese Lücke soll nun im Rahmen der vorliegenden Arbeit geschlossen werden.

### 3.1.2 Ansätze zur Bestimmung von Personendichte und Nutzerströme

Zur Bestimmung von Personendichte und Nutzerströme müssen zunächst Messstationen entsprechende Signalbeobachtungen innerhalb eines Interessensbereichs aufzeichnen. Eine Beobachtung  $o_m$  der Messstation  $m$  besteht dabei aus dem Tupel  $o_m := \{t, \text{MAC}, \text{RSS}\}$ , wobei  $t$  den Zeitpunkt der Beobachtung, MAC die MAC-Adresse des mobilen Geräts und RSS die Empfangssignalstärke repräsentiert. Ein algorithmischer Ansatz zur Identifizierung geeigneter Messstellen wird im späteren Verlauf (vgl. Kapitel 4.1) der vorliegenden Arbeit vorgestellt. Für den Moment soll eine abstrakte Betrachtungsweise ausreichen, die in Abbildung 3.1 schematisch dargestellt ist. Dabei deutet  $r$  den Radius

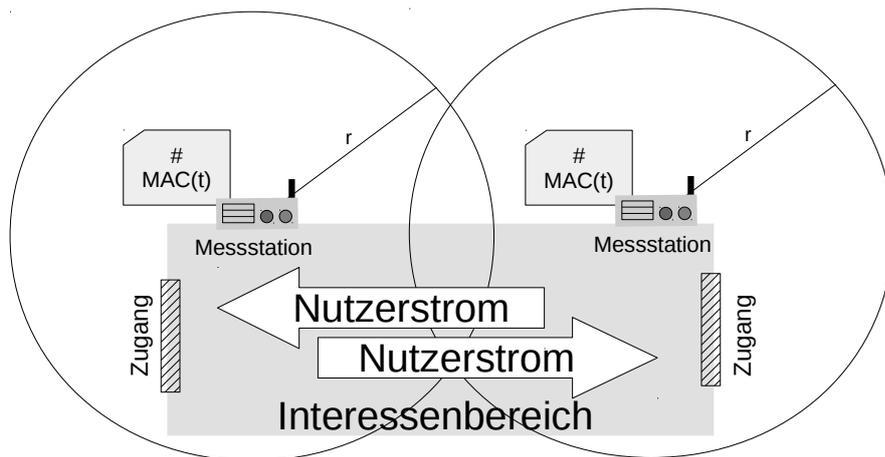


Abbildung 3.1: Schematische Darstellung zur Erfassung von Dichte und Strom.

eines vereinfacht angenommenen Empfangsbereichs an. Die Dichte bzw. der Strom von Personen lässt sich auf dieser Ebene wie folgt beschreiben:

Die Personendichte wird definiert als die Anzahl an individuellen Personen, die sich für eine knappe Zeitspanne in einer bestimmten Flächeneinheit aufhält. Unter der vereinfachten Annahme, dass jede individuelle MAC-Adresse zu einer Einzelperson gehört, lässt sich die Personendichte bzgl. der Fläche, die von einer Messstation beobachtet wird, über die Anzahl der MAC-Adressen bestimmen, die von dieser Messstation innerhalb einer kurzen Zeitspanne aufgezeichnet wird. Es sei an dieser Stelle angemerkt, dass in der Realität nicht jede Person erfasst werden kann und gleichzeitig manche Nutzer aufgrund mehrerer aktiver Geräte häufiger beobachtet werden. Dies wirkt sich natürlich auf

die Schätzung von absoluten Personenzahlen aus, hat aber auf die Bestimmung von relativen Tendenzen im Zeitverlauf wenig Einfluss.

Ein Nutzerstrom wird hingegen definiert als die Menge an Personen, die sich innerhalb einer begrenzten Zeitspanne über ein bestimmtes Gebiet (den Interessenbereich) in die gleiche bzw. ähnliche Richtung bewegt. Anders als die Personendichte, die von jeder Messstation einzeln bestimmt wird, benötigt man zur Erfassung eines einfachen Nutzerstroms mindestens zwei örtlich getrennte Stationen. Deren Beobachtungen lassen sich für ein Gerät anhand der MAC-Adresse zusammenführen. Mithilfe der Zeitinformation kann daraufhin eine grobe Bewegung von einem Messknoten zum anderen errechnet werden. Es sei angemerkt, dass ein solcher Vergleich von Beobachtungen auf Basis einer einfachen Nahbereichserkennung nur dann adäquat funktioniert, wenn die Messknoten, wie in Abbildung 3.1 dargestellt, den Interessenbereich an allen diskreten Zugängen überwachen und ihre beobachtbaren Bereiche möglichst disjunkt sind. Hier wären Bluetooth-Sensoren aufgrund der geringeren Reichweiten theoretisch besser geeignet. Besteht der Interessenbereich allerdings aus einer größeren Freifläche, auf der verschiedenste Nutzerströme möglich sind, reicht die bloße Nahbereichserkennung in der Regel nicht mehr aus, und andere Methoden müssen verwendet werden, um genauere Nutzerbewegungen ermitteln zu können. Ansätze hierzu werden später in Kapitel 4.2 diskutiert.

Im Folgenden soll die einfache Nahbereichserkennung verwendet werden, um unabhängig von aufwendigeren Lokalisierungsmethoden die Eignung zur Nutzerstromerfassung untersuchen zu können. Dabei lassen sich Nutzerbewegungen nur durch den Vergleich aller beobachteten Tupel  $o_i, o_j$  mit  $i \neq j$  feststellen, womit vier prinzipielle Möglichkeiten zur Verfügung stehen:

1. **Naiver Ansatz:** Hierbei werden alle individuellen MAC-Adressen gezählt, die innerhalb einer begrenzten Zeitspanne an den Messstationen  $i$  und  $j$  beobachtet wurden. Dieser Basisansatz berücksichtigt weder die Richtung der Bewegung, noch den Fall, dass sich die beobachtbaren Bereiche überschneiden und somit zu falsch-positiven Zählungen führen können.
2. **Zeit-basierter Ansatz:** Durch die zusätzliche Berücksichtigung der Zeitinformation kann die Bewegung von  $i$  nach  $j$  oder umgekehrt mittels einer positiven Zeitdifferenz zwischen den entsprechenden Beobachtungen bestimmt werden.
3. **RSS-basierter Ansatz:** Durch die Berücksichtigung der Empfangssignalstärke können Beobachtungen ignoriert werden, die einen zu geringen RSS-Wert aufweisen. Dies schränkt den beobachtbaren Bereich ein, womit Überschneidungen vermieden und die falsch-positiven Zählungen reduziert werden können. Allerdings erfordert dies einen festgelegten Grenzwert, der für reale Szenarien schwer zu ermitteln ist. Zur Bestimmung der Bewegung müssen analog zum naiven Ansatz die MAC-Adressen der Beobachtungen verglichen werden.

4. **Hybrider Ansatz:** Hierbei werden alle Informationen verwendet und eine Kombination der vorherigen Ansätze realisiert. Eine Bewegung von  $i$  nach  $j$  wird demnach als solche registriert, wenn die Empfangssignalstärken in den Tupel  $o_i$  und  $o_j$  mit gleicher MAC-Adresse mindestens einmal größer als ein bestimmter Wert  $\theta$  sind und es eine positive Zeitdifferenz zwischen  $o_i$  und  $o_j$  gibt. Auch wenn dieser Ansatz am sinnvollsten erscheint, wirkt sich die Bestimmung von  $\theta$  auf das Ergebnis der Zählung aus und kann dadurch ggf. verfälscht werden.

Die vorgestellten Möglichkeiten sollen nun auf Basis von Bluetooth- bzw. WLAN-Signalbeobachtungen in Hinblick auf eine bekannte Grundwahrheit evaluiert werden. Zur Datenerfassung wurde ein Experiment unter realen Bedingungen durchgeführt, welches nachfolgend vorgestellt wird.

### 3.1.3 Experimenteller Aufbau und Implementierung des Flughafenszenarios

In Anlehnung an die abstrakte Darstellung aus Abbildung 3.1 und den beschriebenen Überlegungen, wurden zwei identische Laptops an geeigneten, örtlich getrennten Stellen im Terminal 1C am Flughafen München platziert, um Bluetooth- und WLAN-Signale von mobilen Geräten in Reichweite aufzuzeichnen. Die Laptops realisieren also die Messstationen und werden nachfolgend als *Monitore* bezeichnet. Der genaue Aufbau ist in Abbildung 3.2 dargestellt und gestaltet sich wie folgt:

Der erste Monitor befindet sich an einem Informationsschalter im öffentlichen Bereich. Dieser ist ca. 20 Meter von der Sicherheitskontrolle und ca. zehn Meter von der Stelle entfernt, an der die Bordkarten der Passagiere abgescannt werden, die durch die Kontrolle in den Sicherheitsbereich laufen. Damit überwacht Monitor 1 den Eingangsbereich der Sicherheitskontrolle, die den Interessenbereich zur Nutzerstromerfassung darstellt, welche in Abbildung 3.2 rot umrandet ist. Der zweite Laptop wird an der Theke eines Schnellrestaurants im Sicherheitsbereich platziert und überwacht u.a. den Ausgang der Sicherheitskontrolle, der ca. zehn Meter entfernt ist. Die Distanz zwischen beiden Laptops beträgt damit knapp 40 Meter. Durch diesen Aufbau ergeben sich folgende Eigenschaften und Vorteile:

- Minimaler Einsatz von zusätzlicher Hardware (zwei Laptops).
- Reduktion der Komplexität durch die Überwachung eines einzelnen deterministischen Nutzerstroms, der nur in eine Richtung verläuft.
- Zugang zu einer genauen Grundwahrheit über die vorhandenen Passagierströme mittels der entsprechenden Bordkarten-Scans, die dem Autor der vorliegenden Arbeit zur Verfügung gestellt wurden.

Beide Laptops wurden zuvor über das *Network Time Protocol* (NTP) synchronisiert, damit die entsprechenden Signalbeobachtungen zeitlich vergleichbar

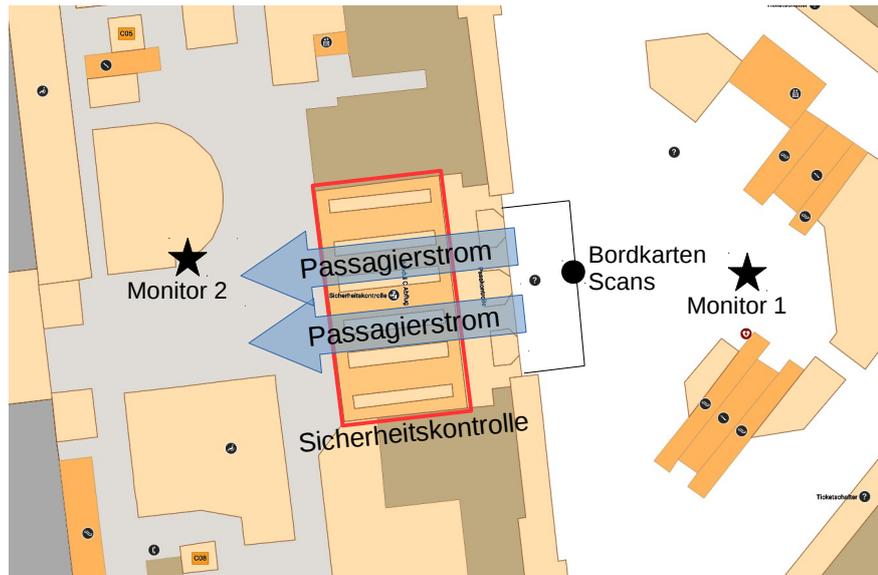


Abbildung 3.2: Kartenausschnitt<sup>2</sup> des Flughafens zur Darstellung des experimentellen Aufbaus mit Lage der Passagierströme durch die Sicherheitskontrolle, Bordkarten-Scans und Monitore.

sind. Jeder Laptop ist ausgestattet mit einem Bluetooth- und einem WLAN-Adapter, der im Monitor-Modus auf dem gebräuchlichen Kanal (engl.: *common channel*) 6 operiert. Über das Linux-Tool *tcpdump*<sup>1</sup> werden alle empfangenen IEEE 802.11 Management-Frames (bis auf Beacon-Frames) aufgezeichnet. Zur Messung von Bluetooth-Signalen wird der *BlueZ-Stack*<sup>3</sup> des Linux-Kernels verwendet, der pro Minute einen Inquiry-Scan initiiert, um Interferenzen auf dem ISM-Band mit WLAN-Signalen gering zu halten. Entsprechende Inquiry-Antworten von Bluetooth-Geräten werden wie die WLAN-Frames in separate Log-Dateien geschrieben. Hieraus lassen sich dann die Beobachtungstupel  $o_m := \{t, \text{MAC}, \text{RSS}\}$  generieren, die zur Bestimmung von Personendichte und Nutzerströme herangezogen werden.

### 3.1.4 Evaluation und Ergebnisse des Flughafenszenarios

Mithilfe der eben vorgestellten Implementierung und dem beschriebenen Versuchsaufbau wurden über 16 Tage lang Bluetooth- und WLAN-Signale aller erreichbaren Geräte von Personen aufgezeichnet. Darunter befinden sich neben den Passagieren, die durch die Sicherheitskontrolle gingen, auch Besucher, Mitarbeiter oder andere Nutzer, die sich durch mindestens einen beobachtbaren Bereich einer Messstationen bewegten ohne dabei ihre Bordkarte zu scannen. Daher werden die gesammelten Daten zunächst verwendet, um allgemeine In-

<sup>1</sup>Siehe: <http://www.tcpdump.org/>

<sup>2</sup>Quelle: Google Maps – <https://maps.google.de>

<sup>3</sup>Siehe: <http://www.bluez.org/>

formationen über die beobachtete Masse zu extrahieren. Anschließend folgt die Untersuchung der Personendichte über die Zeit und darauf aufbauend die Evaluation der vier Ansätze zur Bestimmung von Nutzerströmen.

### Allgemeine Informationen

Während des 16-tägigen Experiments wurden im öffentlichen Bereich des Münchner Flughafens über elf Millionen Probe Requests und 6.600 individuelle SSIDs aufgenommen. Im Sicherheitsbereich waren es erwartungsgemäß etwas weniger, nämlich 8,5 Millionen Probes und 4.000 eindeutige SSIDs. Der Grund ist, dass sich hier in der Regel keine Besucher, sondern nur die Passagiere der jeweiligen Flüge aufhalten. Das Verhältnis zwischen gerichteten- und Broadcast-Probe Requests lag bei 37% im öffentlichen und bei 47% im Sicherheitsbereich. Im Durchschnitt wurden pro Tag 6.211 individuelle WLAN-MAC-Adressen und nur 250 Bluetooth-MAC-Adressen von Monitor 1 beobachtet. Dies entspricht einem Bluetooth/WLAN-Verhältnis von 4% für den öffentlichen Bereich. Da Monitor 2 insgesamt weniger Signale beobachten konnte, wurden hier nur durchschnittlich 3.784 WLAN- und 107 Bluetooth-Adressen pro Tag aufgezeichnet. Daher liegt das Bluetooth/WLAN-Verhältnis für den Sicherheitsbereich bei 2,8%.

Da jede MAC-Adresse in den ersten drei Bytes die IEEE Herstellerkennung kodiert, lässt sich für alle gesammelten MAC-Adressen ein sogenannter *Organizationally Unique Identifier Lookup* durchführen, welcher den Namen des jeweiligen Adapter-Herstellers zurückliefert. Dadurch können erfasste Geräte bereits anhand ihrer Hardware-Adresse grob analysiert und nach der jeweiligen Herstellerkennung zusammengefasst werden. Eine solche Gruppierung auf Basis der Aufzeichnungen im öffentlichen Bereich ist in Abbildung 3.3 dargestellt. Dabei zeigt Grafik 3.3a die Verteilung der am meisten erfassten Bluetooth-Geräte und Abbildung 3.3b die Hersteller der gebräuchlichsten WLAN-Adapter, die im Verlauf des Experiments entdeckt wurden. Die Aufzeichnungen im Sicherheitsbereich lieferten eine vergleichbare Verteilung.

Wie bereits vermutet, reagieren modernere Geräte wie bspw. *Apples iPhone* weniger auf Bluetooth Inquiry-Anfragen, da sie häufig unsichtbar sind. Stattdessen können ältere Modelle von Herstellern wie *Nokia* oder *RIM* (BlackBerry) mittels Bluetooth eher erfasst werden. Für WLAN-Aufzeichnungen ergibt sich ein konträres Bild. Hier zeigt sich wie in vergleichbaren Studien [20, 145] eine signifikante Dominanz von Apple-Geräten. Diese spiegelt jedoch nicht die Realität wider, wonach Apple Ende März 2016 lediglich einen Marktanteil von knapp 19% in Europa besaß [102]. Die Beobachtung ist eher der Tatsache geschuldet, dass viele Apple-Geräte im Vergleich zu Anderen eine größere Anzahl an Probe Requests versenden, um schneller Zugangspunkte in der Umgebung auszumachen. Dadurch steigt logischerweise die Wahrscheinlichkeit, dass diese Geräte auch erfasst werden. Eigene empirische Untersuchungen bestätigen dies und zeigen, dass im Vergleich zu einem Android-Smartphone ca. drei Viertel aller erfassten Probe-Anfragen von dem im Test verwendeten iPhone stam-

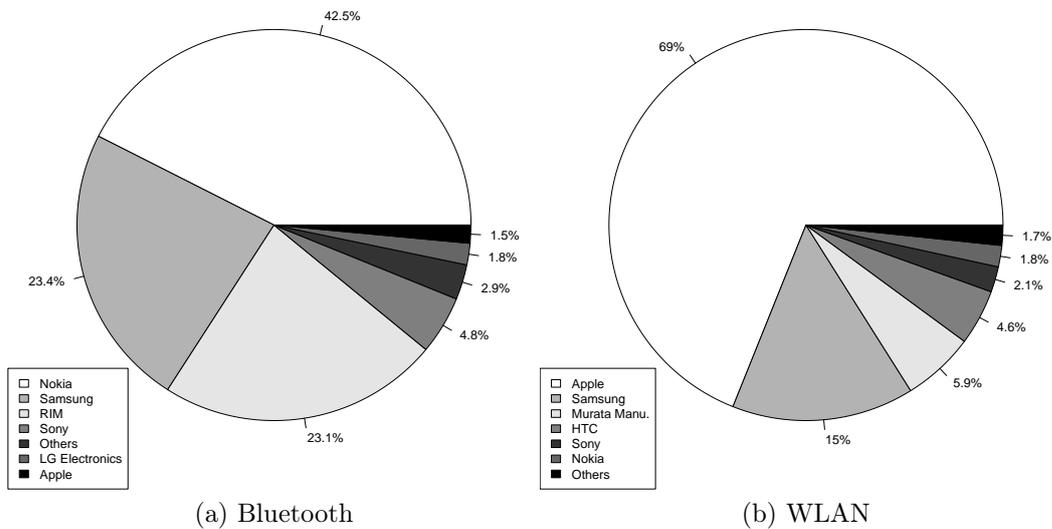


Abbildung 3.3: Verteilung der erfassten Geräte im öffentlichen Bereich gruppiert nach Herstellerkennung.

men [186]. Zudem sei angemerkt, dass im Flughafenumfeld ggf. tatsächlich mehr Apple-Geräte verwendet werden, als in der europäischen Gesamtstatistik angegeben ist. Zusammenfassend ist festzuhalten, dass die ermittelten Verteilungen nur einen Aufschluss darüber geben können, welche Geräteart wie häufig im gegebenen Umfeld erfasst wurde. Dies gilt auch für die nachfolgenden Untersuchungen zur Personendichte.

### Personendichte

Wie in Abschnitt 3.1.2 beschrieben, wird zur Bestimmung der Personendichte jede Messstation einzeln betrachtet und die Anzahl der jeweiligen Beobachtungen im Zeitverlauf analysiert. Das heißt für das vorliegende Experiment, dass die Dichte im öffentlichen und im Sicherheitsbereich sowie für Bluetooth- und WLAN-Aufzeichnungen jeweils separat bestimmt wird. In diesem Fall genügen die Bordkarten-Scans nicht als volle Grundwahrheit, da die Personendichte nicht nur von den Passagieren abhängt, die durch die Sicherheitskontrolle gehen, sondern auch von allen anderen Personen, die sich in den jeweiligen Bereichen aufhalten. Jedoch lassen sich die Bordkarten-Daten für eine grobe Bewertung der Dichtemessung nutzen, da davon auszugehen ist, dass die Personendichte in den jeweiligen Bereichen auch ansteigen bzw. fallen muss, wenn an der Sicherheitskontrolle ein hoher bzw. geringer Andrang herrscht.

So zeigt Abbildung 3.4 den zeitlichen Verlauf der jeweiligen Personendichte exemplarisch für einen Tag des Experiments zusammen mit der Anzahl entsprechender Bordkarten-Scans. Darin ist klar erkennbar, dass die errechnete Personendichte für Bluetooth und WLAN im öffentlichen Bereich stets höher ist als

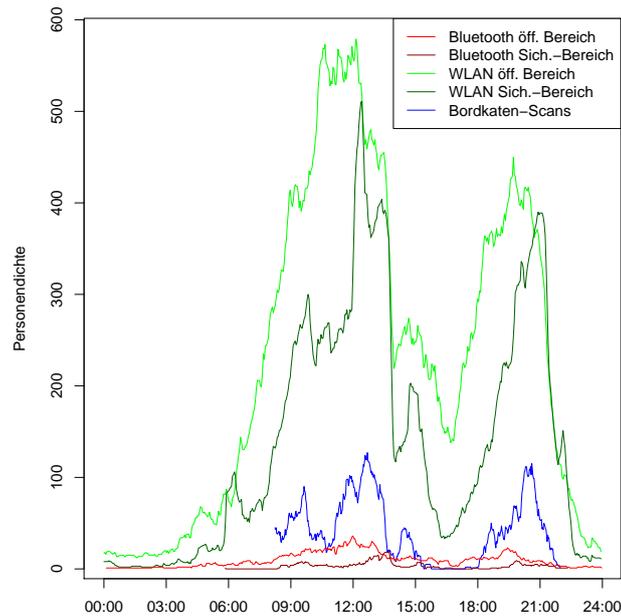


Abbildung 3.4: Tagesverlauf der ermittelten Personendichte im öffentlichen und im Sicherheitsbereich im Vergleich zu Bordkarten-Scans.

im Sicherheitsbereich. Dies ist ein plausibles Ergebnis, da sich im öffentlichen Bereich in der Regel mehr Menschen aufhalten oder diesen durchqueren. Wie zu erwarten, zeigt Bluetooth im Gegensatz zu WLAN eine deutlich niedrigere Dichte, was neben der kleineren Reichweite vor allem auf die geringere Erfassbarkeit von Bluetooth-Geräten zurückzuführen ist. Insgesamt unterschätzt also die Bluetooth-basierte Personendichte die Anzahl der Bordkarten-Scans, während die WLAN-basierte Berechnung die Verteilung stark überschätzt. Dies wiederum kommt durch die große Empfangsreichweite von WLAN zustande.

Wichtiger als die absoluten Personenzahlen sind aber die zeitlichen Tendenzen, die sich zur Analyse der Passagierströme nutzen lassen. So liegen die lokalen Maxima vor allem bei den WLAN-Berechnungen in einer positiven zeitlichen Reihenfolge mit den Spitzenwerten der Bordkarten-Scans. Dabei ergibt sich stets ein ähnliches Bild: Zunächst erfolgt ein Anstieg der Dichte im öffentlichen Bereich und dann fast zeitgleich der Anstieg von Bordkarten-Scans und Personenanzahl im Sicherheitsbereich. Danach flachen die Kurven wieder ab. In Hinblick auf das Flughafen-Szenario spiegeln diese Beobachtungen den erwartbaren Ablauf wieder: Die Passagiere kommen in den von Monitor 1 beobachteten Bereich. Sobald ihre Bordkarten gescannt wurden, gehen sie zur Sicherheitskontrolle und werden von Monitor 2 erfasst. Nach der Kontrolle verlassen sie den überwachten Bereich. Dieser Ablauf bildet die Grundlage der zu erwartenden Passagier- bzw. Nutzerströme.

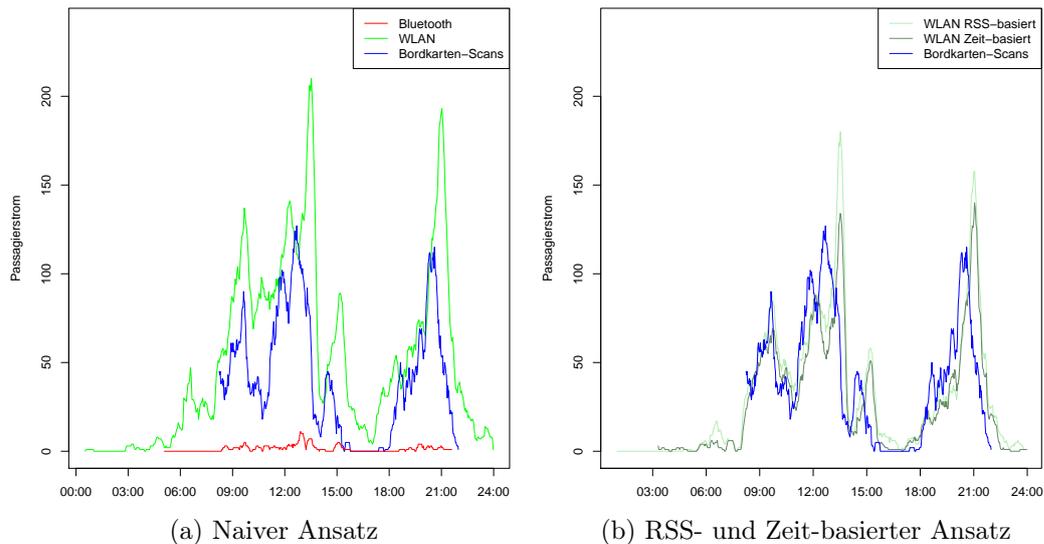


Abbildung 3.5: Errechnete Passagierströme über verschiedene Ansätze im Vergleich zu Bordkarten-Scans an einem Tag.

## Nutzerströme

Zur Bestimmung der Nutzerströme auf Basis einer Nahbereichserkennung wurden im Abschnitt 3.1.2 vier Ansätze vorgestellt. Diese sollen nun mithilfe der Daten aus den Bordkarten-Scans einzeln evaluiert werden. Hierzu zeigt Abbildung 3.5 die errechneten Passagierströme im Vergleich zur Grundwahrheit exemplarisch für einen Tag des Experiments. Abbildung 3.5a veranschaulicht die Ergebnisse des naiven Ansatzes für beide Technologien. Dabei lässt sich erneut feststellen, dass Bluetooth den Verlauf der Bordkarten-Scans immer unterschätzt, während die WLAN-basierten Ströme stets darüber liegen. Im Fall von WLAN ist dies ein unerwartetes Verhalten, da nicht alle Personen erfassbar sind. Somit müsste die errechnete Passagierzahl eher unterhalb der Grundwahrheit liegen, selbst wenn einige Nutzer mehr als ein aktives Gerät bei sich tragen oder das Personal ohne Bordkarte durch die Sicherheitskontrolle geht. Die hohen Werte resultieren demnach aus falsch-positiven Zählungen, die durch den naiven Ansatz und eine Überlappung der beobachtbaren Bereiche entstehen.

Um solche Effekte zu vermeiden, sollte eine zusätzliche Information wie die Empfangssignalstärke oder die Zeitdifferenz mit berücksichtigt werden. Abbildung 3.5b stellt dazu die Ergebnisse des RSS- bzw. des Zeit-basierten Ansatzes für WLAN dar, die eine augenscheinlich bessere Approximation der Grundwahrheit bewirken. Dabei zeigt der Zeit-basierte Ansatz eine genauere Annäherung und kommt im Gegensatz zum RSS-basierten Ansatz ohne Festlegung einer oberen Schranke aus. Im Fall von Bluetooth können diese Erweiterungen keine Verbesserung erwirken, da bereits beim naiven Ansatz zu wenig Passa-

giere erfasst wurden.

Unter allen Ansätzen zeigen die Ergebnisse der Nutzerstromberechnungen eine zeitliche Verzögerung gegenüber der Grundwahrheit. Diese ergibt sich durch den Versuchsaufbau, in welchem die Passagiere zunächst ihre Bordkarte scannen und dann eine gewisse Zeit benötigen, um den von Monitor 1 beobachteten öffentlichen Bereich zu verlassen. Rechnet man diese Verzögerung heraus, überlagern sich die Kurven auch bzgl. der X-Achse und es zeigt sich augenscheinlich ein enger Zusammenhang zwischen Berechnung und Grundwahrheit. Für eine objektive Bewertung dieser Beziehung wird nachfolgend die Pearson-Korrelation verwendet, die ein anerkanntes und anschauliches Maß zur Beschreibung des linearen Zusammenhangs zweier Variablen darstellt. Dabei wird ein Korrelationskoeffizient  $r$  mit  $-1 \leq r \leq 1$  berechnet. Das Vorzeichen trifft eine Aussage darüber, ob eine positive oder eine negative lineare Korrelation vorliegt. Der Betrag hingegen gibt den Grad des Zusammenhangs an. Dieser lässt sich nach Dancey und Reidy [46] wie folgt einordnen und in natürlicher Sprache beschreiben:

Wertebereich	Grad der Korrelation
$ r  \approx 0$	unkorreliert
$0,1 \leq  r  \leq 0,3$	schwach
$0,4 \leq  r  \leq 0,6$	moderat
$0,7 \leq  r  \leq 0,9$	stark
$ r  = 1$	perfekt

Um nun die bestmögliche Korrelation zwischen Messung und Grundwahrheit anhand von  $r$  zu bestimmen, müssen die Messungen zunächst zeitlich mit den Bordkarten-Scans übereinstimmen. Hierzu wird die oben angesprochene Verzögerung herausgerechnet, so dass der durchschnittliche Korrelationskoeffizient maximal wird. Danach erfolgt die Berechnung von  $r$  für alle Nutzerströme, die mittels Bluetooth und den vier WLAN-basierten Ansätzen ermittelt wurden. Die Ergebnisse hierzu finden sich in Tabelle 3.1 und unterstreichen die bisher gewonnenen Erkenntnisse. Dabei weisen sowohl Bluetooth als auch der naive WLAN-basierte Ansatz die schwächste Korrelation von ca. 0,4 im Durchschnitt auf. Die anderen Methoden erreichen hingegen höhere Durchschnittswerte von bis zu 0,57. Dies zeigt eine deutliche Verbesserung an. Interessant ist, dass Bluetooth trotz der wenigen erfassten Geräte eine etwas höhere durchschnittliche Korrelation aufweist als der naive WLAN-basierte Ansatz.

Bei keiner Methode wird jedoch eine starke Korrelation von mehr als 0,65 im Durchschnitt erreicht. Die Ursache hierfür liegt wieder in den falsch-positiven Zählungen, die bspw. auch in der Nacht entstehen, in der die Sicherheitskontrolle geschlossen ist. Es sei angemerkt, dass solche Beobachtungen auch durch das Personal verursacht werden können und nicht unbedingt auf einen Systemfehler zurückzuführen sind. Mithilfe des Wissens, dass die Sicherheitskontrolle nur zwischen 6 und 22 Uhr offen ist, lassen sich solche Zählungen ausschließen,

	Blt	WLAN naiv	WLAN RSS	WLAN Zeit	WLAN Hyb.
Max	0,73	0,82	0,93	0,93	0,93
∅	0,44	0,41	0,56	0,47	0,57

Tabelle 3.1: Korrelationskoeffizienten für alle Ansätze nach Synchronisation mit der Grundwahrheit.

	Blt	WLAN naiv	WLAN RSS	WLAN Zeit	WLAN Hyb.
Max	0,79	0,86	0,91	0,91	0,91
∅	0,53	0,61	0,74	0,63	0,75

Tabelle 3.2: Korrelationskoeffizienten nach Synchronisation mit der Grundwahrheit und einer eingeschränkten Beobachtungsdauer.

die sich negativ auf die Korrelationsanalyse auswirken. Dieser Schritt ist generalisierbar, da in vielen Bereichen eine vergleichbare Hintergrundinformation über das Anwendungsszenario vorliegt. Die Ergebnisse der Korrelationsanalyse für eine eingeschränkte Beobachtungsdauer von 6 bis 22 Uhr sind in Tabelle 3.2 festgehalten. Dabei zeigt sich, dass der durchschnittliche Korrelationskoeffizient wie erwartet bei allen Verfahren gesteigert werden konnte. Die größte Verbesserung um ca. 48% ergab sich im Fall des naiven WLAN-basierten Ansatzes. Der Grund ist, dass der naive Ansatz die höchste falsch-positiv Rate aufweist, die durch einen eingeschränkten Beobachtungszeitraum am meisten verringert wird. Im Gegenzug konnte der Bluetooth-basierte Ansatz, der wegen seiner geringen Reichweite eine niedrige falsch-positiv Rate besitzt, nur um ca. 20% gesteigert werden. Bluetooth weist damit nach wie vor die schwächste der hier berechneten Korrelationen von 0,53 auf. Ebenfalls werden im Fall von WLAN durch den naiven und den Zeit-basierten Ansatz lediglich moderate Korrelationen im Durchschnitt erzielt. Durch Hinzunahme der RSS-Information kann jedoch der beobachtbare Bereich eingeschränkt und die falsch-positiven Zählungen weiter minimiert werden. Dadurch wird sowohl über den hybriden als auch über den RSS-basierten Ansatz eine durchschnittlich starke Korrelation von bis zu 0,75 erreicht. Unter Berücksichtigung der Einfachheit des Verfahrens stellt dies eine brauchbare Qualität für die Nutzerstromerfassung dar. Allerdings können damit eher tendenzielle Verläufe als absolute Personenströme verlässlich analysiert werden.

Für eine abschließende Bewertung zeigt Abbildung 3.6 den Gesamtverlauf der ermittelten Nutzerströme im Vergleich zur Grundwahrheit für jeden der 16 Tage des Experiments. Dabei wird nochmals deutlich, dass Bluetooth zur Bestimmung von Passagierströmen ungeeignet ist, da schlichtweg zu wenig Geräte erfasst werden. WLAN hingegen erreicht passable Resultate, die den wahren Verlauf (blaue Kurve) adäquat widerspiegeln. Interessant ist hierbei, dass der naive Ansatz über die falsch-positiv Zählungen auch die absoluten Personen-zahlen genauer abschätzt, während der hybride Ansatz die Anzahl konsequent

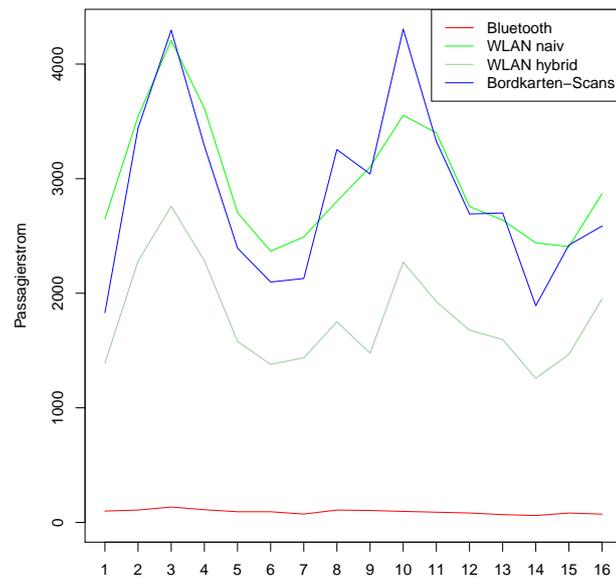


Abbildung 3.6: Ermittelte Nutzerströme und Bordkarten-Scans über den kompletten Zeitraum des Experiments.

unterschätzt, dafür aber den Kurvenverlauf der Bordkarten-Scans präziser wiedergibt. Es ist anzunehmen, dass mit Zunahme von WLAN-aktiven Geräten im Alltag auch der hybride Ansatz die echte Anzahl genauer approximieren kann, weshalb dieser für eine Nutzerstromerfassung unter realen Bedingungen geeigneter erscheint.

### 3.1.5 Zusammenfassung und Diskussion

Anhand der eben vorgestellten Ergebnisse konnte dargelegt werden, wie genau sich Nutzerströme unter realen Bedingungen mithilfe aufgezeichneter Bluetooth- bzw. WLAN-Signale erfassen lassen. Dabei wurde deutlich, dass ein einfaches Zählen der beobachteten MAC-Adressen, wie es in vielen verwandten Arbeiten oder Projekten vorkommt, nur einen moderaten Zusammenhang mit der Realität aufweist und somit häufig nicht ausreicht. Es konnte zudem gezeigt werden, dass sowohl Bluetooth- als auch WLAN-Signale prinzipiell nutzbar sind, um Rückschlüsse auf die Nutzer zu ziehen. Allerdings wurde mittels Bluetooth nur ein Bruchteil der vorhandenen Geräte erfasst, was sich nachweislich auch negativ auf die Genauigkeit der Dichte- und Nutzerstromberechnungen auswirkte. Im besten Fall konnte eine Relation zur Grundwahrheit mit einem Korrelationskoeffizienten von 0,53 festgestellt werden, was kein solider Wert für ein verlässliches Tracking-System ist. Da WLAN hier eindeutig die besseren Ergebnisse zeigt, soll der Fokus im weiteren Verlauf der Arbeit auf einer WLAN-basierten Standortverfolgung liegen.

Insbesondere durch die Einbeziehung von Empfangssignalstärke und Zeit konnten über WLAN-Beobachtungen genauere Nutzerströme ermittelt werden. Durch zusätzliche Hintergrundinformationen über das Anwendungsszenario ergaben sich letztendlich stark korrelierte Zusammenhänge zwischen den Berechnungen und der Grundwahrheit. Allerdings konnte die absolute Anzahl an Personen nur selten approximiert werden. Zudem zeigte die Nahbereichserkennung bei der Bestimmung von Nutzerströmen eine hohe falsch-positiv Rate, die in der zugrundeliegenden Versuchsumgebung durch sich überschneidende Beobachtungsbereiche verursacht wurde. Dies kann entweder durch einen größeren Abstand der Monitore, einen festen Grenzwert für die Empfangssignalstärke oder über erweiterte Lokalisierungsmethoden wie bspw. einem Fingerprinting-Ansatz verhindert werden. Es sei an dieser Stelle darauf hingewiesen, dass die hier erreichten Ergebnisse auf einem realistischen, aber speziellen Szenario basieren, welches ggf. einen signifikanten Einfluss auf die Signalbeobachtungen hat. Demnach sind weitere Experimente in anderen Umgebungen erforderlich, um die hier gewonnenen Erkenntnisse zu validieren.

In Hinblick auf das durchgeführte Experiment und den dabei erreichten Ergebnissen lässt sich zusammenfassend festhalten, dass WLAN bereits heute als Tracking-Technologie in realen Umgebungen eingesetzt werden kann, um die Personendichte und Nutzerbewegungen zwar nicht absolut, aber zumindest tendenziell abzuschätzen. Durch die anwachsende Verbreitung mobiler Geräte ist anzunehmen, dass in naher Zukunft auch genauere Berechnungen von absoluten Personenzahlen möglich sind. Allerdings bedarf es aufgrund der großen Reichweite und des nicht-deterministischen Probe-Verhaltens von WLAN-aktiven Geräten genauere Methoden als die Nahbereichserkennung, um Nutzerströme auch in komplexen Szenarien erfassen zu können. Diese werden in einem späteren Kapitel der vorliegenden Arbeit vorgestellt und evaluiert.

## 3.2 Simulationsumgebung zur Datenerfassung

Die gewonnenen Erkenntnisse aus dem realen Experiment sollen nun im zweiten Teil des Kapitels genutzt werden, um eine realitätsnahe Simulationsumgebung für WLAN-Beobachtungen zu entwickeln. Diese muss in der Lage sein, Signale von Endgerät-Nutzern, die durch ein gegebenes Gebäude laufen, möglichst realistisch zu simulieren, um letztendlich weitere Versuche mit einer entsprechenden Grundwahrheit durchführen zu können.

Die hier entwickelte Simulationsumgebung basiert im Wesentlichen auf drei aufeinander aufbauenden Modellen: Einem Umgebungsmodell für ein entsprechendes Gebäude, einem Bewegungsmodell zur Simulation von menschlichem Bewegungsverhalten in diesem Gebäude und einem neuartigen Aussendungsmodell für IEEE 802.11 Probe Requests. Die beiden ersten Modelle wurden bereits in [175] und das Ausbreitungsmodell in [180] veröffentlicht. Bei beiden Publikationen stammen alle Inhalte, wie im Kapitel 1.3 erwähnt, allein vom Autor der vorliegenden Arbeit. Die in Unterkapitel 3.2.4 beschriebene Modell-

validierung wurde für die vorliegende Arbeit neu durchgeführt und ist noch nicht veröffentlicht.

### 3.2.1 Umgebungsmodell für Gebäude

Umgebungsmodelle für Gebäude können im Prinzip beliebig aufwendig werden, wenn bspw. mehrstöckige, weitläufige und komplexe öffentliche Einrichtungen modelliert werden sollen. Daher ist es für den Entwurf eines geeigneten Modells wichtig, einen guten Kompromiss zwischen Skalierbarkeit, Verfügbarkeit, Zweckdienlichkeit und Einfachheit zu finden. Zudem muss jedes Umgebungsmodell die Fähigkeit besitzen, Ortseigenschaften und Geometrie in geeigneter Weise darzustellen [217].

Im Bereich der Indoor-Navigation werden deshalb vermehrt Bitmap-Grafiken von Gebäudeplänen als Umgebungsmodell eingesetzt [216,218]. Diese erfüllen die meisten der oben genannten Anforderungen bereits, da sie auf minimalistische Weise die geometrischen Eigenschaften eines Gebäudes in einem Pixel-basierten Koordinatensystem konservieren. Bitmaps sind von geringer Größe und Komplexität. Sie können prinzipiell mithilfe eines jeden Bildbearbeitungsprogramms aus beliebigen Gebäudeplänen erstellt werden, womit eine hohe Verfügbarkeit einhergeht. Zudem lassen sich alle nicht benötigten Informationen leicht entfernen, was die geforderte Einfachheit des Modells sicherstellt. Durch den starken Bezug zum Koordinatensystem zählen Bitmaps in diesem Zusammenhang zu den *metrischen* Umgebungsmodellen. Im Gegensatz dazu, modellieren *topologische* Modelle die Beziehungen zwischen einzelnen Orten, wie bspw. durch einen Zugangsgraph [217]. Auch solche topologischen Informationen lassen sich auf Bitmaps übertragen.

Aus den genannten Gründen wird für die hier entwickelte Simulationsumgebung ebenfalls eine Bitmap-Darstellung eines beliebigen Raumplans als Umgebungsmodell verwendet. Dabei wird das Originalbild zunächst in eine binäre Bitmap konvertiert. Anschließend folgt ein Vorverarbeitungsschritt, bei dem alle störenden und nicht benötigten Elemente wie bspw. Türen, Markierungen, Raumnamen, usw. entfernt werden. Allerdings dürfen dabei die topologischen Informationen über vorhandene Räume nicht verloren gehen, damit sie in einem späteren Bewegungsmodell verwendet werden können. Zu diesem Zweck existieren auch automatisierte Verfahren, die bspw. in [217] aufgezeigt werden. Als Resultat liegt schließlich ein minimales und vereinfachtes metrisches 2D-Modell des betrachteten Gebäudes mit Informationen über vorhandene Räume vor. Darin kann über den binären Pixelwert zwischen begehbaren und nicht-begehbaren Flächen unterschieden werden. Weiße Stellen, die mit 1 kodiert sind, zeigen begehbare Regionen an, während schwarze Pixel den Wert 0 haben und Hindernisse bzw. Wände darstellen, die nicht überquert werden können.

Betrachtet man alle weißen Pixel als Knoten, welche die Menge  $V$  definieren und alle direkten Verbindungen zwischen zwei benachbarten Pixeln  $i$  und  $j$  mit  $i, j \in V$  als gewichtete Kanten  $e_{i,j} \in E$  mit dem Kantengewicht

$f(e_{i,j}) = \delta(i, j)$ , so lässt sich die topologische Eigenschaft des Modells durch einen kantengewichteten Graphen  $G = (V, E)$  abbilden. Dieser wird im Folgenden als Navigations- bzw. Routing-Graph verwendet, um menschliche Bewegungen zu simulieren. Entsprechende Entitäten können sich darauf komplett frei in jede Richtung durch das Gebäude bewegen. Um jedoch unrealistische bzw. willkürliche Bewegungsabläufe zu vermeiden, bedarf es eines geeigneten Modells, mit dessen Hilfe sich ein realitätsnahes Bewegungsverhalten von Menschen umsetzen lässt.

### 3.2.2 Bewegungsmodell für Personen im Gebäude

Bewegungsmodelle beschreiben das Bewegungsverhalten, also die Veränderung von Geschwindigkeit, Richtung und Aufenthaltsort einer mobilen Entität über die Zeit. Solche Modelle werden in der Literatur für diverse Bereiche diskutiert. Ein umfassender Überblick hierzu wird bspw. von Bai und Helmy [18] gegeben. In diesem unterteilen die Autoren bestehende Modelle in zufällige (engl.: *random*) und abhängige bzw. restriktive Ansätze.

In der erstgenannten Kategorie befinden sich weit verbreitete Modelle, wie bspw. das *Random Waypoint Model* oder das *Random Walk Model*, die aufgrund ihrer Einfachheit häufig als Benchmark benutzt werden. Dabei werden keinerlei Einschränkungen bzgl. der Mobilität getroffen und die Bestimmung von Geschwindigkeit, Richtung und Ziel geschieht komplett zufällig. Dies widerspricht allerdings dem natürlichen Bewegungsverhalten von Menschen, welche in der Regel nicht willkürlich, sondern mit einer bestimmten Absicht ein örtliches Ziel wählen, das sie auf wohl-definierten (und oftmals kürzesten) Pfaden erreichen [96]. In realen Umgebungen wird die Pfadauswahl zudem von Hindernissen und anderen lokalen Gegebenheiten beeinflusst. So bewegen sich Fußgänger in der Stadt bspw. auf Bürgersteigen entlang des Straßennetzes, oder Besucher eines Gebäudes müssen die vorhandenen Gänge und Treppen benutzen, um in entsprechende Räume zu gelangen. Dieser Umstand wird durch restriktive Bewegungsmodelle, wie dem *Obstacle Mobility Model* oder dem *Pathway Mobility Model* berücksichtigt [18]. Gerade letzteres ist für die Simulation von menschlichen Bewegungen in Gebäuden interessant und soll daher näher betrachtet werden.

Das *Pathway Mobility Model* basiert zwar auf dem *Random Waypoint Model*, in welchem ein Ziel für jeden Bewegungsschritt zufällig gewählt wird, allerdings wird die Bewegungsfreiheit auf vordefinierte Kanten eines Navigationsgraphen eingeschränkt. Dadurch bewegt sich eine mobile Entität in einer pseudo-zufälligen Art auf kürzester Strecke zum gewählten Ziel. Zudem wird der Startpunkt ebenfalls willkürlich festgelegt. Wurde ein Ziel erreicht, verweilt die Entität dort für eine gewisse Zeit und sucht sich erneut ein zufälliges Ziel. Diese Prozedur wird solange wiederholt, bis die Simulation endet. Damit erfüllt das *Pathway Mobility Model* bereits die folgenden Anforderungen für Personenbewegungen im Gebäude:

- Eine Person wählt ein bestimmtes Ziel  $d_i \in D$  aus der Menge an möglichen Zielen in einem Gebäude und versucht dieses auf dem kürzesten Pfad zu erreichen.
- Dabei ist die Pfadwahl auf die begehbaren Bereiche eingeschränkt, die den momentanen Aufenthaltsort mit dem aktuellen Ziel  $d_i$  verbinden.
- Wird  $d_i$  erreicht, verweilt die Person eine gewisse Zeit an diesem Ort, um bspw. eine bestimmte Aufgabe zu erledigen.
- Im Anschluss wird ein neues Ziel  $d_j \in D$  gewählt und wieder auf kürzestem Wege aufgesucht.

Allerdings wird bisher nicht berücksichtigt, dass Menschen ihren Zielort nicht zufällig wählen. Gerade innerhalb von Gebäuden verfolgen Besucher in der Regel eine bestimmte Absicht und suchen deshalb einen entsprechenden Ort auf, wie bspw. das Büro zum Arbeiten, den Vorlesungssaal zum Studieren oder die Cafeteria zum Essen. Außerdem tauchen Personen nicht plötzlich irgendwo im Gebäude auf, sondern betreten bzw. verlassen dies über einen geeigneten Eingang. Aus diesen Überlegungen heraus, wird das *Pathway Mobility Model* zunächst um die folgenden Punkte erweitert und anschließend in der Simulationsumgebung verwendet:

- Eine Person betritt und verlässt das Gebäude jeweils durch eine Eingangstür  $\tau$ , welche mit einer Wahrscheinlichkeit  $p(\tau)$  für den Startpunkt und für das letzte Ziel gewählt werden muss. Eine solche Modellierung erlaubt es, Gebäude mit verschiedenen Eingängen wie bspw. einem Haupt-, Neben- oder Hintereingang zu berücksichtigen, die unterschiedlich häufig genutzt werden. Dies setzt allerdings voraus, dass die Eingänge und deren Nutzungshäufigkeit im Umgebungsmodell bekannt sind.
- Unter der Annahme, dass Menschen einen Zielort mit einer gewissen Absicht wählen, ist die Wahrscheinlichkeit hoch, dass innerhalb eines Gebäudes stets ein Raum (bspw. Büro, Wartezimmer, Toilette, usw.) als Ziel gewählt wird. Gänge oder Treppen hingegen werden eher durchquert und nicht zur Erledigung bestimmter Aufgaben genutzt. Aufgrund dieser Annahmen wird die Menge aller möglichen Ziele definiert als  $D := \{r_1, r_2, \dots, r_n\}$ , wobei  $r_i$  einen von  $n$  Räumen des Umgebungsmodells repräsentiert.
- Jeder Zielort  $d_i$  wird von einer Person in Abhängigkeit der Zeit  $t$  mit der Wahrscheinlichkeit  $p(d_i|t)$  gewählt. Dadurch lässt sich die Wahl eines Ziels realistischer modellieren, da gewisse Räume zu bestimmten Zeiten mit einer höheren Wahrscheinlichkeit ausgewählt werden, wie bspw. die Cafeteria zur Mittagszeit oder ein Wartezimmer während der Sprechstunde. Allerdings erfordert dies ein zum Teil umfangreiches apriorisches Wissen über die Umgebung und das zugrunde liegende Szenario, was

**Algorithmus 2** Simulation von Personenbewegungen im Gebäude

---

**Require:** Navigationsgraph  $G$ , Zeit  $t$

```

1:  $P :=$  Anzahl Personen
2: for ( $\rho = 1 : P$ ) do
3:    $Z :=$  Anzahl Ziele
4:   start = wähle  $\tau$  mit  $p(\tau)$ 
5:   for ( $i = 1 : Z$ ) do
6:     if ( $i == Z$ ) then
7:       ziel = wähle  $\tau$  mit  $p(\tau)$ 
8:     else
9:       ziel = wähle  $d_i \in D$  mit  $p(d_i|t)$ 
10:    pfad = berechne_dijkstra( $G$ , start, ziel)           ▷ Pfad speichern
11:     $t = t + \Delta t$                                    ▷ Zeit Update
12:    start = ziel

```

---

beliebig komplex sein kann. Liegt kein solches Wissen vor, kann auch eine Gleichverteilung oder eine Verteilung über die jeweilige Raumgröße angenommen werden.

Auf Basis des vorgestellten Umgebungs- und Bewegungsmodells lässt sich eine beliebige Anzahl an Personen simulieren, die sich durch ein gegebenes Gebäude auf kürzesten Pfaden zu bestimmten Zielorten bewegt. Die Pfadberechnung basiert im vorliegenden Fall auf dem Dijkstra-Algorithmus, der neben dem Start- und Zielort als Eingabeparameter den Navigationsgraphen des Bewegungsmodells erhält. An jedem erreichten Ziel wird eine gewisse Zeit gewartet und anschließend ein neues Ziel mit einer bestimmten Wahrscheinlichkeit gewählt. Zum Schluss wird das Gebäude über einen der Eingänge verlassen. Der gesamte Ablauf ist abschließend als Pseudocode in Algorithmus 2 zusammengefasst. Auf Grundlage der erzeugten räumlich-zeitlichen Trajektorien werden schließlich Probe Requests mithilfe eines neuartigen Aussendungsmodells simuliert, das im nächsten Abschnitt beschrieben ist.

### 3.2.3 Aussendungsmodell für Probe Requests

Wie aus Abschnitt 2.1.2 hervorgeht, führen mobile Endgeräte in unregelmäßigen Abständen einen aktiven Netzwerk-Scan durch, sobald ihr WLAN-Modul eingeschaltet ist. Dieses nicht-deterministische Sendeverhalten soll nun durch ein geeignetes Aussendungsmodell für IEEE 802.11 Probe Requests möglichst realitätsnah wiedergegeben werden.

In der Literatur finden sich hierzu nur wenige bzw. unzureichende Modelle, da viele Arbeiten die Erfassbarkeit mobiler Geräte entweder allgemein [90] oder empirisch bestimmen [62]. Goodall et al. [70] geben hingegen ein Modell zur Entdeckungswahrscheinlichkeit von Fahrzeugen im Straßennetz an, die von WLAN-Sensoren erfasst werden können. Dabei identifizieren die Autoren drei

Einflussfaktoren, nämlich die Zeitspanne  $T$ , die ein Fahrzeug im Sensor-Bereich verbringt, die Scan-Frequenz  $\omega$  des Geräts sowie eine Erfolgsrate für die Transmission  $\rho$ . Der Ansatz vernachlässigt allerdings die Nutzeraktivität, welche das Sendeverhalten nachweislich beeinflusst. Außerdem wird  $\omega$  empirisch ermittelt und mit einem festen Wert belegt, was die Generalisierbarkeit des Modells verringert. Musa und Eriksson [145] verwenden für ihren Tracking-Ansatz ein HMM und beschreiben dazu ein Modell für die Emissionswahrscheinlichkeit  $p(\text{obs} | s)$ . Diese gibt an, wie hoch die Wahrscheinlichkeit ist, in einem lokalen Zustand  $s$  eine WLAN-Beobachtung  $\text{obs}$  von einem mobilen Gerät zu machen. Ähnlich wie in der zuvor genannten Arbeit beziehen sich die Autoren auf Straßensegmente und geben einen konstanten Wert für die Sendewahrscheinlichkeit mit  $p_{\text{tx}} = \frac{1}{300}$  an, da ein geeignetes Aussendungsmodell fehlt.

Ein solch deterministischer Wert ist jedoch sehr spezifisch und kann das unregelmäßige Probe-Verhalten nicht generisch abbilden, wie eigene empirische Untersuchungen darlegen [186]. Ein probabilistischer Ansatz, der statt eines definierten Werts eine Wahrscheinlichkeitsverteilung betrachtet, erhöht die Flexibilität des Modells und scheint deshalb zur Modellierung des Sendeverhaltens geeigneter zu sein. Im Folgenden wird deshalb ein entsprechendes Aussendungsmodell vorgestellt, was die genannten Schwächen bisheriger Ansätze ausgleicht. Analog zu den verwandten Vorarbeiten und auf Basis eigener Beobachtungen werden zunächst drei Faktoren bestimmt, die einen nachweisbaren Einfluss auf die Sendewahrscheinlichkeit  $p_{\text{tx}}$  nehmen:

1. **Nutzeraktivität:** Wie bereits erwähnt, hat die Nutzeraktivität einen direkten Einfluss auf das Sendeverhalten. So führen die meisten mobilen Endgeräte einen unmittelbaren aktiven Scan durch, wenn der Nutzer das Gerät einschaltet bzw. der Standby-Modus verlassen wird. Dieser Umstand wird deshalb unabhängig von den übrigen Faktoren modelliert.
2. **Zeit:** Probe Requests werden zwar in unregelmäßigen Intervallen ausgesendet, allerdings zeigen verschiedene Arbeiten wie [26] oder [43] auf empirische Weise, dass ein maximaler zeitlicher Abstand von 1-2 Minuten zwischen den meisten Signalfolgen vorliegt. Die vergangene Zeit spielt also für die Sendewahrscheinlichkeit eine entscheidende Rolle. Zudem steigt mit ihr die Wahrscheinlichkeit für eine Nutzeraktivität.
3. **Gerät:** Das Sendeverhalten hängt ebenfalls von der verwendeten Hardware und dem Betriebssystem ab. So messen Goodall et al. [70] für unterschiedliche Geräte ein durchschnittliches Intervall zwischen 18 und 82 Sekunden. Zusätzlich legen eigene Untersuchungen [186] eine doppelt so hohe Probe-Frequenz für ein iOS-Gerät im Vergleich zu einem Android-Smartphone nahe.

Unter Berücksichtigung dieser drei Faktoren, wird die Sendewahrscheinlichkeit  $p_{\text{tx}}$  auf Basis von  $p_{\text{activity}}$  und  $p_{\text{device}}$  bestimmt. Dabei steht  $p_{\text{activity}}$  für die Wahrscheinlichkeit, dass eine Nutzeraktivität auftritt, während  $p_{\text{device}}$  die

Wahrscheinlichkeit modelliert, dass ein bestimmtes Gerät einen IEEE 802.11 aktiven Scan durchführt. Die Wahrscheinlichkeiten werden wie folgt berechnet:

$$p_{\text{activity}} = t_{\text{activity}} \times \frac{1}{c_{\text{activity}}} \quad (3.1)$$

$$p_{\text{device}} = F_X(t_{\text{device}}) \quad (3.2)$$

Hierbei stellt  $t_{\text{activity}}$  und  $t_{\text{device}}$  jeweils die Zeit dar, die seit der letzten Nutzeraktivität bzw. seit der letzten gerätespezifischen Aussendung von Probe Requests vergangen ist. Nach Gleichung 3.1 steigt die Wahrscheinlichkeit für  $p_{\text{activity}}$  mit der Zeit linear, wobei  $c_{\text{activity}} > 0$  eine Konstante repräsentiert, welche die Steigung festlegt. Gleichung 3.2 hingegen modelliert  $p_{\text{device}}$  als kumulative WDF  $F_X$  mit  $X \sim \mathcal{N}(\mu, \sigma^2)$ . Dadurch wird ein normalverteiltes Sendeverhalten des Geräts  $X$  modelliert, welches mit dem durchschnittlichen Intervall  $\mu$  und einer Standardabweichung von  $\sigma$  Probe Requests aussendet. Die Parameter können für beliebige Geräte empirisch bestimmt werden. In Anlehnung an [70] wird  $\mu = 65$  und  $\sigma = 20$  Sekunden angenommen.

Während der Simulation wird  $t_{\text{activity}}$  und  $t_{\text{device}}$  kontinuierlich bestimmt. In jedem Simulationsschritt wird für alle Geräte  $p_{\text{activity}}$  und  $p_{\text{device}}$  berechnet und anhand dieser entschieden, ob Probe Requests versendet werden oder nicht. Im positiven Fall wird über das Pfadverlustmodell aus Formel 2.1 eine WLAN-Signalausbreitung auf Basis der aktuellen Position und des zugrunde liegenden Umgebungsmodells simuliert. Daraus lässt sich die Empfangssignalstärke für jeden beliebigen Ort im Koordinatensystem bestimmen. Um realistischerweise die auftretenden Fluktuationen im RSS-Wert zu simulieren, wird zusätzlich über eine angenommene Normalverteilung mit  $\mathcal{N}(0, 4)$  ein zufälliges Rauschen erzeugt und zur errechneten Empfangssignalstärke hinzuaddiert. Da in realen Umgebungen zudem Fehler bei der drahtlosen Kommunikation auftreten, wird mit einer Wahrscheinlichkeit von 25% die simulierte Übertragung verworfen. Es sei angemerkt, dass dieser Wert nur ganz grob den tatsächlichen Paketverlust beschreiben kann, da in realen Umgebungen die Verluste zwischen 10% und 60% liegen [190]. Für die Simulation wird eine konstante Verlustrate aber als ausreichend angesehen.

Bei erfolgreicher Übermittlung an eine Messstation  $m$  speichert diese das entsprechende Beobachtungstupel  $o_m := \{t, \text{MAC}, \text{RSS}\}$ . Aufgrund von Verzögerungen bei der Übertragung und der Tatsache, dass reale Probe Requests in *Bursts* versendet werden, wird zur Bestimmung von  $o_m$  nicht eine einzelne Beobachtung zu einem bestimmten Zeitpunkt betrachtet. Stattdessen werden die Signale einer MAC-Adresse, die innerhalb der kurzen Zeitspanne  $\Delta t$  aufgezeichnet werden, zu einer Beobachtung zusammengefasst und der Durchschnitt aller Empfangssignalstärken als finaler RSS-Wert verwendet. Sind mehrere  $n$  Messstationen vorhanden, können die einzelnen Beobachtungen zu einem Beobachtungsvektor  $\vec{v}_{t, \text{MAC}} = \{o_1, o_2, \dots, o_n\}$  aggregiert werden. Hat eine Messstation für die Zeitspanne keine entsprechende Beobachtung gemacht, lässt

sich der fehlende Wert in Anlehnung an [104] mit -100 dBm als untere Schranke komplettieren. Dies ist wichtig, wenn Beobachtungsvektoren gleicher Länge verglichen werden sollen. Insgesamt ist  $\vec{v}_{t,MAC}$  damit genauso aufgebaut wie Beobachtungsvektoren aus einer realen Umgebung, wodurch sich die Simulationsumgebung anhand tatsächlicher WLAN-Aufzeichnungen validieren lässt.

### 3.2.4 Modellvalidierung anhand realer Daten

Zur Bewertung der Realitätstreue wird im Folgenden eine ergebnisbezogene Modellvalidierung der vorgestellten Simulationsumgebung durchgeführt. Hierzu werden auf Basis einer äquivalenten Versuchsanordnung simulierte und tatsächliche WLAN-Beobachtungen verglichen und die Ergebnisse anschließend diskutiert. Abbildung 3.7 zeigt schematisch den entsprechenden Aufbau eines realen Experiments, welches im Simulator identisch abgebildet und durchgeführt wird. Dabei überwachen fünf WLAN-Monitore den dargestellten

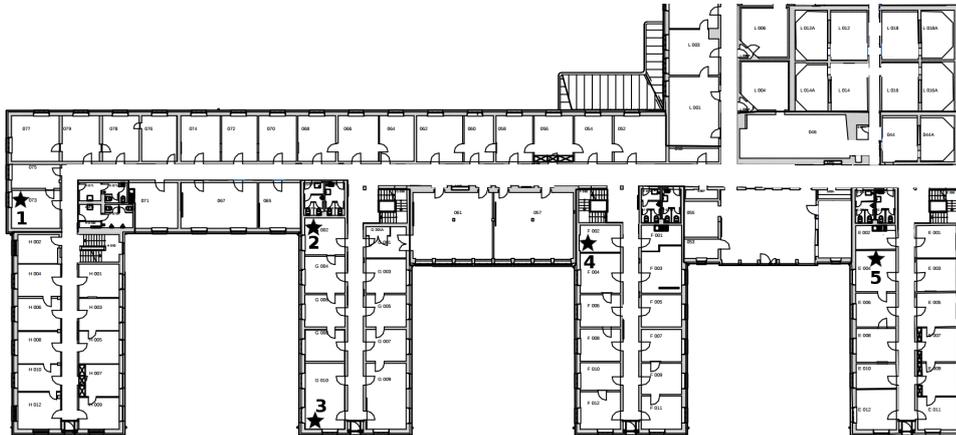


Abbildung 3.7: Versuchsanordnung zur Modellvalidierung: Fünf WLAN-Monitore, symbolisiert durch nummerierte Sterne, zeichnen Probe Requests im dargestellten Gebäudebereich auf.

Abschnitt eines Universitätsgebäudes und zeichnen für einen Werktag Probe Requests von mobilen Teilnehmern auf. Ein Beobachtungsvektor  $\vec{v}_{t,MAC}$  aggregiert dabei die fünf Beobachtungstupel, die während einer Zeitspanne von  $\Delta t = 5$  Sekunden registriert wurden. Fehlende Werte werden darin (wie zuvor angedeutet) mit -100 dBm ergänzt.

Mithilfe der realen Versuchsanordnung konnten insgesamt 1295 eindeutige MAC-Adressen erfasst werden. Davon waren 468 nur sehr kurz im überwachten Bereich und wurden weniger als fünfmal beobachtet, weshalb diese für die nachfolgenden Untersuchungen ignoriert werden. Betrachtet wird also eine Stichprobe für die reale Umgebung von  $n_r = 827$  Geräten, die sich an diesem Tag für eine unterschiedlich lange Zeitspanne irgendwo im entsprechenden Gebäudeabschnitt aufgehalten haben. Außer den gespeicherten Beobach-

tungsvektoren ist über die Einzelnutzer nichts weiter bekannt, weshalb genaue Trajektorien vorerst nicht nachgebildet werden können.

Bei der Simulation wird deshalb eine identische Stichprobe von  $n_s = n_r = 827$  Nutzer erzeugt, die sich über das Bewegungsmodell aus Unterkapitel 3.2.2 mit konstanter Geschwindigkeit von  $1,2\text{ m/s}$  durch den dargestellten Gebäudeabschnitt bewegen. Dabei wählt jeder Nutzer zufällig ein bis vier Ziele aus und verweilt dort jeweils zwischen einer halben Minute und maximal zwei Stunden. Betreten wird das Gebäude zwischen 7:00 und 18:00 Uhr. Für das Aussendungsmodell wird die generische Verteilung des Sendeverhaltens mit  $\mu = 65$  und  $\sigma = 20$  approximiert (vgl. Abschnitt 3.2.3). Außerdem wird  $c_{\text{activity}} = 3600$  verwendet, womit mindestens eine Nutzeraktivität pro Stunde angenommen wird. Es sei an dieser Stelle darauf hingewiesen, dass die Simulation die Realität nur vereinfacht bzw. abstrakt wiedergeben kann. Daher wird nicht das komplexe Verhalten einzelner Personen in realen Umgebungen abgebildet. Stattdessen wird ein möglichst generisches, aber plausibles Nutzerverhalten auf Basis des subjektiven Wissens über die vorliegende Domäne simuliert. Typische Effekte der Modellvalidierung wie Überanpassung oder Einzelfallprüfungen werden dadurch vermieden [29]. Allerdings lässt sich der Verhaltensunterschied nur relativ für das Gesamtsystem und nicht für Einzelobjekte messen, was bei den folgenden Ergebnissen berücksichtigt werden muss.

Auf Basis der beiden Stichproben  $n_r$  und  $n_s$  wird zunächst ein grafischer Vergleich der jeweiligen Beobachtungshäufigkeit durchgeführt, der in Abbildung 3.8 dargestellt ist. Optimal wäre es, wenn simulierte und reale Daten hierbei identische Verteilungen zeigen. Insgesamt wird jedoch deutlich, dass die Beobachtungsfrequenz im Simulator etwas höher liegt als bei den realen Daten. So stellt Abbildung 3.8a die Verteilung der Beobachtungen pro erfasster MAC-Adresse als Histogramm dar. Dabei überlappen sich zwar die Häufigkeiten bzgl. der geringsten Frequenz, allerdings sind bis zur Grenze von 135 Beobachtungen pro Nutzer die realen Messungen häufiger vertreten und danach dominieren die simulierte Werte. Dies hat zwei mögliche Gründe: Zum einen ist das tatsächliche Nutzerverhalten komplexer als im Simulator dargestellt, wodurch es in realen Umgebungen auch häufiger zu kurzen bzw. vereinzelt Randbeobachtungen kommt. Zum anderen sind die Parameter im Aussendungsmodell mit  $\mu = 65$  und  $\sigma = 20$  u.U. zu klein gewählt.

Um das Sendeverhalten unabhängig von der Nutzeraktivität zu untersuchen, wird in Abbildung 3.8b die Verteilung der Beobachtungsfrequenz in Abhängigkeit der Zeit betrachtet. Dabei zeigen die beiden Boxplots insgesamt die gleiche Tendenz wie zuvor, nämlich eine geringere Frequenz im Fall von realen Beobachtungen. Allerdings weisen diese Messungen auch deutlich mehr Ausreißer nach oben und eine größere Streuung der Daten auf. Dies veranschaulicht einmal mehr das nicht-deterministische Probe-Verhalten mobiler Geräte in realen Umgebungen, was sich schwer durch ein Modell vollständig beschreiben lässt. In Hinblick auf die beiden Mittelwerte lässt sich jedoch feststellen, dass die Simulation im Gesamtdurchschnitt lediglich um 43,8 Sekunden abweicht, was

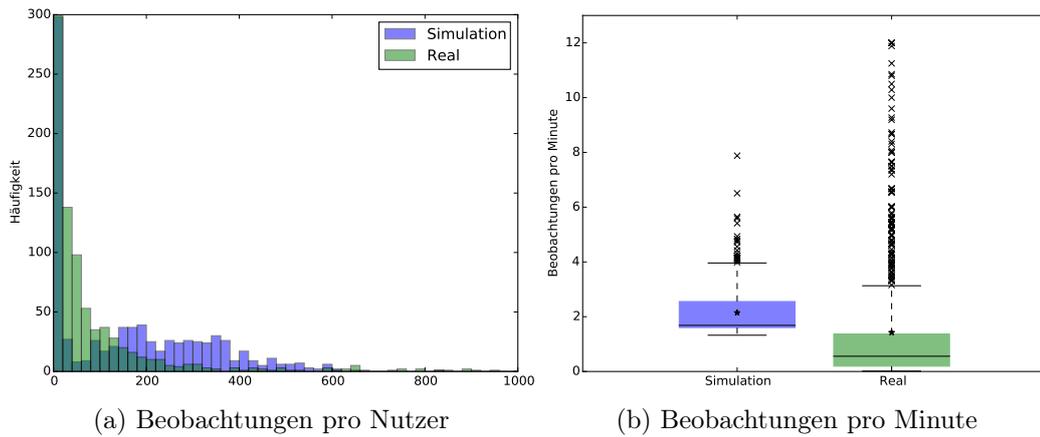


Abbildung 3.8: Grafischer Vergleich zur Modellvalidierung unter Berücksichtigung der Beobachtungshäufigkeit.

Monitor nr.	1	2	3	4	5
Sim. (Min/Max)	-99/-73	-99/-67	-99/-81	-99/-67	-98/-68
Real (Min/Max)	-99/-95	-99/-85	-99/-87	-99/-76	-98/-78
Sim. ( $\mu$ )	-93,75	-91,33	-94,76	-89,65	-90,72
Real ( $\mu$ )	-99,09	-94,71	-95,90	-91,62	-92,17
Sim. ( $\sigma$ )	6,95	8,69	5,00	8,52	8,16
Real ( $\sigma$ )	0,96	4,02	3,27	5,85	5,26

Tabelle 3.3: Vergleich der durchschnittlichen RSS-Werte in dBm.

im Rahmen der Untersuchung einen akzeptablen Wert darstellt. Durch Erhöhung von  $\mu$  und  $\sigma$  kann das simulierte Verhalten auch in Bezug auf die hier beobachtete Verteilung angepasst werden, wobei zur Überprüfung der Generalisierbarkeit weitere Untersuchungen nötig sind.

Im nächsten Schritt sollen zur Modellvalidierung die simulierten Empfangssignalstärken mit den tatsächlich gemessenen RSS-Werten verglichen werden. Die Tabelle 3.3 stellt hierzu die durchschnittlichen Minima, Maxima sowie den Gesamtdurchschnitt  $\mu$  und die mittlere Standardabweichung  $\sigma$  der entsprechenden RSS-Werte dar, die an den fünf WLAN-Monitoren gemessen wurden. In Bezug auf die durchschnittlichen Minimalwerte zeigt der Simulator die selben Resultate wie das reale Experiment. Dies ist einfach zu erklären, da in beiden Szenarien der beobachtbare Bereich von Personen verlassen wird, wobei die geringstmögliche Signalstärke durch die untere Schranke für fehlende Werte bei -100 dBm liegt. Interessanter sind dagegen die beobachteten Mittel- und Maximalwerte. Hierbei zeigt sich, dass der Simulator konsequent stärkere Signale erfasst. Diese Abweichung kann auf die Parameterwahl beim Pfadverlustmodell (vgl. Formel 2.1) zurückgeführt werden, welche sich durch eine entsprechende Kalibrierung des Modells ausgleichen lässt. Vor allem für die

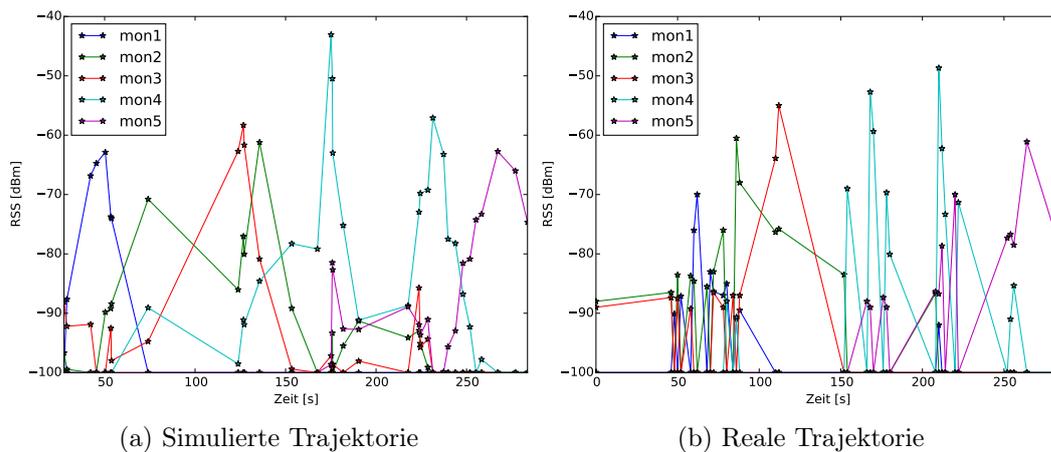


Abbildung 3.9: Modellvalidierung am Beispiel einer Einzeltrajektorie.

beiden ersten Monitore ist die Abweichung am größten, was sich auch in der mittleren Standardabweichung bemerkbar macht. Hier liegt der Hauptgrund darin, dass der linke Gebäudeflügel, der von diesen Sensoren überwacht wird, in der Realität weniger stark frequentiert ist, während der Simulator aufgrund der zufälligen Wahl von Zielen diesbezüglich keinen Unterschied macht. Daher fällt die mittlere Standardabweichung im Fall von Monitor 1 für reale Beobachtungen auch am geringsten aus. Betrachtet man wieder den Gesamtdurchschnitt, so bleiben die Abweichungen zwischen simulierten und realen Daten für die restlichen Monitore im vertretbaren Rahmen von weniger als 3,5 dBm. Dadurch zeigt sich die prinzipielle Validität des Modells auch in Hinblick auf die gemessenen Empfangssignalstärken.

Bei den bisherigen Vergleichen wurde immer das Gesamtsystem und nicht einzelne Trajektorien berücksichtigt, da für die Gesamtdaten keine valide Grundwahrheit vorhanden ist. Um dennoch eine feingranularere Bewertung über die Realitätstreue der Simulation auf Pfadenebene diskutieren zu können, wurden vereinzelte Wege durch das Gebäude gezielt abgelaufen, wobei die Testperson vier verschiedene Smartphones auf unterschiedliche Weise genutzt hat. Interessant ist vor allem die Situation, in der das mobile Gerät ohne aktive Benutzung in der Hosentasche mitgeführt wurde, da diese als eine der häufigsten Situationen angenommen wird. Bei den Testläufen wurde die Grundwahrheit aufgezeichnet. So können die entsprechenden Einzeltrajektorien im Simulator äquivalent nachgebildet und die resultierenden Beobachtungen verglichen werden. Abbildung 3.9 zeigt dazu den zeitlichen Verlauf der simulierten und realen Empfangssignalstärken, die während eines Testlaufs an den fünf WLAN-Monitoren beobachtet wurden. Die Trajektorie verlief dabei vom linken unteren Bereich des Gebäudes über alle Flügel bis hin zum rechten unteren Gang.

Auch wenn die Kurvenverläufe nicht komplett identisch sind, weisen sie gemeinsame Tendenzen auf. Zum einen unterscheiden sich die benötigten Zeiten für den Ablauf des kompletten Bewegungspfades lediglich um vier Sekunden.

Allerdings ist anzumerken, dass sich sowohl die reale als auch die simulierte Testperson mit konstanter Geschwindigkeit fortbewegt, was nicht immer der Fall ist. Zum anderen weisen die lokalen Maxima in den meisten Fällen eine ähnliche zeitliche Reihenfolge und vergleichbare RSS-Werte auf. Auch die absolute Anzahl an Datenpunkten ist mit einer geringen Abweichung von drei Beobachtungen nahezu gleich. Größere Unterschiede sind hingegen im Bereich zwischen 50 und 100 Sekunden zu entdecken, in welchem eine erhöhte Frequenz bei den realen Beobachtungen festzustellen ist. Dies verdeutlicht die Schwierigkeit der Vorhersage des tatsächlichen Probe-Verhaltens mobiler Geräte.

Für eine objektive Bewertung der Realitätstreue bietet sich ein Vergleich von Konfidenzintervallen an. Dazu wird im Folgenden das Verfahren der gepaarten Stichproben (*Paired t-Konfidenzintervalle*) verwendet, da hierbei keine Annahmen bzgl. gleicher Varianz bzw. der Unabhängigkeit der Stichproben gefordert wird. Allerdings müssen die Beobachtungszahlen der Stichproben identisch sein [29]. Aus diesem Grund werden die Mittelwerte der realen und simulierten Empfangssignalstärken  $v_{R_m}$  und  $v_{S_m}$  als Stichprobe verwendet, die von jedem der  $n = 5$  Monitore während des Ablaufs des Bewegungspfads gemessen werden. Damit lässt sich die mittlere Abweichung der Stichprobe  $\hat{\mu}_{RS} = \hat{\mu}_R - \hat{\mu}_S$  mit  $\hat{\mu}_R = \frac{1}{n} \sum_{m=1}^n v_{R_m}$  und  $\hat{\mu}_S = \frac{1}{n} \sum_{m=1}^n v_{S_m}$  berechnen. Die Standardabweichung der Stichprobe  $s$  ergibt sich daraus wie folgt:

$$s_{RS} = \sqrt{\frac{\sum_{m=1}^n ((v_{R_m} - v_{S_m}) - \hat{\mu}_{RS})^2}{n(n-1)}}$$

Mittels  $\hat{\mu}_{RS}$  und  $s_{RS}$  kann das Konfidenzintervall bestimmt werden, welches die Lage eines gesuchten Parameters für das Gesamtsystem mit einer gewissen Häufung (dem Konfidenzniveau) beschreibt. Im vorliegenden Fall wird die durchschnittliche Abweichung zwischen simulierten und realen Beobachtungen gesucht, die im Idealfall null ist. Daher muss untersucht werden, ob das Konfidenzintervall für die Differenz der Mittelwerte auch die Null beinhaltet. Für die in Abbildung 3.9 dargestellte Trajektorie ist das der Fall. Hierbei ergibt sich ein Konfidenzintervall von  $[-4,5; 1,5]$  bei einem Konfidenzniveau von 95%, was ein akzeptables Ergebnis für die Abschätzung darstellt. Somit konnte die Validität der vorgestellten Simulationsumgebung nochmals bestätigt werden.

### 3.2.5 Zusammenfassung und Diskussion

Zusammenfassend lässt sich festhalten, dass reale WLAN-Beobachtungen im Durchschnitt einer geringeren Frequenz unterliegen als hier im Modell angenommen. Allerdings treten gleichzeitig bei den realen Daten auch vermehrt obere Ausreißer auf. Dieses schwer vorhersagbare Probe-Verhalten wird durch das vorgestellte Aussendungsmodell im Rahmen eines akzeptablen Verhaltensunterschieds approximiert. Bei der durchgeführten Modellvalidierung zeigte sich, dass der Gesamtdurchschnitt für die Beobachtungsfrequenz um 43,8 Se-

kunden abweicht. Zudem wurde die Empfangssignalstärke konsequent überschätzt, was durch eine entsprechende Anpassung des Pfadverlustmodells korrigiert werden kann. Die Mittelwerte weisen hingegen nur einen geringen Unterschied von maximal 3,38 dBm bei vier von fünf Monitoren auf. Insgesamt zeigt sich somit die prinzipielle Validität der verwendeten Simulationsumgebung, die auch für einzelne Trajektorien bestätigt werden konnte. Eine genauere Anpassung an das jeweilige Anwendungsszenario lässt sich durch eine Kalibrierung der Modellparameter erreichen, wofür weitere Untersuchungen notwendig sind.

## 3.3 Zusammenfassung

In diesem Kapitel wurde zunächst untersucht, wie und mit welcher Genauigkeit Signalbeobachtungen von Bluetooth- bzw. WLAN-aktiven Geräten erfasst werden können, um tatsächliche Nutzerströme zu analysieren. Hierzu wurden mögliche Ansätze zur Bestimmung von Personendichte und Nutzerströmen vorgestellt, die erstmals unter realen Bedingungen mit einer bekannten Grundwahrheit evaluiert wurden. Dabei stellte sich heraus, dass ein einfaches Zählen der beobachteten MAC-Adressen, wie es in einigen verwandten Arbeiten propagiert wird, nur einen moderaten Zusammenhang mit den tatsächlichen Nutzerbewegungen liefert. Zudem konnten sowohl Bluetooth- als auch WLAN-Beobachtungen verwendet werden, um generelle Informationen über die Nutzer zu gewinnen. Es zeigte sich allerdings, dass WLAN bei allen Versuchen genauere Ergebnisse aufgrund höherer Erkennungsraten liefert. Durch zusätzliche Hintergrundinformationen und die Berücksichtigung von Zeit und Empfangssignalstärken konnte ein stark korrelierter Zusammenhang zwischen den Berechnungen und der Grundwahrheit festgestellt werden. Insgesamt lässt sich festhalten, dass WLAN bereits heute als Tracking-Technologie geeignet ist, um Tendenzen über reale Nutzerbewegungen zu erfassen. Absolute Personenzahlen konnten hingegen nicht genau ermittelt werden. Jedoch liegt die Vermutung nahe, dass die Ergebnisse durch die zunehmende Verbreitung mobiler Geräte in Zukunft noch genauer werden.

Die Erkenntnisse wurden im zweiten Teil genutzt, um eine realitätsnahe Simulationsumgebung für WLAN-Beobachtungen zu entwickeln. Diese ist für weitere Untersuchungen nützlich, da das Problem der fehlenden Grundwahrheit für reale Umgebungen fortlaufend besteht. Zur Umsetzung wurden drei Modelle vorgestellt. Dabei basiert das Umgebungsmodell auf einer Bitmap-Darstellung herkömmlicher Gebäudepläne, während das Bewegungsmodell durch eine Anpassung des Pathway-Mobility-Models realisiert wird. Das Aussendungsmodell wurde neu entwickelt und berücksichtigt erstmals neben der Zeit auch die Aktivität sowie das Endgerät des Nutzers. Zur Überprüfung der Realitätsstreue wurde eine ergebnisbezogene Modellvalidierung anhand realer WLAN-Beobachtungen durchgeführt. Dabei zeigten sich geringe Verhaltensunterschiede zwischen den simulierten und den realen Aufzeichnungen, weshalb die entwickelte Simulationsumgebung als valide betrachtet wird.

## 4 Extraktion der Ortsinformation zur Nutzerstromanalyse

Nachdem im vorangegangenen Kapitel die Datenerfassung im Fokus stand, beschäftigt sich der vorliegende Abschnitt mit der Extraktion von Ortsinformationen aus diesen Daten und der damit verbundenen Ermittlung von Bewegungspfaden. Wie bereits erwähnt, müssen für eine kontinuierliche und flächendeckende Standortverfolgung mehrere Messstationen geeignet verteilt werden, um mögliche Signale von mobilen Nutzern aufzuzeichnen. Da die Anzahl und die Platzierung der Messknoten einen signifikanten Einfluss auf die Genauigkeit und vor allem auf die Skalierbarkeit des Tracking-Systems haben [32], werden in diesem Kapitel zunächst Strategien entwickelt, um eine annähernd optimale Lösung für das vorliegende Verteilungsproblem zu erreichen. Zur Evaluation wird eine Simulation durchgeführt und die Ergebnisse werden mit einer Brute-Force-Methode verglichen. Die hier beschriebenen Kerninhalte wurden bereits als internationaler Konferenzbeitrag [175] veröffentlicht. In diesem sind auch die Abbildungen 4.1, 4.3, 4.4, 4.5 und 4.6 enthalten. Die inhaltliche Vorarbeit bezüglich der Modifikation des TRACCLUS-Algorithmus wurde als Fachgesprächsbeitrag [183] publiziert. Wie aus Abschnitt 1.3 hervorgeht, stammen die wesentlichen Inhalte wie Idee, Konzeption und Evaluation jeweils vom Autor der vorliegenden Arbeit, wobei [175] komplett in Alleinautorschaft entstand.

Nach der Diskussion des Verteilungsproblems werden in Abschnitt 4.2 probabilistische Ansätze zur Pfadermittlung untersucht und eine neuartige hybride Methode entwickelt, um die Nachteile der vorhandenen Verfahren zu minimieren. Alle Ansätze werden auf Basis anerkannter Metriken aus dem Bereich der Indoor-Positionierung bewertet und miteinander verglichen. Die Hauptinhalte dieses Abschnitts wurden bereits als internationaler Konferenzbeitrag [181] publiziert. Dabei stammt vor allem die grundsätzliche Idee sowie die Durchführung der Experimente und die vollständige Evaluation vom Autor der vorliegenden Arbeit. Die Abbildungen 4.8, 4.9, 4.10, 4.11, 4.12 und 4.13 sind ebenfalls in der Publikation enthalten.

Zum Abschluss des Kapitels wird im Abschnitt 4.3 eine Zusammenfassung der wichtigsten Resultate und Erkenntnisse gegeben.

## 4.1 Strategien zur Platzierung von Messknoten

Für eine optimale Platzierung von Messknoten zur Erfassung von Nutzerströmen existieren prinzipiell zwei konkurrierende Ziele: Zum einen soll die Anzahl der WLAN-Monitore möglichst klein sein, um Installations- und Wartungskosten aber auch die Menge der aufgezeichneten Daten gering zu halten. Zum anderen soll der gewünschte Bereich möglichst komplett und ausreichend überwacht werden, um Nutzerbewegungen vollständig erfassen zu können.

Diese Problematik beschreibt eine Abwandlung des bekannten und NP-schweren Problems der Museumswächter (engl.: *art gallery problem*) [189]. Bei diesem gilt es, eine möglichst geringe Menge an Punkten  $\mathcal{G}$  in einem gegebenen Polygon  $\mathcal{W}$  zu finden, so dass jede Stelle des Polygons in direktem Sichtkontakt zu einem Punkt  $p \in \mathcal{G}$  steht. Laut Eidenbenz et al. [53] gehört das Problem der Museumswächter zur Klasse der sogenannten *APX-Probleme* (APX für engl. *approximable*). Das bedeutet, dass es zwar durch keinen Algorithmus vollständig in Polynomialzeit gelöst werden kann, aber Algorithmen existieren, welche die optimale Lösung bis zu einer bestimmten Schranke genau approximieren können [11]. Bevor nun ein Überblick über derartige Lösungsansätze aus der Literatur gegeben wird, soll zunächst die hier vorliegende Problemstellung konkret formalisiert werden, da sich diese vom klassischen Art-Gallery-Problem unterscheidet.

### 4.1.1 Problemstellung der Messknotenverteilung

Wie bereits erwähnt, treten beim vorliegenden Optimierungsproblem zwei Ziele in Konkurrenz, nämlich die Minimierung der Knotenanzahl und die Maximierung der erfassbaren Trajektorien. Zur Formulierung des Problems muss zunächst geklärt werden, unter welchen Bedingungen eine Trajektorie formal als *erfassbar* angenommen wird. Dazu sei die folgende Definition gegeben:

**Definition 1** (Erfassbare Trajektorie):

Eine Trajektorie  $\tau$  gilt als erfassbar, wenn mindestens ein Punkt  $p \in \tau$  und ein Messknoten  $m$  existieren, so dass  $p$  innerhalb der Beobachtungreichweite  $r_m$  von  $m$  liegt. Formal lässt sich dies ausdrücken als:

$$\exists p \in \tau. \exists m \in M : \delta(p, m) \leq r_m \quad (4.1)$$

Das bedeutet allerdings, dass eine komplette Trajektorie bereits dann als erfassbar gilt, wenn diese in ihrem Gesamtverlauf an irgendeiner Stelle einen beobachtbaren Bereich schneidet. Das ist jedoch für komplexe Bewegungsabläufe innerhalb von Gebäuden wenig zielführend, da die Trajektorie so nicht reproduzierbar ist. Daher scheint es sinnvoll, die Gesamttrajektorie  $\tau$  mittels einer geeigneten Segmentierung zunächst in eine Menge kleinerer Subtrajektorien zu partitionieren und die Definition 1 auf diese anzuwenden.

Nach Lee et al. [121] lassen sich Trajektorien an sogenannten *charakteristischen Punkten* segmentieren. Wie in Abschnitt 2.4.2 erwähnt, gibt es hierfür diverse Strategien. Für den vorliegenden Fall soll es allerdings ausreichen, wenn lediglich die Richtungsänderung für die Segmentierung berücksichtigt wird, da Personen innerhalb von Gebäuden ihre Bewegungsrichtung häufig an strategisch wichtigen Stellen ändern, wie bspw. an Eingängen, Ecken, Türen usw. Aus diesem Grund wird ein charakteristischer Punkt hier wie folgt definiert:

**Definition 2** (Charakteristischer Punkt):

Ein Punkt  $p \in \tau$  gilt als charakteristischer Punkt  $cp$ , wenn sich der Richtungsvektor an der Stelle  $p$  um einen bestimmten Winkel  $\alpha \geq \theta$  ändert.

Dabei ist  $\theta$  ein zu bestimmender Schwellwert, der je nach Gebäude verschieden sein kann. Auf Basis der charakteristischen Punkte lassen sich nun Subtrajektorien bestimmen:

**Definition 3** (Subtrajektorie):

Eine Subtrajektorie  $v \subseteq \tau$  repräsentiert eine Partition der Gesamttrajektorie  $\tau$  und stellt ein Liniensegment dar, welches durch zwei aufeinanderfolgende charakteristische Punkte beschrieben wird.

Sind nun alle Subtrajektorien nach Definition 1 erfassbar, kann auch der gesamte Bewegungsablauf bei entsprechenden Signalbeobachtungen genauer reproduziert werden. Zur besseren Veranschaulichung stellt Abbildung 4.1 diese Idee nochmals graphisch an einem Beispiel dar. Dabei verläuft die Gesamttrajektorie vom Startpunkt  $S$  zum Ziel  $Z$  und wird an den Ecken segmentiert. Die ersten drei Segmente schneiden den überwachten Bereich eines Messknotens und gelten deshalb als erfassbar, während die letzte Subtrajektorie nicht beobachtet werden kann. Zudem ist der mittlere Messknoten in dem gezeigten Beispiel überflüssig und könnte bei einer Optimierung entfallen, da er nur eine Subtrajektorie beobachtet, die bereits als erfassbar gilt. Die Beobachtungreichweiten sind hier vereinfacht als Kreise dargestellt. In Wahrheit wird die Reichweite eines WLAN-Monitors nicht nur durch die Distanz, sondern auch durch Wände und andere Hindernisse beschränkt, was sich durch das Pfadverlustmodell (vgl. Formel 2.1) beschreiben lässt. Dadurch ergibt sich auch der Unterschied zum klassischen Art-Gallery-Problem: Anstatt von einem Museumswächter auszugehen, der alle Punkte in einem freien Sichtfeld unendlich weit *sehen* kann, *hört* ein Messknoten in seinem Umfeld auf ankommende Signale. Dabei ist zwar die Reichweite beschränkt, allerdings können auch Punkte überwacht werden, die sich hinter Wänden oder Hindernissen befinden. Auf Basis dieser Überlegungen wird in Anlehnung an [69] ein erweitertes Art-Gallery-Problem definiert, welches die vorliegende Problemstellung folgendermaßen beschreibt:

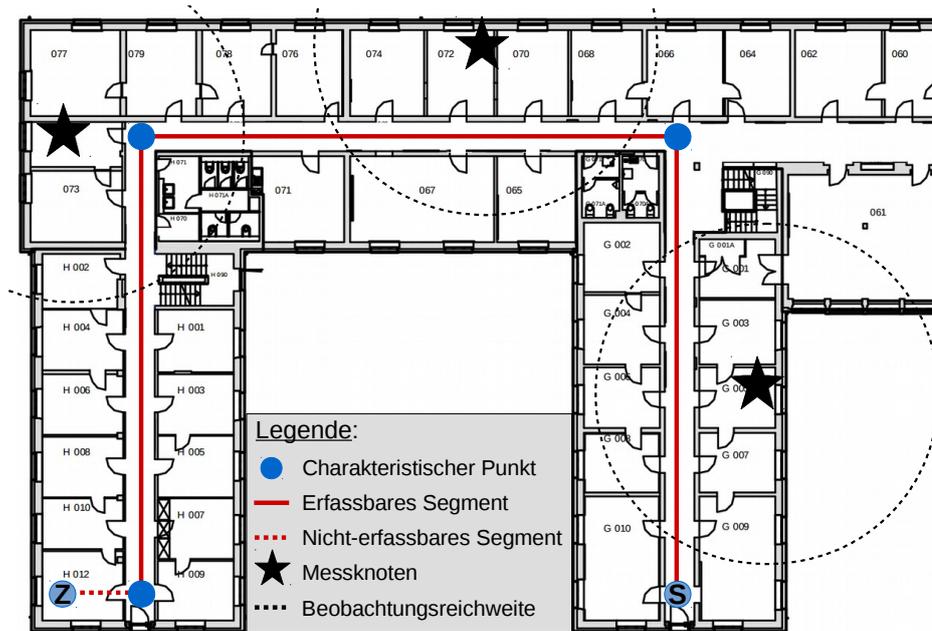


Abbildung 4.1: Beispiel für einen Bewegungspfad durch ein Gebäude mit erfassbaren und nicht-erfassbaren Subtrajektorien.

**Problem 1** (Erweitertes Art-Gallery-Problem):

Gegeben sei ein Layout  $\mathcal{W} \subset \mathbb{R}^2$  und eine Menge an Trajektorien  $T$ . Finde eine minimale Anzahl an Positionen  $\mathcal{G} = \{g_{m_1}, g_{m_2}, \dots, g_{m_n}\}$  in  $\mathcal{W}$ , so dass die maximale Anzahl vorhandener Subtrajektorien von mindestens einer Position in  $\mathcal{G}$  nach Definition 1 erfassbar ist.

Zur Lösung des Problems liegt die Idee zugrunde, dass Messknoten nur an den Stellen platziert werden dürfen, an denen die meisten Personen vorbeikommen bzw. von denen aus eine erhöhte Anzahl an Subtrajektorien erfasst werden kann. Wie bereits erwähnt, werden solche Orte im Rahmen des Trajectory Computings häufig als *Hotspots* bezeichnet, die in verschiedenen Arbeiten aus Trajektorien-Daten gewonnen werden (vgl. Abschnitt 2.4.5). In diesem Kapitel werden drei weitere Verfahren zur Erkennung von Hotspots entwickelt und anhand der beschriebenen Problemstellung evaluiert. Zunächst soll jedoch ein Überblick über existierende Ansätze gegeben werden, die sich allgemein mit Platzierungsstrategien von Mess- bzw. Sensorknoten auseinandersetzen.

### 4.1.2 Existierende Ansätze von Platzierungsstrategien

Die Suche nach einer möglichst optimalen Platzierung von Sensorknoten ist aufgrund der NP-Schwere des Problems nach wie vor herausfordernd. So existieren in der Literatur diverse Ansätze, die das Problem auf unterschiedliche Weise angehen. Einen allgemeinen und umfassenden Überblick über Platzie-

rungsstrategien im Bereich der drahtlosen Sensornetze (WSN für engl. *Wireless Sensor Networks*) liefert das Survey von Younis und Akkaya [237]. Die Autoren unterteilen dabei die untersuchten Strategien in statische und dynamische Ansätze. Dabei sind Letztere eher für klassische WSNs mit vielen Knoten geeignet, um auf dynamische Ereignisse wie bspw. einem Energieausfall reagieren zu können. Im Fall von WLAN-Monitoren, die innerhalb eines Gebäudes fest installiert werden, spielen solche Aspekte hingegen keine Rolle, weshalb hier der Fokus auf den statischen Ansätzen liegt. Diese identifizieren eine geeignete Platzierung vor dem Netzbetrieb mithilfe statischer Metriken wie bspw. Anzahl, Abstände oder Reichweite der einzusetzenden Knoten. Dabei kann die prinzipielle Ortswahl über eine zufällige Ziehung aus einer angenommenen Verteilung [93] oder auch kontrolliert [28] bzw. deterministisch anhand bestimmter Kriterien [149] ablaufen.

Gerade innerhalb von Gebäuden spielt die deterministische Ortswahl eine große Rolle, da die Menge der möglichen Monitorpositionen oftmals durch bauliche Eigenschaften vorgegeben ist. Hier liegt häufig ein kombinatorisches Optimierungsproblem vor, welches ebenfalls NP-schwer ist und auf verschiedene Weise thematisiert wird. So gehen bspw. Krause et al. [112,113] in ihren Arbeiten von einer gegebenen endlichen Menge möglicher Messknotenpositionen aus, von denen die geeignetsten  $k$  Plätze anhand einer submodularen Funktion oder anhand eines gewählten Qualitätskriteriums ermittelt werden. Ein anderer kombinatorischer Ansatz zur Platzierung von Bluetooth-Sensoren (*Beacons*) für die Indoor-Lokalisierung wird von Chawathe [34] verfolgt. Hierbei werden keine festen Annahmen über die Beobachtungreichweiten der einzelnen Sensoren getroffen. Die möglichen Beacon-Positionen bilden auf Basis einer willkürlich gewählten Reichweite einen Hypergraphen. Dieser wird durch einen einfachen Greedy-Algorithmus auf einen Subgraphen mit weniger, aber bedeutenderen Hyperkanten reduziert. Dadurch wird zwar nicht das optimale Ergebnis, aber zumindest eine verbesserte Platzierung im Sinne des gewählten Qualitätskriteriums erreicht.

Auch Farkas et al. [58] untersuchen eine Platzierungsstrategie zur Indoor-Positionierung über Lateration. Die Autoren geben mögliche Positionen von WLAN Access Points mittels eines definierten Rasters vor und identifizieren die geeignetsten Kandidaten anhand eines Ansatzes zur simulierten Abkühlung (engl.: *simulated annealing*). Dabei wird die Anzahl aller auf dem Raster verteilten Access Points soweit reduziert, bis vorhandene Redundanzen minimiert sind und gleichzeitig jeder Punkt auf der Karte mindestens drei Access Points über ein Ausbreitungsmodell erreichen kann. Der Ansatz wird mithilfe einer Brute-Force Methode evaluiert, die zwar zum Teil bessere Ergebnisse liefert, allerdings auch eine deutlich längere Laufzeit hat.

Rajagopal et al. [168] suchen nach der optimalen Platzierung von Beacons zur Indoor-Lokalisierung auf Basis von Signalreichweiten. Dabei argumentieren die Autoren, dass zwei Beacons ausreichend sind, wenn die Topologie des Gebäudes bekannt ist. Ähnlich wie zuvor werden bei dem Ansatz zunächst

alle möglichen Beacon-Positionen untersucht. Dabei wählt der Algorithmus die Besten anhand einer speziell eingeführten Metrik aus, so dass letztendlich überall im Gebäude eine Lokalisierung unter den gewählten Anforderungen durchgeführt werden kann. Ein komplexeres Verfahren wird von Leune et al. [124] vorgestellt. Auch hier soll eine optimale Platzierung von Knoten für eine Laterations-basierte Indoor-Positionierung realisiert werden. Mittels physikalischer Signalverfolgung (engl.: *raytracing*) und einem genetischen Algorithmus erreichen die Autoren, dass die Anzahl der Messknoten und die durchschnittliche quadratische Abweichung in der Positionsbestimmung reduziert wird. Dieser Ansatz wird ebenfalls mithilfe einer Brute-Force-Methode verglichen.

Die eben erwähnten Ansätze zeigen nur einen kleinen Teil der vorhandenen Arbeiten, die sich mit Optimierungsstrategien zur Platzierung von Sensorknoten beschäftigen. Insgesamt hängen solche Verfahren und deren Bewertung immer vom jeweiligen Einsatzszenario, dem Optimierungsziel und vom gewählten Qualitätskriterium ab. Nach Kenntnisstand des Autors der vorliegenden Arbeit existiert allerdings noch kein Verfahren, das eine Platzierung von Messknoten hinsichtlich der Erfassung räumlicher Trajektorien in Anlehnung an die beschriebene Problemstellung aus Abschnitt 4.1.1 optimiert. Diese Lücke soll mithilfe der folgenden Beiträge geschlossen werden.

### 4.1.3 Beiträge zur Erkennung von Hotspots

Es wird nun das Ziel verfolgt, stark frequentierte Orte zu identifizieren, um eine effiziente Platzierung von Messknoten zur Erfassung von Trajektorien zu realisieren. Hierzu werden im Folgenden drei Ansätze vorgestellt, welche die geometrischen Eigenschaften vorhandener Bewegungspfade berücksichtigen.

#### Raster-basierter Ansatz

Der Raster-basierte Ansatz ist eine relativ einfache Methode und wird deshalb in vielen verwandten Arbeiten auf ähnliche Weise verwendet [32, 50, 58]. Er soll hier als Benchmark für die nachfolgenden Verfahren dienen. Dabei wird allgemein der betrachtete Bereich auf Basis eines festgelegten Rasters in Zellen unterteilt. Die geeignetsten Zellen zur Platzierung von Messknoten werden dann anhand eines Qualitätskriteriums ausgewählt. Die Zellengröße hat dabei einen direkten Einfluss auf das Ergebnis und fällt je nach Anwendungsszenario unterschiedlich aus.

Im vorliegenden Fall wird ein Raster aus quadratischen Zellen mit einer konstanten Seitenlänge von  $a = r_m \cdot \sqrt{2}$  gewählt, so dass jede Zelle mindestens im Umkreis der Beobachtungsreichweite  $r_m$  eines WLAN-Monitors  $m$  liegt. Dadurch kann jeder Punkt des Gebäudes im Bereich des Rasters erfasst werden, wenn ein Monitor pro Zellenmittelpunkt installiert wird. Dies stellt gleichzeitig den möglichen Lösungsraum dar. Um daraus die geeignetsten Stellen zu ermitteln, muss jede Zelle hinsichtlich der Frequenz an Trajektorien verglichen

werden, welche durch den entsprechenden Zellenbereich verlaufen. Zur Umsetzung werden in einer Bitmap des Gebäudeplans die Werte der Pixel, die von einer Trajektorie durchlaufen wurden, pro Durchlauf um 1 inkrementiert und gespeichert. Dies erfordert allerdings eine genaue Kenntnis über den Verlauf der Nutzerbewegungen, weshalb später zu Evaluationszwecken simulierte Trajektorien verwendet werden. Das Qualitätskriterium zur Auswahl geeigneter Zellen basiert demnach auf der akkumulierten Pixelsumme  $\Sigma_s$  der Zelle  $s$ . Der entsprechende Zellenmittelpunkt  $c_s$  wird dann als Hotspot deklariert, wenn gilt:  $\Sigma_s \geq \varepsilon \cdot \arg \max_{s \in S} \{\Sigma_s\}$ . Dabei sei  $S$  die Menge aller Zellen und  $\varepsilon$  stellt einen Schwellwert dar, der den erforderlichen Anteil an der Maximalsumme festlegt. Je nach Anforderung oder dem gewünschten Optimierungsziel kann  $\varepsilon$  entsprechend gewählt werden.

Es sei angemerkt, dass dieser Ansatz aufgrund seiner Einfachheit den Verlauf von Nutzerpfaden bei der Identifizierung von Hotspots nicht berücksichtigt. Stattdessen wird lediglich auf Basis der festgelegten Parameter und der Häufigkeitsverteilung über den Zellen eine Lösung zur Platzierung von Messknoten zurückgeliefert. Auch ist die Menge des Lösungsraums aufgrund des gewählten Rasters unflexibel und beschränkt sich auf die Mittelpunkte der Zellen. Daher findet das Verfahren zwar stark frequentierte Bereiche, was für eine erste Analyse hilfreich sein kann, allerdings wird die Erfassbarkeit von Subtrajektorien nicht zwangsläufig optimiert. So werden bspw. Zellen mit vielen kleinen Liniensegmenten häufig vernachlässigt, während Bereiche mit wenigen aber längeren Subtrajektorien eher als Hotspots identifiziert werden. Um derartige Probleme zu umgehen, werden nun zwei weitere geometrische Ansätze vorgestellt.

### Dichte-basierter Ansatz

Dieser Ansatz nutzt den DBSCAN-Algorithmus [56], um ein Dichte-basiertes Clustering von charakteristischen Punkten in den vorhandenen Trajektorien durchzuführen. Die dabei entstehenden Cluster stellen je nach Wahl des *minPts*-Parameters stark frequentierte Bereiche dar, die im Gegensatz zum Raster-basierten Ansatz den Verlauf der Trajektorien berücksichtigen. Die Cluster-Zentren werden anschließend als Hotspots identifiziert und geben den möglichen Lösungsraum für die Platzierung von Messknoten an.

Als erster Schritt werden nach Definition 2 die charakteristischen Punkte bestimmt, wobei  $\theta = \frac{1}{3}\pi$  gewählt wird, was einer Richtungsänderung von  $60^\circ$  entspricht. Dieser intuitive Wert stellt eine plausible untere Schranke für die Änderung der Bewegungsrichtung von Personen dar, ab der ein Punkt als charakteristisch angenommen wird. Es sei angemerkt, dass innerhalb von Gebäuden häufig stärkere Richtungsänderungen von ca.  $90^\circ$  auftreten, nämlich dann, wenn bspw. Personen um eine Ecke gehen oder einen Raum von einem Gang aus betreten. Die so ermittelten charakteristischen Punkte werden als Ergebnismenge  $R$  dem DBSCAN-Algorithmus für den Clustering-Schritt übergeben. Des Weiteren benötigt DBSCAN einen Wert für *minPts* und die Angabe der  $\varepsilon$ -Nachbarschaft. Da ein Messknoten potentiell auf ein Cluster-

Zentrum platziert wird, um möglichst das komplette Cluster zu erfassen, sollte die  $\varepsilon$ -Nachbarschaft geringer ausfallen als die Beobachtungsreichweite eines WLAN-Monitors. Daher gilt im vorliegenden Fall:  $\varepsilon < r_m$ . Zudem sollte ein Cluster nicht mehr Punkte beinhalten, als Trajektorien vorhanden sind, weshalb für  $minPts$  gilt:  $1 \leq minPts \leq |T|$ . Beide Parameter haben einen direkten Einfluss auf die Cluster-Bildung. Daher sollte deren Wahl bewusst unter folgender Überlegung getroffen werden: Für  $minPts = 1$  entstehen die meisten Kernpunkte und je nach Wahl von  $\varepsilon$  tendenziell viele unterschiedliche Cluster. Für  $minPts = |T|$  werden hingegen nur wenige bis keine Cluster gefunden. Da die Menge und Lage der identifizierten Cluster auch den Lösungsraum zur Platzierung von Messknoten bestimmt, sollte das Ergebnis im Optimalfall wenige gleichgroße Cluster mit Radius  $r = m_r$  beinhalten. Zudem sollten sie den gesamten Bereich möglichst gleichmäßig überdecken. Um ein solches Resultat zu erreichen, muss also eine geeignete Abstimmung der beiden Parameter  $\varepsilon$  und  $minPts$  erfolgen. Dies wird im Rahmen der Evaluation (vgl. Abschnitt 4.1.4) für ein bestimmtes Szenario näher untersucht. Im Folgenden wird der Dichte-basierte Ansatz erweitert, so dass nicht nur charakteristische Punkte, sondern auch weitere geometrische Eigenschaften zur Identifizierung von Hotspots herangezogen werden.

### Trajektorien-basierter Ansatz

Dieser Ansatz nutzt einen leicht veränderten TRACCLUS-Algorithmus [121] zur Erzeugung repräsentativer Trajektorien aus gefundenen Clustern. Anders als im ursprünglichen Algorithmus wird hier nicht das MDL-Prinzip genutzt, sondern es werden die charakteristischen Punkte, die nach Definition 2 bestimmt werden, zur Segmentierung verwendet. Dies führt im vorliegenden Fall zu weniger, aber genaueren Segmentierungspunkten, wodurch sich der Rechenaufwand reduzieren lässt, der bei der anschließenden Gruppierungs-Phase entsteht. Eine genauere Untersuchung zur Modifikation des TRACCLUS-Algorithmus für Indoor-Szenarien findet sich in der veröffentlichten Vorarbeit [183].

Wie in Abschnitt 2.4.3 beschrieben, nutzt der TRACCLUS-Algorithmus die Prinzipien des DBSCAN und verlangt für den Clustering-Schritt die beiden Parameter  $\varepsilon$  und  $minLns$ . Diese werden wie beim vorherigen Ansatz mit  $\varepsilon < r_m$  und  $1 \leq minLns \leq |T|$  eingegrenzt. Somit können Cluster entstehen, die eine minimale Anzahl an Subtrajektorien beinhalten und durch einen WLAN-Monitor vollständig erfassbar sind. Dabei gelten die gleichen Überlegungen zur geeigneten Abstimmung der beiden Parameter, wie sie bereits beim Dichte-basierten Ansatz getroffen wurden.

Pro Cluster wird anschließend eine repräsentative Trajektorie  $\tau_r \in T_r$  erzeugt, welche die charakteristische Bewegungsrichtung der gruppierten Subtrajektorien zusammenfasst. Damit beinhaltet die Menge  $T_r$  auch die Information über stark frequentierte Bereiche. Deshalb ist es wichtig, eine Platzierung von Messknoten zu finden, so dass alle repräsentativen Trajektorien nach Definition 1 erfassbar sind. Die einfachste Lösung wäre, wenn pro  $\tau_r$  ein Messknoten

in Reichweite platziert werden würde. Allerdings könnte dies zu Redundanzen führen, was den Ansatz ineffizient macht. Um das Problem zu vermeiden, wird nochmals ein Gruppierungsschritt vorgenommen. Ziel dabei ist es, diejenigen Stellen zu identifizieren, von denen aus mehrere repräsentative Trajektorien erfasst werden können. Hierzu wird der DBSCAN-Algorithmus mit  $\varepsilon = r_m$  und  $minPts = 1$  auf alle Start- und Endpunkte in  $T_r$  angewendet. Das Ergebnis sind wiederum Cluster und Rauschpunkte. Die Cluster fassen nun mindestens zwei Start- bzw. Endpunkte der repräsentativen Trajektorien zusammen, die innerhalb der Beobachtungsreichweite eines Messknotens liegen. Daher dienen die Cluster-Zentren als mögliche Knotenpositionen. Die Rauschpunkte dürfen hierbei nicht vernachlässigt werden, sondern müssen jeweils durch eine eigene Messstation erfasst werden, da sie aufgrund ihrer isolierten Lage nicht mit anderen Punkten zusammengefasst werden können. Als Ergebnis des Ansatzes werden also die Cluster-Zentren zusammen mit den Rauschpunkten als mögliche Monitorpositionen zurückgeliefert.

#### 4.1.4 Evaluation und Ergebnisse der Hotspoterkennung

In diesem Abschnitt werden die eben vorgestellten Ansätze mithilfe einer Simulation für ein größeres Gebäude evaluiert. Dabei sollen in einem ersten Schritt die jeweiligen Parameter analysiert werden, die zur Identifizierung der entsprechenden Hotspots führen. Das jeweils beste Resultat wird anschließend mit einer Brute-Force-Methode verglichen, die mittels Zufallsprinzip für eine gewisse Zeit nach der optimalen Platzierung sucht.

##### Aufbau und Durchführung

Als Grundlage für das Umgebungsmodell der Simulation dient ein Gebäudeplan des LMU Hauptgebäudes<sup>1</sup>. Dieser wird, wie in Abschnitt 3.2.1 beschrieben, einem Vorverarbeitungsschritt unterzogen und schließlich als bereinigte Bitmap-Darstellung verwendet. Das Ergebnis ist in Abbildung 4.2 dargestellt. Zur Evaluation werden auf dem Umgebungsmodell 20 Simulationen durchgeführt. Dabei wird die Zahl der simulierten Nutzer sukzessive von wenigen (10) auf viele (200) erhöht. Jeder Nutzer wählt zufällig fünf Räume als Ziele aus und bewegt sich anhand des in Abschnitt 3.2.2 beschriebenen Bewegungsmodells durch das gegebene Gebäude. Als maximale Beobachtungsreichweite eines WLAN-Monitors wird hier vereinfacht  $r_m = 25$  m angenommen, was nach Abedi et al. [1] einen realistischen unteren Wert darstellt und für die folgenden Versuche ausreichend ist. Die Reichweite lässt sich auch genauer über das Pfadverlustmodell aus Gleichung 2.1 bestimmen, wobei dafür die Modellparameter des vorliegenden Gebäudes bekannt sein müssen.

Damit liegen alle Eingabewerte für das erweiterte Art-Gallery-Problem vor. Zur Bewertung der Ansätze wird noch ein Qualitätskriterium benötigt. Hierzu

---

<sup>1</sup>Quelle: [https://www.uni-muenchen.de/funktionen/gebaeudeplaene/0000\\_d\\_00.pdf](https://www.uni-muenchen.de/funktionen/gebaeudeplaene/0000_d_00.pdf)

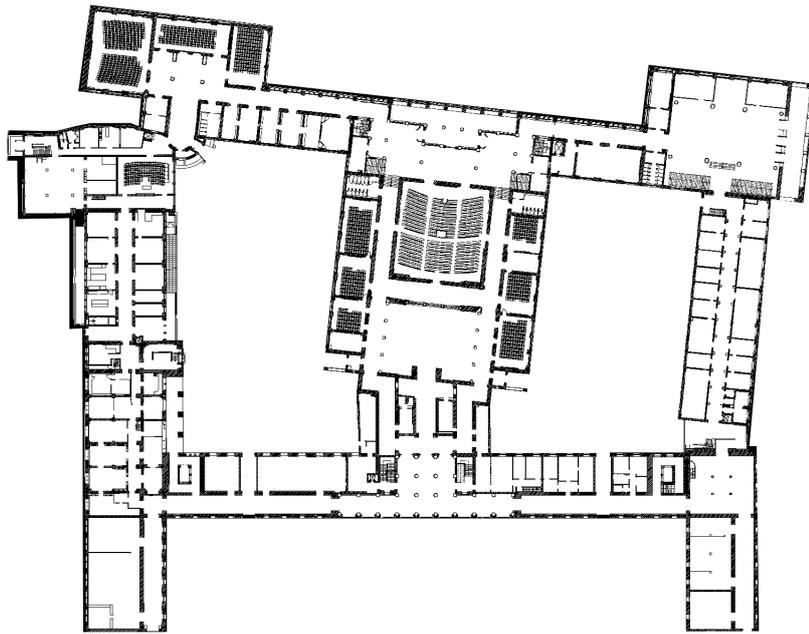


Abbildung 4.2: Darstellung des verwendeten Umgebungsmodells zur Simulation von Bewegungspfaden.

wird die akkumulierte Länge der erfassbaren Liniensegmente im Verhältnis zur Gesamtlänge aller Trajektorien berücksichtigt – anstelle nur die Anzahl der nach Definition 1 erfassbaren Subtrajektorien zu verwenden. Der Grund ist, dass dadurch die Länge als natürliche Gewichtung herangezogen wird. Somit sind kürzere Segmente bei der Bewertung weniger wichtig, wodurch der Einfluss von kurzen Subtrajektorien auf das Ergebnis verringert wird.

### Analyse der Parameter

Da das Ergebnis des jeweiligen Ansatzes von den gewählten Parameterwerten abhängt, soll zunächst deren Einfluss auf das Resultat in Bezug auf das gewählte Qualitätskriterium untersucht werden. Hierzu werden die Parameterwerte über den vorhandenen Definitionsbereich iteriert, wobei die resultierende Anzahl identifizierter Hotspots bzw. der Anteil der erfassbaren Länge evaluiert wird. Zudem soll die Anzahl an simulierten Trajektorien mitberücksichtigt werden, weshalb die folgenden Untersuchungen immer auf Basis von 10, 100 und 200 Nutzern durchgeführt werden, die jeweils fünf Ziele im Gebäude aufsuchen.

Im Fall des Raster-basierten Ansatzes wird der Schwellwert  $\varepsilon$  von 0,0 bis 1,0 iteriert. Dies wirkt sich konsequenterweise auf die Anzahl der als Hotspot identifizierten Zellenmittelpunkte und demnach auch auf den Anteil der erfassbaren Subtrajektorien aus, wie in Abbildung 4.3 zu sehen ist. So wird für  $\varepsilon = 1,0$  genau ein Hotspot gefunden (vgl. Abbildung 4.3a), in dessen Reichweite sich die meisten Subtrajektorien befinden. Im Gegenzug werden bei  $\varepsilon = 0,0$

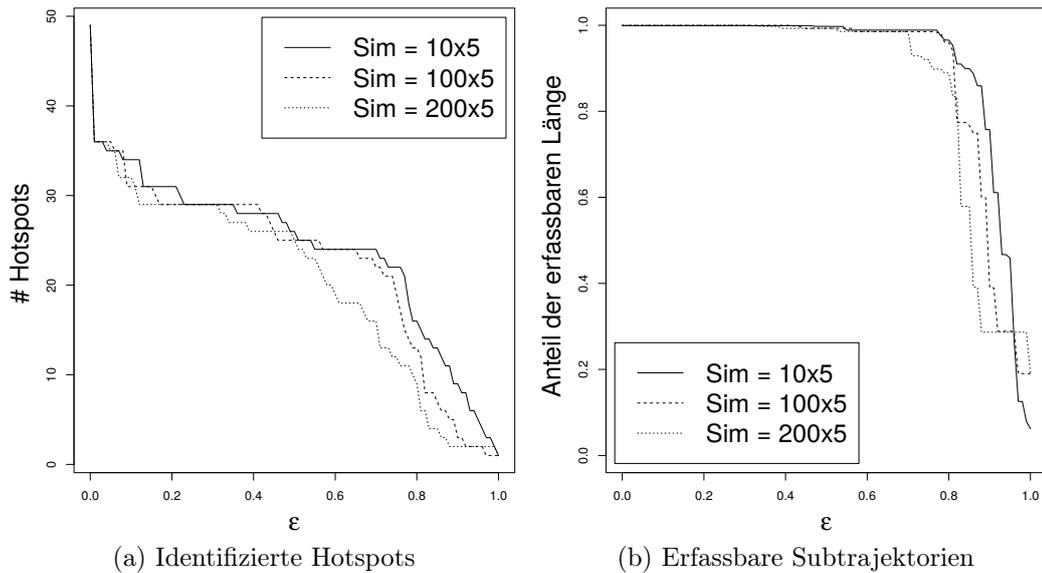


Abbildung 4.3: Einfluss des  $\varepsilon$ -Parameters auf das Ergebnis des Rasterbasierten Ansatzes.

alle Zellenmittelpunkte als Hotspots deklariert, wodurch alle vorhandenen Liniensegmente erfassbar sind (vgl. Abbildung 4.3b). Dabei zeigt sich, dass für eine geringere Anzahl an Trajektorien generell mehr Zellen als Hotspots identifiziert werden, vor allem im Bereich von  $\varepsilon > 0,5$ . Dies lässt sich dadurch erklären, dass die Pixelsumme bei weniger Pfaden nicht so stark variiert, weshalb mehr Zellen bei gleichem  $\varepsilon$ -Wert den geforderten Anteil des Maximalwerts überschreiten und somit als Hotspots deklariert werden.

Unabhängig davon ist in Abbildung 4.3b eine interessante Eigenschaft zu beobachten: Der Anteil der erfassbaren Länge bleibt im Bereich von  $0, 0 \leq \varepsilon \leq 0,7$  nahezu konstant bei knapp 100% und nimmt erst bei  $\varepsilon > 0,7$  rapide ab. Vergleicht man dazu die identifizierten Hotspots in Abbildung 4.3a lässt sich hier ein guter Kompromiss zwischen maximaler Erfassbarkeit und minimaler Anzahl an Hotspots bei  $\varepsilon \approx 0,73$  ausmachen. Bei dieser Belegung werden mit dem Rasterbasierten Ansatz auf Basis von 200 simulierten Trajektorien im vorliegenden Gebäude 13 Monitor-Positionen gefunden, mit deren Hilfe 92,9% der Gesamtlänge als erfassbar angenommen werden.

Im Fall des Dichte-basierten Ansatzes müssen die beiden Parameter  $\varepsilon$  und  $minPts$  analysiert werden, da diese das Clustering und somit auch das Gesamtergebnis beeinflussen. In einer Vorstudie wurde daher zunächst die  $\varepsilon$ -Nachbarschaft für das gegebene Szenario mit  $1 < \varepsilon < r_m$  untersucht. Dabei zeigte sich, dass für  $\varepsilon \approx \frac{1}{4}r_m$  augenscheinlich die geeignetsten Cluster-Größen gebildet werden konnten, die etwas kleiner als die angenommene Reichweite eines Messknotens ausfallen, was hier vorteilhaft ist. Für  $\varepsilon = r_m$  waren die Cluster hingegen zu groß, während für  $\varepsilon < \frac{1}{4}r_m$  zu viele kleine Cluster mit einer hohen Anzahl an Rauschpunkten entdeckt wurden. Aus diesem Grund

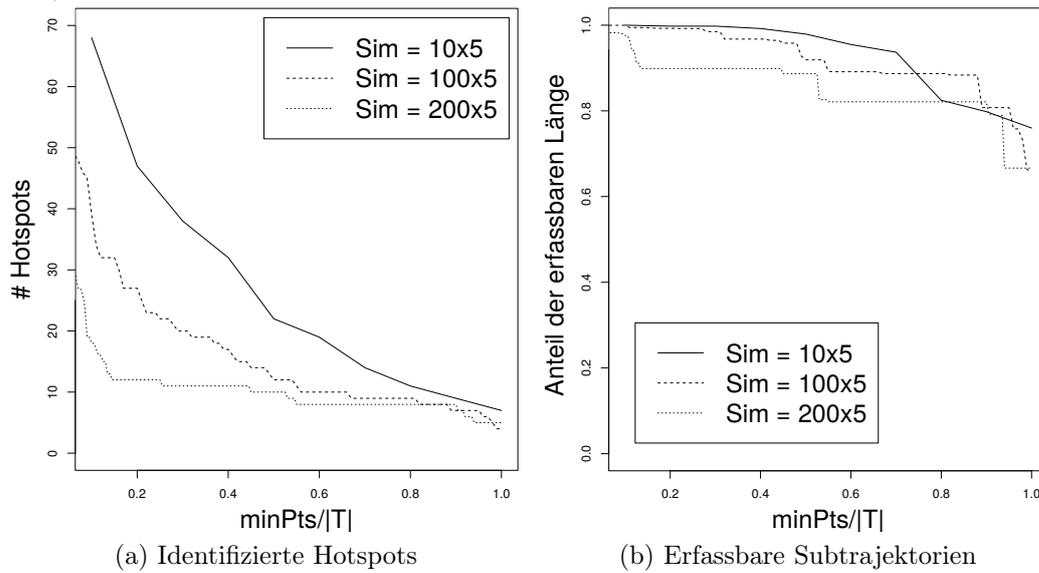


Abbildung 4.4: Einfluss des  $minPts$ -Parameters auf das Ergebnis des Dichte-basierten Ansatzes.

wird im Folgenden  $\varepsilon = \frac{1}{4}r_m$  festgelegt und der Einfluss des  $minPts$ -Parameters für die verschiedenen Simulationsstufen analysiert.

Abbildung 4.4 zeigt hierzu wieder den Einfluss auf das Ergebnis, wenn  $minPts$  von 1 bis  $|T|$  iteriert wird. Zur besseren Veranschaulichung wird dabei der Parameterwert mittels  $|T|$  normalisiert. Wie aus Abbildung 4.4a ersichtlich wird, sinkt die Anzahl an identifizierten Hotspots mit steigendem Wert von  $minPts$ , was aufgrund des DBSCAN-Algorithmus so zu erwarten ist. Gleichzeitig hat die Menge der Trajektorien wieder einen Einfluss auf das Ergebnis, was sich ebenfalls durch den Clustering-Ansatz begründen lässt. Demnach führen mehr Pfade zu mehr Segmentierungspunkten und somit auch zu einer allgemein höheren Punktdichte. Dadurch entstehen bei gleichem  $minPts$ -Wert weniger aber größere Cluster, weshalb auch weniger Hotspots gefunden werden.

Vergleicht man die Anzahl der Hotspots wieder mit der anteiligen Länge der erfassbaren Subtrajektorien aus Abbildung 4.4b, so zeigt sich, dass mindestens 65% der Länge durch sieben Messknoten erfasst werden können. Dies ist ein adäquates Verhältnis für das gegebene Gebäude. Um jedoch alle 200 Trajektorien zu erfassen, benötigt man hingegen bis zu 28 Monitore. Ein guter Kompromiss wird bei  $minPts = 25$  festgestellt. Hierbei werden 91,2% der Länge mittels 15 Monitoren als erfassbar angenommen, was allerdings eine leichte Verschlechterung zum vorherigen Ansatz darstellt.

Vielversprechender ist der Trajektorien-basierte Ansatz. Allerdings wird bei diesem das Ergebnis von den beiden Parameter  $\varepsilon$  und  $minLns$  beeinflusst. Da die  $\varepsilon$ -Nachbarschaft über Liniensegmente hier eine Ellipse bildet und nicht wie zuvor einen Kreis, muss zunächst ein geeigneter Wert für  $\varepsilon$  bestimmt werden. Zu diesem Zweck wurden ebenfalls in einer Vorstudie beide Parameter

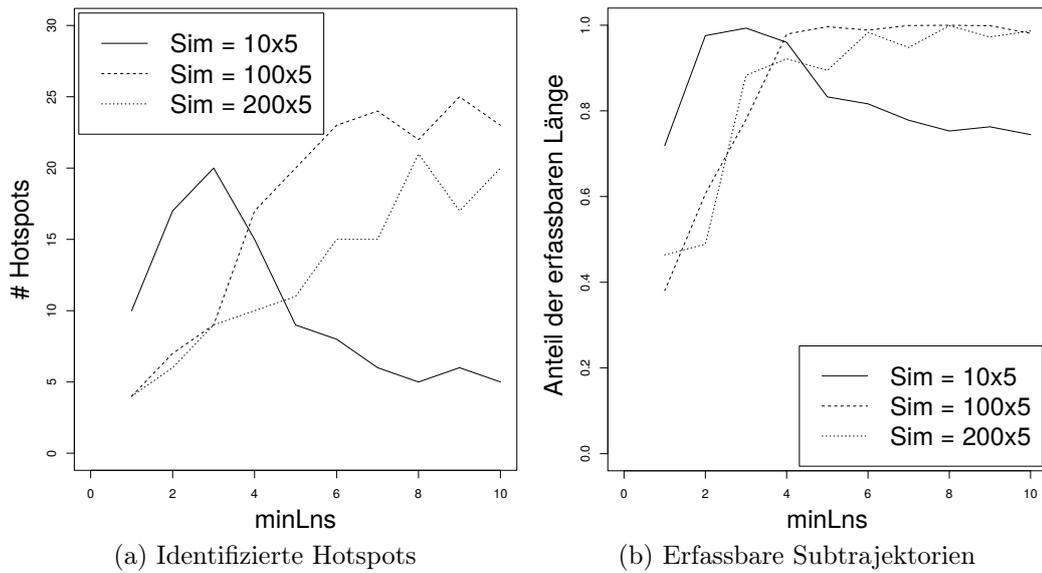


Abbildung 4.5: Einfluss des  $minLns$ -Parameters auf das Ergebnis des Trajektorien-basierten Ansatzes.

über  $1 \leq \varepsilon \leq r_m$  und  $1 \leq minLns \leq |T|$  variiert und das Cluster-Ergebnis augenscheinlich untersucht. Dabei zeigten sich brauchbare Cluster-Größen für  $\varepsilon \approx \frac{1}{8}r_m$  und  $minLns \leq 10$ . Im Fall von  $minLns > 10$  haben sich adäquate Cluster aufgelöst, wodurch sich die Zahl der Rauschlinien erhöht hat.

Deshalb soll auf Basis von  $\varepsilon = \frac{1}{8}r_m$  der Einfluss des  $minLns$ -Parameters für  $1 \leq minLns \leq 10$  untersucht werden. In Abbildung 4.5 sind die entsprechenden Ergebnisse dargestellt. Im Gegensatz zu den beiden vorherigen Ansätzen zeigt sich hier ein deutlicher Unterschied, wenn nur wenige Trajektorien vorliegen. Dies lässt sich durch den Gruppierungsschritt des TRACCLUS-Algorithmus begründen, der bei wenigen Liniensegmenten auch wenige Cluster findet, die mit der Erhöhung von  $minLns$  nach und nach zerfallen und in Rauschlinien übergehen. Demnach sind 10x5 simulierte Trajektorien bei diesem Gebäude und für den untersuchten Ansatz zu wenig. Die erzielten Ergebnisse für 100 und mehr Simulationen zeigen hingegen einen erwartbaren Verlauf. Dabei erhöht sich mit steigendem  $minLns$ -Wert auch die Anzahl identifizierter Hotspots und damit auch der Anteil erfassbarer Subtrajektorien. Der Grund ist, dass durch Erhöhung von  $minLns$  mehr kleinere Cluster gefunden werden, welche zu mehr repräsentativen Trajektorien führen. Ein geeigneter Kompromiss zwischen maximaler Erfassbarkeit und minimaler Anzahl an Hotspots kann bei  $minLns = 6$  für 200 simulierte Trajektorien beobachtet werden. Dabei werden 15 Positionen für Messknoten identifiziert, wobei 98,2% der Gesamtlänge als erfassbar angenommen wird. Dieses Verhältnis stellt das beste Ergebnis dar, was im Rahmen der durchgeführten Untersuchungen erreicht wurde. Abbildung 4.6 zeigt daher die entsprechende Platzierung für das gegebene Gebäude. Dabei lässt sich feststellen, dass die gesetzten Monitorknoten, die hier als Ster-

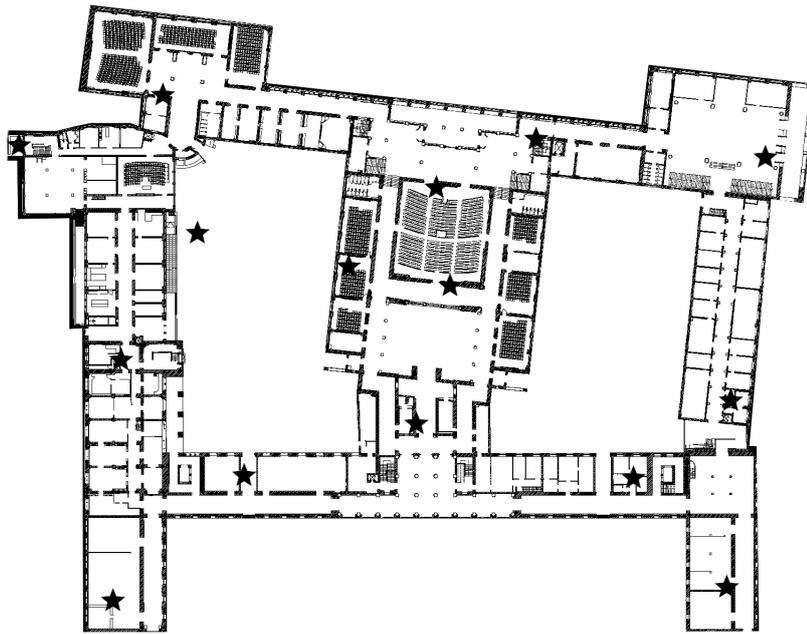


Abbildung 4.6: Sterne veranschaulichen die Platzierung von 15 Messknoten im Gebäude anhand des Trajektorien-basierten Ansatzes.

ne symbolisiert sind, an strategisch wichtigen Orten des Gebäudes liegen und alle Eingänge sowie die Ecken der Gänge überwachen können. Somit stellt dies ein plausibles Ergebnis für die effiziente Erfassung von Nutzerströmen dar.

### Bewertung der Ergebnisse

Zur Bewertung werden die erreichten Ergebnisse mit einer Brute-Force-Methode verglichen, die alle möglichen Kombinationen durchprobiert und das beste Ergebnis hinsichtlich des gewählten Qualitätskriteriums zurückliefert. So wird zufällig nach einer optimalen Platzierung für die gleiche Knotenanzahl gesucht, die von den jeweiligen Ansätzen gefunden werden. Dabei gehören die Koordinaten aller weißen Pixel zur möglichen Lösungsmenge, wodurch sichergestellt ist, dass sich die optimale Platzierung hinsichtlich des gewählten Qualitätskriteriums in der Lösungsmenge befinden muss. Um diese zu identifizieren, müssen bei  $n$  möglichen Positionen und einer Anzahl von  $k$  Messknoten  $\binom{n}{k}$  Kombinationen ausprobiert und hinsichtlich des Qualitätskriteriums bewertet werden. Dies ist für größere Bitmaps wie im vorliegenden Fall extrem zeitaufwendig und daher nicht vollständig durchführbar. Aus diesem Grund wird hier die Brute-Force-Methode auf 100.000 Iterationen beschränkt, was bereits mehr Zeit zur Berechnung in Anspruch nimmt, als die evaluierten Verfahren benötigen. Dadurch wird zwar das optimale Resultat nicht zwangsläufig erreicht, aber aufgrund der hohen Anzahl an Iterationen zumindest approximiert. Das Ergebnis wird jeweils mit den drei geometrischen Ansätzen verglichen.

Ansatz	Raster	Dichte	Trajektorien
Parameter	$\varepsilon = 0,73$	$\varepsilon = \frac{1}{4}r_m$ $minPts = 25$	$\varepsilon = \frac{1}{8}r_m$ $minLns = 6$
# Hotspots	13	15	15
Erfassbare Länge (Ansatz)	92,2%	91,2%	98,2%
Erfassbare Länge (Brute-Force)	97,9%	98,1%	98,1%
Differenz zwischen Ansatz und Brute-Force	-5,7%	-6,9%	0,1%

Tabelle 4.1: Ergebnisse der Ansätze zur Identifizierung von Hotspots im Vergleich zur Brute-Force-Methode.

Tabelle 4.1 fasst diesen Vergleich zusammen. Dabei werden zur Bewertung jeweils die besten Ergebnisse berücksichtigt, die mittels der genannten Parameter auf 200 simulierten Trajektorien erzielt wurden. Ein Ansatz gilt dann als effizient, wenn er bei gleicher Knotenanzahl eine höhere Güte als die Brute-Force-Methode erreicht. Wie aus der Tabelle 4.1 (letzte Reihe) hervorgeht, trifft dieses Kriterium nur auf den Trajektorien-basierten Ansatz zu, der damit auch insgesamt das beste Resultat liefert. Der Dichte-basierte Ansatz schneidet am schlechtesten ab, gefolgt vom Raster-basierten Ansatz. Es ist anzumerken, dass diese beiden Ansätze das Ergebnis der Brute-Force-Methode nur knapp verfehlen. Dadurch lässt sich schlussfolgern, dass selbst eine einfache Strategie helfen kann, geeignete Stellen zur Platzierung von Messknoten zu identifizieren. Eine willkürliche Platzierung wie sie oft in verwandten Arbeiten zu finden ist, scheint auf Grundlage der erreichten Ergebnisse die schlechtere Wahl zu sein.

#### 4.1.5 Zusammenfassung und Diskussion

Auf Basis der durchgeführten Evaluation konnte gezeigt werden, dass geeignete Platzierungsstrategien das Verhältnis von Knotenanzahl und Erfassbarkeit von Trajektorien verbessern können. Dadurch lassen sich Redundanzen verringern und Kosten für Installations- und Wartungsarbeiten einsparen. Im Gegensatz zu verwandten Arbeiten basieren die hier vorgestellten Ansätze rein auf der geometrischen Information vorhandener (simulierter) Trajektorien. Während der Raster-basierte Ansatz die bloße Lage von Pfaden innerhalb festgelegter Zellen analysiert, nutzt der Dichte-basierte Ansatz die Häufung charakteristischer Punkte, welche eine signifikante Richtungsänderung beschreiben. Der Trajektorien-basierte Ansatz hingegen vereint mehrere geometrische Informationen wie Lage, Richtung und Länge einzelner Liniensegmente und fasst diese als repräsentative Trajektorien zusammen.

Insgesamt lassen sich die Verfahren generisch auf beliebige Bewegungsdaten und Gebäudepläne anwenden. Allerdings erfordert diese Generalisierbar-

keit eine individuelle Wahl der Parameterwerte, welche die Ergebnisse stark beeinflussen und somit als kritisch zu betrachten sind. Der Vorteil des Rasterbasierten Ansatzes ist deshalb, dass dieser nur einen Wert benötigt und weniger stark von der Gesamtzahl vorhandener Trajektorien beeinflusst wird. Zudem werden hiermit bessere Ergebnisse erreicht, als mit dem Dichte-basierten Ansatz, welcher zwei Parameter für den Clustering-Schritt verlangt.

Das Problem des Dichte-basierten Ansatzes ist, dass Segmentierungspunkte zusammengefasst werden, die nur auf der Richtungsänderung der Trajektorien beruhen. Somit werden wichtige Charakteristika wie bspw. die Länge oder Lage der Subtrajektorien nicht berücksichtigt, was zu einem schlechteren Resultat bzgl. des gewählten Qualitätskriteriums führt. Dieser Nachteil wird durch den Trajektorien-basierten Ansatz ausgeglichen, der insgesamt ein adäquates Ergebnis im Rahmen der durchgeführten Evaluation erzielt. Hierbei sei angemerkt, dass auch auf anderen Gebäudeplänen, die in [175] evaluiert wurden, tendenziell ähnliche Ergebnisse für die drei Ansätze beobachtet wurden.

So zeigt der Trajektorien-basierte Ansatz insgesamt das beste Resultat auch im Vergleich zur Brute-Force-Methode. Allerdings erwartet dieser eine gewisse Häufung an Liniensegmenten, um geeignete Cluster finden zu können. Der Trajektorien-basierte Ansatz ist somit wie der Dichte-basierte Ansatz stärker abhängig von der Lage und Anzahl der gegebenen Trajektorien als der Rasterbasierte Ansatz. Zudem werden ebenfalls zwei Parameter für den Clustering-Schritt benötigt, was eine individuelle Kalibrierung erfordert. Zusammenfassend lässt sich festhalten, dass die entwickelten Verfahren eine generelle Hilfestellung geben, um adäquate Monitor-Positionen für eine effiziente Erfassung von Nutzerströmen zu finden. Auf Basis einer entsprechenden Platzierung von Messknoten können nun Bewegungspfade ermittelt werden.

## 4.2 Pfadermittlung auf Basis ungenauer Positionsdaten

Wie anfangs erwähnt, zählt der aktuelle Ort eines Nutzers zu den wichtigsten und primären Typen von Kontextinformationen [2]. Zudem lassen sich aus Zeitreihen von Positionsdaten auch andere Kontextinformationen wie Bewegung, Verhalten oder Vorlieben eines Nutzers ableiten [243]. Auch zur Erfassung und Analyse von Nutzerströmen spielt die kontinuierliche und möglichst exakte Lokalisierung von Personen eine entscheidende Rolle. Allerdings ist eine solche Standortverfolgung auf Basis von passiven WLAN-Beobachtungen nach wie vor herausfordernd: Zum einen ist das Sendeverhalten der mobilen Geräte nicht deterministisch, weshalb Beobachtungen nur schwer vorhersagbar sind (vgl. Abschnitt 2.1.2), zum anderen kommt es, wie bereits erwähnt, innerhalb von Gebäuden vermehrt zu Signaldämpfung bzw. Mehrwegeausbreitung, was zu verrauschten Empfangssignalstärken und somit zu größeren Fehlern bei der Positionsbestimmung führt.

Angesichts solcher Schwierigkeiten und der damit zu erwartbaren Ungenauigkeit in den Positionsschätzungen erscheinen vor allem probabilistische Verfahren zielführend. Daher werden im vorliegenden Teil des Kapitels zunächst verwandte Arbeiten vorgestellt und davon die gebräuchlichsten Verfahren für die WLAN-basierte Standortverfolgung in Gebäuden untersucht. Allerdings zeigen diese verschiedene Nachteile in Bezug auf anerkannte Metriken aus der Indoor-Positionierung, weshalb im weiteren Verlauf ein hybrider Ansatz entwickelt wird, der hinsichtlich dieser Metriken eine verbesserte Performanz erreicht. Die diskutierten Verfahren werden schließlich innerhalb einer realen Umgebung evaluiert und die Ergebnisse miteinander verglichen und bewertet.

### 4.2.1 Verwandte Arbeiten

RADAR von Bahl und Padmanabhan [16] gilt als eines der ersten WLAN-basierten Tracking-Systeme für Gebäude. Es nutzt ein Fingerprinting-Verfahren mit einem k-nächste-Nachbarn-Algorithmus und erreicht einen durchschnittlichen Positionsfehler von 2,94 Metern. In einer erweiterten Version führen Bahl et al. [17] einen Viterbi-basierten Algorithmus zur kontinuierlichen Standortverfolgung von Nutzern ein und reduzieren damit den durchschnittlichen Fehler auf 2,37 Meter. Dies verdeutlicht, dass probabilistische Ansätze zur Pfadermittlung besser geeignet sind und eine höhere Genauigkeit erzielen als deterministische Verfahren. Auch Krumm und Horvitz [116] nutzen einen Viterbi-basierten Ansatz und zwei unterschiedliche HMMs, um den Bewegungsstatus und den aktuellen Ort eines mobilen Geräts mithilfe von WLAN-Messungen zu ermitteln. Auf Basis von aktiven WLAN-Scans, die während zehn kurzer Bewegungspfade durchgeführt wurden, erreichen die Autoren einen durchschnittlichen Fehler von 1,53 Metern.

Durch die weite Verbreitung moderner mobiler Geräte hat in den letzten Jahren vor allem das Interesse an Echtzeit-Tracking-Systemen zugenommen. So stellen bspw. Trogh et al. [201] ein solches System vor, das ebenfalls den Viterbi-Algorithmus und zusätzlich semantische Daten nutzt. Die Positionsbestimmung basiert dabei auf WLAN-Fingerprinting mit einer berechneten Radiomap, wobei der durchschnittliche Positionsfehler unter zwei Metern liegt. Musa und Eriksson [145] untersuchen die WLAN-basierte Standortverfolgung für den Außenbereich auf Basis von Messknoten, die über ein Straßennetz verteilt sind. Darauf wird ein HMM konstruiert und der Viterbi-Algorithmus zur Bestimmung des wahrscheinlichsten Bewegungspfad verwendet. Im Vergleich zu GPS-Daten als Grundwahrheit erreichen die Autoren eine durchschnittliche Abweichung von unter 70 Metern. Andere Arbeiten, die sich mit WLAN-basierten Indoor-Tracking-Systemen beschäftigen, berechnen hingegen eine einfache Wahrscheinlichkeitsverteilung für den aktuellen Ort mithilfe der Signalstärke, die an unterschiedlichen Messstationen empfangen wurde [127]. Zum Teil wird auch ein reiner Laterations-Ansatz durchgeführt, der allerdings nur eine Genauigkeit von 15 Metern erreicht [171].

Die bisher erwähnten Arbeiten zeigen, dass der Viterbi-Algorithmus in Kombination mit geeigneten HMMs eine weite Verbreitung in der WLAN-basierten Standortverfolgung findet. Als gängige Alternative hierzu werden sehr häufig Partikel-Filter eingesetzt, die zusammen mit der entsprechenden Karteninformation aus Gebäudeplänen eine hohe Genauigkeit bei der Pfadermittlung erreichen [65, 123, 153, 208]. Auch werden solche Filter gerne in Kombination mit anderen Verfahren genutzt. Bartoletti et al. [21] entwickeln bspw. ein komplettes Indoor-Tracking-System, welches mobile Objekte über zwei Techniken verfolgt. Zum einen wird über Ultrabreitbandsensoren und das Laufzeitverfahren *Time-of-Arrival* eine Distanzmessung durchgeführt und die Position des Objekts über Lateration bestimmt. Zur kontinuierlichen Standortverfolgung wird dann zusätzlich ein Partikel-Filter eingesetzt. Khan et al. [105] vergleichen Kalman Filter, den erweiterten Kalman Filter und einen Partikel-Filter für die Standortverfolgung basierend auf gemessenen Signalstärken. Ähnliche Untersuchungen werden von Chen et al. [35] durchgeführt, wobei hier der Fokus auf Indoor-Umgebungen liegt. Zusätzlich werden diverse Vereinfachungen für den SIR Partikel-Filter diskutiert, die zu einer performanteren Standortverfolgung führen, wenn mehrere Ziele gleichzeitig verfolgt werden.

Obwohl der Viterbi-Algorithmus und der Partikel-Filter zwei sehr geläufige Verfahren zur probabilistischen Standortverfolgung darstellen, wurden diese in der betrachteten Literatur bisher nicht miteinander in Beziehung gestellt oder anhand anerkannter Metriken verglichen. Diese Lücke soll im weiten Verlauf des vorliegenden Kapitels geschlossen werden. Zusätzlich wird ein sogenannter *State-Partikel-Filter* entwickelt und evaluiert. Dieser kombiniert die Flexibilität eines SIR Freiflächen-Partikel-Filters mit diskreten Zuständen eines zugrunde liegenden HMMs. Nach heutigem Kenntnisstand des Autors der vorliegenden Arbeit wurde ein derartiger Ansatz für die WLAN-basierte Standortverfolgung bis dato nicht vorgestellt bzw. untersucht. Der Begriff *State-Partikel-Filter* kam bisher lediglich in einigen wenigen Publikationen vor, die sich mit der Verfolgung von Formanten [242] oder der Erkennung der Kopfhaltung von Personen in Videosequenzen [15] beschäftigen.

## 4.2.2 Anwendung bestehender Verfahren

Wie zuvor deutlich wurde, findet sowohl der Viterbi-Algorithmus als auch der Partikel-Filter in verschiedenen Variationen eine große Verbreitung bei der kontinuierlichen Lokalisierung von Nutzern. Die allgemeinen Grundlagen hierzu wurden bereits im Abschnitt 2.3 erläutert. Im Folgenden werden nun beide Verfahren konkret für das vorliegende Problem der Pfadermittlung auf Basis von Signalbeobachtungen in einem Gebäude angewendet. Zum Vergleich der Positionsschätzungen bietet sich ein herkömmlicher deterministischer WLAN-Fingerprinting-Ansatz an.

### Positionsbestimmung mittels WLAN-Fingerprinting

Im vorliegenden Fall wird die Radiomap mittels des Pfadverlustmodells aus Gleichung 2.1 für eine bestimmte Anzahl an Referenzpunkten und einer gegebenen Anordnung an WLAN-Monitoren in einem Gebäude berechnet. Details zum konkreten Versuchsaufbau und zu den gewählten Modellparametern werden im späteren Abschnitt 4.2.4 vorgestellt.

Eine Positionsschätzung für ein bestimmtes Gerät kann immer dann erfolgen, wenn in einer kurzen Zeitspanne  $\Delta t$  die zugehörige MAC-Adresse von einem Monitor  $m$  beobachtet wird. Es muss also wieder das entsprechende Beobachtungstupel  $o_m := \{t, \text{MAC}, \text{RSS}\}$  an mindestens einer Messstation  $m \in M$  aufgezeichnet werden. Die Tupel von allen Monitoren für die Zeitspanne  $[t - \Delta t, t]$  werden zu einem Beobachtungsvektor  $\vec{v}_{t,\text{MAC}}$  aggregiert und fehlende Werte mit -100 dBm komplettiert (vgl. Abschnitt 3.2.3). Ein solcher Vektor beinhaltet demnach die Empfangssignalstärken eines Geräts an allen verfügbaren Messstationen zum Zeitpunkt  $t$ . Somit repräsentiert  $\vec{v}_{t,\text{MAC}}$  den RSS-Vektor, welcher in der Online-Phase des WLAN-Fingerprinting Verfahrens zum Abgleich mit der Radiomap und damit zur Positionsbestimmung verwendet wird.

Im vorliegenden Fall wird zur Bestimmung der Position ein klassischer deterministischer Ansatz umgesetzt, der auf dem Prinzip des gewichteten *k-Nächste-Nachbarn-Verfahrens* (kNN) basiert. Dabei werden die Gewichte der  $k$  nächstgelegenen Referenzpositionen  $w_i$  mit  $i \in \{1, \dots, k\}$  wie folgt bestimmt:

$$w_i = \left( d_i \sum_{j=1}^k \frac{1}{d_j} \right)^{-1} \quad (4.2)$$

Dabei stellt  $d_i = \delta(\vec{v}_{t,\text{MAC}}, \vec{r}_i)$  die euklidische Distanz im Signalraum zwischen dem Beobachtungsvektor  $\vec{v}_{t,\text{MAC}}$  und einem bestimmten Eintrag  $\vec{r}_i$  in der Fingerprinting-Datenbank dar. Die  $k$  Gewichte werden normalisiert, so dass gilt:  $\sum_{i=1}^k w_i = 1$ . Dadurch lässt sich die Position über die Summe der gewichteten  $k$  Referenzpositionen bestimmen.

Über den Zeitverlauf entstehen so auf Basis der Beobachtungsvektoren und des hier beschriebenen Verfahrens geräteabhängige Zeitreihen von unregelmäßigen und ungenauen Positionsdaten. Ziel ist es nun, den wahrscheinlichsten Bewegungspfad eines Nutzers zu ermitteln. Hierzu bieten sich die aus der Literatur bekannten Verfahren an.

### Viterbi-Algorithmus zur Pfadermittlung

Um den Viterbi-Algorithmus zur Pfadermittlung nutzen zu können, muss zunächst ein geeignetes HMM erstellt werden. Wie aus Abschnitt 2.3.2 hervorgeht, wird ein HMM formal als ein Fünfer-Tupel  $\lambda = (S; O; A; B; \pi)$  definiert. Die Modellparameter werden sukzessive für das vorliegende Problem der WLAN-basierten Standortverfolgung bestimmt.

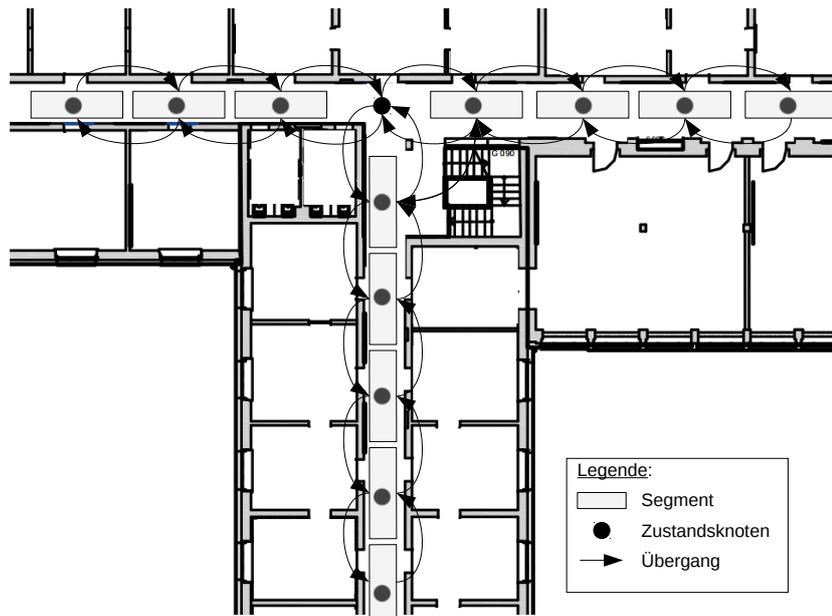


Abbildung 4.7: Schematische Darstellung des gewählten Zustandsmodells in einem Gebäudeabschnitt. Kanten werden dabei in Segmente unterteilt und von je einem Zustandsknoten repräsentiert.

**Menge der verborgenen Zustände ( $S$ ):** Das Zustandsmodell besteht aus diskreten Positionen innerhalb des gegebenen Gebäudes und muss letztendlich die Menge aller möglichen Pfade, die von einem Nutzer begangen werden können, in einer hinreichenden Genauigkeit abbilden können. Aus diesem Grund werden im vorliegenden Fall die Zustände als Referenzpositionen bzw. als sogenannte Zustandsknoten (engl.: *state nodes*) modelliert. Diese müssen so verteilt werden, dass die Topologie des Gebäudes und die Bewegungsfreiheit der Nutzer durch den Zustandsraum repräsentiert wird. Einfach ausgedrückt sollen die Zustandsknoten hauptsächlich entlang der Gänge eines Gebäudes platziert werden, da hier auch die größte Bewegung stattfindet. Dabei lassen sich in Analogie an das *Map-Matching-Problem* auf Straßenkarten, welches in [145] für die WLAN-basierte Standortverfolgung untersucht wird, Gangkreuzungen als Knoten und die Gänge selbst als Kanten eines ungerichteten Graphen auffassen. Lange Kanten können zudem in Segmente gleicher Länge unterteilt werden, um feingranularere Abschnitte zu erhalten. Alle Knoten und Segmente dieses Graphen müssen letztendlich, wie in Abbildung 4.7 schematisch dargestellt, durch einen Zustandsknoten repräsentiert werden. Dabei gelten die folgenden beiden Bedingungen:

1. Zwei benachbarte Knoten befinden sich immer in direktem Sichtkontakt zueinander. Das bedeutet, dass eine direkte Verbindung zwischen den Knoten existiert, die frei von Hindernissen ist.

2. Zwei benachbarte Knoten  $i, j \in S$  haben immer den gleichen Abstand  $d_{i,j}$  zueinander, der durch einen konstanten Wert oder wie im vorliegenden Fall durch eine einfache Heuristik festgelegt wird.

Zur Berechnung von  $d_{i,j}$  wird hier eine konstante Gehgeschwindigkeit für Fußgänger  $v_p$  angenommen und die festgelegte Zeitspanne der WLAN-Beobachtung  $\Delta t$  herangezogen, so dass gilt:  $d_{i,j} = v_p \cdot \Delta t$ . Diese Heuristik stellt sicher, dass für jede neue WLAN-Beobachtung auch ein neuer Zustandsknoten erreicht werden kann, wenn sich die Nutzer ungefähr mit  $v_p$  fortbewegen. Das System bleibt somit nicht in einem Zustand gefangen und liefert für jeden Zustandsübergang den wahrscheinlichsten Zustandsknoten.

**Übergangswahrscheinlichkeit (A):** Im Gegensatz zu Straßennetzen können sich Personen innerhalb eines Gebäudes nahezu frei bewegen. Das bedeutet, dass es zu plötzlichen Richtungs- bzw. Geschwindigkeitsänderungen in der aktuellen Bewegung kommen kann. Im vorliegenden Modell können Nutzer also an jedem Zustandsknoten stehenbleiben, weitergehen oder sogar komplett umkehren. Die Matrix  $A$  muss diese Bedingung für die Bestimmung der Übergangswahrscheinlichkeiten zwischen zwei Knoten  $i$  und  $x$  berücksichtigen. Da der Viterbi-Algorithmus periodische Beobachtungen voraussetzt, wird der Zustandsübergang auch periodisch auf Basis der WLAN-Beobachtungen  $O$  und der Matrix  $A$  durchgeführt. Dabei kann es vorkommen, dass für eine Periode keine Beobachtung zur Verfügung steht und deshalb keine Annahme über die Fortbewegung einer Person getroffen wird. Liegt hingegen eine Beobachtung  $\vec{v}_{t,\text{MAC}}$  vor, so wird zunächst der Zeitunterschied  $\delta t$  zur letzten Beobachtung dieser MAC-Adresse bestimmt und die Distanz  $d_{i,x}$  vom aktuellen Zustandsknoten  $i$  zu jedem anderen Zustand  $x$  errechnet. Daraufhin lässt sich die Geschwindigkeit  $v_x$  bestimmen, die benötigt wird, um den Zustandsknoten  $x$  in der gegebenen Zeit von  $i$  aus zu erreichen:  $v_x = \frac{d_{i,x}}{\delta t}$ . Unter der Annahme, dass die Gehgeschwindigkeit eines Fußgängers eine normalverteilte Zufallsgröße  $V \sim \mathcal{N}(\mu, \sigma^2)$  mit  $\mu = v_p$  und  $\sigma^2 = 1$  darstellt, wird hier eine gefaltete Normalverteilung  $Y = |V|$  zur Berechnung der Übergangswahrscheinlichkeit zwischen  $i$  und einem anderen Zustand  $x$  herangezogen:

$$p(i \rightarrow x) = f_Y(v_x; \mu, \sigma^2) \quad (4.3)$$

Dabei stellt  $f_Y$  die WDF über  $Y$  dar, gegen die  $v_x$  getestet wird. Insgesamt ergeben sich so die Einträge der Matrix  $A[i, x] = p(i \rightarrow x)$  für den Zustandsübergang.

**Beobachtungswahrscheinlichkeiten (B):** Wie bereits erwähnt, können für Zeitabschnitte, in denen keine Beobachtungen gemacht wurden, auch keine Aussagen über die Bewegung einer Person getroffen werden. Dies ist konträr zu Musa und Eriksson [145], die auch Perioden ohne Beobachtungen für die Pfadermittlung betrachten, indem sie ein einfaches Modell für die Emissionswahrscheinlichkeit von Probe Requests nutzen (vgl. Abschnitt 3.2.3). Ein solcher Ansatz ist allerdings kritisch zu betrachten, da ein generisches Modell

für die Emissionswahrscheinlichkeit fehlt. Zudem kann nicht implizit von einer fehlenden Beobachtung darauf geschlossen werden, dass ein mobiles Endgerät einen anderen Weg genommen hat. Aus diesem Grund werden im vorliegenden Fall nur die tatsächlich aufgenommenen Beobachtungsvektoren für die Pfadermittlung herangezogen. Zur Berechnung der Beobachtungswahrscheinlichkeiten wird wie bei der Radiomap für das WLAN-Fingerprinting ein RSS-Vektor  $\vec{v}_s$  für jeden Zustandsknoten  $s \in S$  über das Pfadverlustmodell errechnet. Analog zum Horus-System [238] wird zusätzlich davon ausgegangen, dass die Signalstärke  $X$ , die an einem Ort gemessen wurde, eine normalverteilte Zufallsgröße  $X \sim \mathcal{N}(\mu, \sigma^2)$  mit dem Erwartungswert  $\mu$  und der Standardabweichung  $\sigma$  darstellt. Die Beobachtungswahrscheinlichkeit  $p(\vec{v}_{t,\text{MAC}}|s)$  in einem gegebenen Zustand  $s \in S$  einen bestimmten Beobachtungsvektor zu erhalten, ist dann formal definiert als:

$$p(\vec{v}_{t,\text{MAC}}|s) = \prod_{i=1}^M f_X(o_i; \mu_i, \sigma^2) \quad (4.4)$$

wobei  $o_i \in \vec{v}_{t,\text{MAC}}$  und  $\mu_i \in \vec{v}_s$  gilt. Zudem stellt  $f_X$  die WDF über  $X$  dar und  $M$  repräsentiert die Menge der vorhandenen Messtationen. Insgesamt ergeben sich mittels der oberen Gleichung 4.4 über alle Zustände und Beobachtungen die Einträge der Matrix  $B$ . Als Anfangswahrscheinlichkeit  $\pi$  wird dementsprechend die Verteilung der Beobachtungswahrscheinlichkeiten für die erste Beobachtung über die Menge der Zustände  $S$  betrachtet, also  $\pi = p(\vec{v}_{0,\text{MAC}}|s)$ .

Damit sind alle Modellparameter für  $\lambda$  definiert. Auf Grundlage einer kontinuierlichen Zeitserie an Beobachtungen einer MAC-Adresse  $o = \vec{v}_{1,\text{MAC}}, \dots, \vec{v}_{T,\text{MAC}} \in O^*$  kann nun der Viterbi-Algorithmus genutzt werden, um die wahrscheinlichste Sequenz an verborgenen Zuständen  $q^* = s_1^*, \dots, s_T^* \in S^T$  zu ermitteln (vgl. Abschnitt 2.3.3). Diese Folge stellt den Viterbi-Pfad dar, der gleichzeitig die wahrscheinlichste Trajektorie des entsprechenden Nutzers repräsentiert. Dabei muss beachtet werden, dass das gewählte Zustandsmodell, welches aus diskreten Positionen besteht, den Lösungsraum für die Pfadermittlung auf die Zustandsknoten hin einschränkt und somit die Genauigkeit des Verfahrens beeinflusst. Es gilt, je dichter die Zustandsknoten verteilt sind, desto genauer kann die Lösung werden, desto mehr Zustände werden aber auch benötigt. Wie in Abschnitt 2.3.3 beschrieben, steigt die Laufzeitkomplexität mit der Mächtigkeit der Zustandsmenge im Quadrat, weshalb mehr Zustände die Skalierbarkeit des Verfahrens verschlechtern. Aus diesem Grund wird im nächsten Abschnitt ein Freiflächen-Partikel-Filter für das vorliegende Problem angewendet, der diesen Nachteil ausgleichen soll.

## Partikel-Filter zur Pfadermittlung

Der grundlegende Ablauf eines Partikel-Filters als parameterfreie Variante des rekursiven Bayes-Filters wurde in Unterkapitel 2.3.1 bereits ausführlich behandelt. Danach gliedert sich ein solcher Filter in die Schritte Initialisierung,

Vorhersage und Korrektur, welche im Folgenden für ein SIR-Ansatz konkret umgesetzt werden.

**Initialisierung:** Zu Beginn muss eine temporäre Menge  $\bar{X}_0 = \{x_0^{[1]}, \dots, x_0^{[N]}\}$  von  $N$  Partikeln erzeugt und zufällig über den kompletten begehbaren Bereich des Umgebungsmodells verteilt werden. Im vorliegenden Fall wird hierzu die WDF einer kontinuierlichen Gleichverteilung  $f(x_0^{[i]})$  verwendet. Jedes Partikel  $x_0^{[i]}$  erhält dadurch eine eindeutige Position  $l_0^{[i]}$ . Zudem wird jedem Partikel zufällig eine Anfangsgeschwindigkeit  $v_0^{[i]}$  und eine initiale Orientierung  $o_0^{[i]}$  zugeordnet. Dies geschieht ebenfalls mittels der WDF einer kontinuierlichen Gleichverteilung von 0 bis  $v_{\max}$  für die Geschwindigkeit und von 0 bis  $2\pi$  für die Orientierung. Für die maximal mögliche Geschwindigkeit von Fußgängern innerhalb eines Gebäudes wird hier  $v_{\max} = 10$  km/h festgelegt, was ungefähr einer doppelten mittleren Gehgeschwindigkeit entspricht und daher für eine obere Schranke als geeignet erscheint.

**Vorhersage:** Für jedes Partikel  $x_{k-1}^{[i]}$  aus der vorhandenen temporären Menge  $\bar{X}_{k-1}$  wird im Vorhersage-Schritt zum diskreten Zeitpunkt  $k$  ein neues Partikel  $x_k^{[i]}$  erzeugt. Dieses erhält über das gewählte Bewegungsmodell  $p(x_k^{[i]} | x_{k-1}^{[i]})$  eine neue Position  $l_k^{[i]}$  sowie eine veränderte Geschwindigkeit  $v_k^{[i]}$  und eine andere Orientierung  $o_k^{[i]}$ . Im vorliegenden Fall wird das Bewegungsmodell von Widyawan et al. [221] zur Berechnung des Übergangs von  $k-1$  zu  $k$  verwendet, wodurch sich u.a. die neue Position wie folgt bestimmen lässt:

$$l_k^{[i]} = \begin{bmatrix} l_{k,x}^{[i]} \\ l_{k,y}^{[i]} \end{bmatrix} = \begin{bmatrix} l_{k-1,x}^{[i]} + v_{k-1}^{[i]} \cos(o_{k-1}^{[i]}) \Delta t + n_{k-1} \\ l_{k-1,y}^{[i]} + v_{k-1}^{[i]} \sin(o_{k-1}^{[i]}) \Delta t + n_{k-1} \end{bmatrix} \quad (4.5)$$

Dabei wird ersichtlich, dass die neue Position auf Basis der Zeit  $\Delta t$ , die seit der letzten Vorhersage vergangen ist, und der vorherigen Lage sowie der Geschwindigkeit und Richtung des Partikels  $x_{k-1}^{[i]}$  berechnet wird. Da WLAN-Beobachtungen in unregelmäßigen Abständen auftreten, wird die Vorhersage hier in zeitlich kurze und regelmäßige Segmente auf Basis einer konstanten Schrittlänge von  $\Delta t = 0,8$  Sekunden aufgeteilt und solange wiederholt, bis eine konkrete Beobachtung vorliegt. Dadurch bleiben die Partikel flexibel, auch wenn längere Zeit keine Beobachtung gemacht wird. Im obigen Modell wird zudem ein optionaler Rauschparameter  $n_{k-1}$  eingeführt, der im vorliegenden Fall jedoch vernachlässigt werden soll. Neben der Position muss noch die Geschwindigkeit und die Orientierung in jedem Vorhersage-Schritt aktualisiert werden. Dies geschieht anhand des Bewegungsmodells für jeden Partikel über die folgenden Gleichungen:

$$v_k^{[i]} \sim |\mathcal{N}(v_{k-1}^{[i]}, 1 \text{ ms}^{-1} \Delta t)| \quad (4.6)$$

$$o_k^{[i]} \sim \mathcal{N}\left(o_{k-1}^{[i]}, 2\pi - \arctan\left(0.5 \sqrt{v_{k-1}^{[i]}} \Delta t\right)\right) \quad (4.7)$$

wobei  $\sim$  hier eine zufällige Ziehung aus der Normalverteilung  $\mathcal{N}(\mu, \sigma^2)$  mit Mittelwert  $\mu$  und der Standardabweichung  $\sigma$  darstellt. Nach der Vorhersage folgt die Korrektur der temporären Partikelmenge anhand der Beobachtung. Dieser Vorgang wird beim SIR Ansatz in die beiden Schritte *Importance Sampling* und *Resampling* aufgeteilt.

**Importance Sampling:** Der Beobachtungsvektor  $\vec{v}_{k,\text{MAC}}$  wird als Messung  $z_k$  in die aktuelle Partikelmenge  $\bar{X}_k$  integriert, indem der *importance factor* berechnet wird. Dieser kann, wie in Abschnitt 2.3.1 erläutert, als individuelles Gewicht  $w_k^{[i]}$  eines Partikels  $x_k^{[i]}$  aufgefasst werden und es gilt:  $w_k^{[i]} = p(z_k | x_k^{[i]})$ . Im vorliegenden Fall berechnet sich das Gewicht durch die Verteilung der Positionsschätzung  $(\mu, \sigma)$ , die über den oben beschriebenen WLAN-Fingerprinting-Ansatz gewonnen wird. Dabei sei  $\mu$  die mittlere Position und  $\sigma$  der durchschnittliche Positionsfehler, die bei einer Beobachtung  $z_k$  gemessen wurden. Zur Bestimmung von  $\sigma$  wird der Laplace-Fehlerschätzer von Marcus [135] auf Grundlage der RSS-Werte aus der entsprechenden Beobachtung verwendet. Damit kann das Partikelgewicht direkt aus der WDF des Fehlerschätzers für die aktuelle Position des jeweiligen Partikels  $l_k^{[i]}$  ausgelesen werden und es gilt:

$$w_k^{[i]} = p(z_k | x_k^{[i]}) = \text{WDF}_{\mu, \sigma}^{\text{Laplace}}(l_k^{[i]}) \quad (4.8)$$

Zuletzt müssen die ermittelten Partikelgewichte noch normalisiert werden, so dass ihre Gesamtsumme wieder 1 ergibt. Danach folgt der im SIR-Ansatz wichtige Resampling-Schritt.

**Resampling:** Aus der vorliegenden Partikelmenge  $\bar{X}_k$  werden unter Berücksichtigung der Partikelgewichte eine gleich große Anzahl von  $N$  Partikeln gezogen, welche die neue Menge  $X_k$  darstellen. Darin sind Partikel mit höherem Gewicht häufiger vertreten, als Partikel, die ein niedriges Gewicht haben. Partikel mit dem Gewicht 0 sind nicht mehr vorhanden. Die neue Menge  $X_k$  stellt also die wahrscheinlichste Positionsschätzung des mobilen Geräts dar und wird als Ausgangslage für den nächsten Vorhersage-Schritt verwendet.

Damit sind alle Schritte des vorliegenden Partikel-Filters definiert. Durch die iterative Verarbeitung der Beobachtung mittels Vorhersage und Korrektur entsteht über die Zeit ein Bewegungspfad, der für das beobachtete Gerät am wahrscheinlichsten ist. Eine Position zum Zeitpunkt  $k$  kann bspw. über den Schwerpunkt der Partikelmenge  $X_k$  bestimmt werden. Aufgrund der sequentiellen Verarbeitung aller  $N$  Partikel in  $X_k$  weist der Partikel-Filter pro Beobachtung eine lineare Laufzeitkomplexität von  $\mathcal{O}(N)$  auf. Damit skaliert dieser Ansatz besser als der Viterbi-Algorithmus, solange  $N \ll |S|^2$  gilt. Dies ist nicht immer der Fall, wie einige Arbeiten zeigen. So werden bspw. in [123] und [153] 400 und mehr Partikel in einem ähnlichen SIR-Ansatz benötigt, um eine akzeptable Genauigkeit bei der Pfadermittlung zu erreichen. Diese hohe Anzahl schmälert den Vorteil der Skalierbarkeit bei vergleichbarer Genauigkeit. Zur Problemlösung wird im Folgenden eine neuartige Variante des eben beschriebenen Filters vorgestellt.

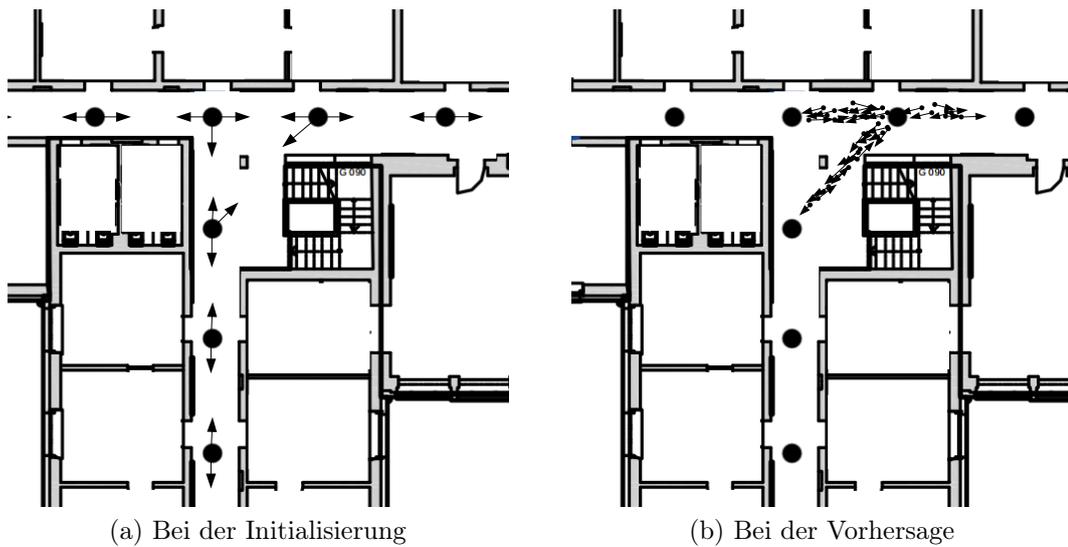


Abbildung 4.8: Darstellung der Lage, Orientierung und Bewegung der Partikel des sPF auf Basis des oben beschriebenen Zustandsmodells.

### 4.2.3 Der State-Partikel-Filter als hybride Lösung

Ziel ist es einen Partikel-Filter zu entwerfen, der mit weniger Partikeln eine vergleichbar hohe Genauigkeit bei der Pfadermittlung erreicht. Dadurch soll eine verbesserte Skalierbarkeit erreicht werden, die zur Erfassung von Nutzerströmen in realen Szenarien essentiell ist. Zu diesem Zweck wird hier der sogenannte *State-Partikel-Filter* (sPF) eingeführt, der eine hybride Methode aus den zuvor beschriebenen Verfahren darstellt. Dabei wird das feste Zustandsmodell des HMM mit dem Bewegungsmodell für Freiflächen von Widyawan et al. kombiniert. Der sPF unterscheidet sich daher vom vorherigen SIR-Partikel-Filter insbesondere im Initialisierungs- und Vorhersage-Schritt. Diese gestalten sich nun wie folgt:

**Initialisierung:** Anstatt Partikel zufällig über die gegebene Freifläche zu verteilen, werden jetzt die vorhandenen  $N$  Partikel gleichmäßig auf die Zustandsknoten der Menge  $S$  platziert, wobei  $N$  ein natürliches Vielfaches von  $|S|$  sein muss. Dabei erhält jedes Partikel wieder eine Anfangsgeschwindigkeit  $v_0^{[i]}$  und eine initiale Orientierung  $o_0^{[i]}$ . Während  $v_0^{[i]}$  wie zuvor über die WDF der kontinuierlichen Gleichverteilung von 0 bis  $v_{\max}$  bestimmt wird, ist die Orientierung nun auf einen beliebigen sichtbaren Zustandsknoten hin ausgerichtet. Ein Knoten ist dann von einem Partikel aus sichtbar, wenn der direkte Weg frei von Hindernissen ist. Dies wird in Abbildung 4.8a beispielhaft für den Initialisierungs-Schritt dargestellt. Insgesamt werden die Partikel so gezwungen, sich entlang des festen Zustandsgraphen zu bewegen. Damit wird zwar die Bewegungsfreiheit eingeschränkt, aber gleichzeitig ein fehlerhaftes Abgleiten der Partikelwolke von Anfang an verhindert.

**Vorhersage:** Um das vorherige Modell  $p(x_k^{[i]} | x_{k-1}^{[i]})$  für den Zustandsüber-

**Algorithmus 3** Vorhersage-Schritt des State-Partikel-Filters**Require:**  $A$ := Winkelmatrix,  $D$ := Distanzmatrix

---

```

1: for all  $i \in N$  do
2:   Bestimme  $l_k^{[i]}$  mittels Gleichung 4.5
3:   Bestimme  $v_k^{[i]}$  mittels Gleichung 4.6
4:    $\text{idx} \sim A[l_k^{[i]}]$  ▷ Wähle Zustandsknoten
5:    $o_k^{[i]} = A[l_k^{[i]}][\text{idx}]$ 
6:    $\text{dist} = D[l_k^{[i]}][\text{idx}]$ 
7:    $t_n = \text{dist}/v_k^{[i]}$ 
8:   while  $t_n < \Delta t$  do ▷ Falls Knoten erreicht wird
9:      $\Delta t = \Delta t - t_n$  ▷ Update
10:     $l_k^{[i]} :=$  Ort des Knotens
11:     $\text{idx} \sim A[l_k^{[i]}]$  ▷ Wähle neuen Zustandsknoten
12:     $o_k^{[i]} = A[l_k^{[i]}][\text{idx}]$ 
13:     $\text{dist} = D[l_k^{[i]}][\text{idx}]$ 
14:     $t_n = \text{dist}/v_k^{[i]}$ 

```

---

gang der Partikel auf den Zustandsgraphen des HMM zu übertragen, wird die neue Orientierung  $o_k^{[i]}$  eines Partikels beim Vorhersage-Schritt durch die Lage der Zustandsknoten bestimmt, anstatt durch Gleichung 4.7. Das bedeutet,  $o_k^{[i]}$  ist der Winkel zwischen der aktuellen Partikelposition  $l_k^{[i]}$  und einem zufällig ausgewählten sichtbaren Zustandsknoten. Liegen mehrere Zustandsknoten in einem engen Sichtfeld, so wird einfach der nächstgelegene Knoten anvisiert. Falls ein ausgewählter Knoten während der Vorhersage-Phase erreicht wird, ändert der Partikel erneut nach dem gleichen Schema seine Orientierung und steuert einen anderen sichtbaren Zustandsknoten an, solange bis eine Messung vorliegt und der Korrektur-Schritt ausgeführt wird. Der Effekt dieser Modifikation ist, dass sich alle Partikel immer in Richtung eines gewählten Knotens und somit auf festen Bahnen entlang des Zustandsgraphen bewegen. Dies zeigt sich grafisch in Abbildung 4.8b.

Algorithmus 3 fasst die gesamte Prozedur des Vorhersage-Schritts als Pseudocode zusammen. Dabei beinhalten die beiden Matrizen  $A$  und  $D$  die Winkel bzw. Distanzen von allen möglichen Positionen zu allen von dort aus sichtbaren Zustandsknoten. Über den jeweiligen Index ist diese Information zur Laufzeit des Filters auf einfache Weise zugänglich (vgl. Zeile 5 u. 6 in Algorithmus 3). Zudem können die beiden Matrizen für ein komplettes Szenario vorberechnet werden, wodurch bei der Vorhersage kein zusätzlicher Aufwand im Vergleich zum ursprünglichen Ansatz entsteht. Der Korrektur-Schritt des sPF erfolgt wieder über das beschriebene Verfahren von *Importance Sampling* und *Resampling* und läuft damit analog zum SIR-Partikel-Filter ab.

### 4.2.4 Evaluation und Ergebnisse der Pfadermittlung

Die vorgestellten Verfahren sollen nun anhand anerkannter Metriken aus dem Bereich der Indoor-Positionierung nach Liu et al. [130] evaluiert werden. Hierzu wird ein Experiment unter realen Bedingungen durchgeführt, das im Folgenden genauer beschrieben wird.

#### Versuchsaufbau und Implementierung

Aus praktischen Gründen wird für den folgenden Versuch der gleiche Aufbau realisiert, welcher bereits bei der Modellvalidierung in Unterkapitel 3.2.4 verwendet wurde. Dieser ist entsprechend in Abbildung 3.7 dargestellt, wobei fünf WLAN-Monitore einen Bereich von ca. 26.800  $m^2$  in dem dargestellten Universitätsgebäude überwachen. Zur Platzierung der Messknoten wurde der in Abschnitt 4.1.3 beschriebene Trajektorien-basierte Ansatz auf dem Gebäudeplan angewendet und das Ergebnis für die Verteilung berücksichtigt.

Die Messknoten selbst sind durch Laptops realisiert und nutzen als Betriebssystem Linux Debian 2.6. Zudem besitzen sie jeweils eine *Netgear wireless PC card* im Monitor Modus zur Aufzeichnung von IEEE 802.11 Probe Requests, wobei hierzu wieder das Linux-Tool *tcpdump* verwendet wird. Alle Messknoten sind über Ethernet mit dem Internet verbunden und werden über NTP zeitlich synchronisiert. Die Datenaggregation und damit die Bildung des Beobachtungsvektors  $\vec{v}_{t,MAC}$  erfolgt über einen zentralen Server. Wie bei der Modellvalidierung umfasst eine Beobachtung die Zeitspanne von  $\Delta t = 5$  Sekunden. Es wird angenommen, dass dieser Wert einen guten Kompromiss darstellt zwischen unterscheidbaren Beobachtungen und der Entfernung, die ein Nutzer während dieser Zeit zurücklegen kann.

Für die Karteninformation des Partikel-Filters wird wieder eine Bitmap-Repräsentation des vorliegenden Gebäudeplans genutzt. Die Modellparameter für den errechneten Pfadverlust nach Formel 2.1 wurden unter realen Bedingungen empirisch für fünf festgelegte Distanzen bestimmt und belaufen sich im Mittel auf  $A = -33$  dBm,  $n = 3,5$  und  $WAF=2,6$ . Die Zustandsknoten des HMM werden entlang der Gänge mit einem festen Abstand von ca. sechs Metern verteilt. Dieser Wert ergibt sich durch die beschriebene Heuristik aus Abschnitt 4.2.2 bei einer Beobachtungszeitspanne von  $\Delta t = 5$  s und für eine angenommene Durchschnittsgeschwindigkeit von  $v_p = 1,2$  m/s.

Um die Ergebnisse der Positionsbestimmung und der Pfadermittlung bewerten zu können, muss die Grundwahrheit bekannt sein. Daher wird der in Abbildung 4.9 grau dargestellte Bereich des Gebäudes explizit abgelaufen, wobei alle 1,5 m ein aktiver IEEE 802.11 Scan durchgeführt wird. Der resultierende Beobachtungsvektor wird mit der bekannten realen Position gespeichert. Dieses Vorgehen wird mit den folgenden vier handelsüblichen Smartphones durchgeführt: LG Nexus 4, iPhone 4s, Samsung Galaxy S4 und einem Samsung Galaxy Note. Insgesamt liegen damit pro Gerät 124 WLAN-Beobachtungen an unterschiedlichen bekannten Positionen zum Zweck der Evaluation vor.

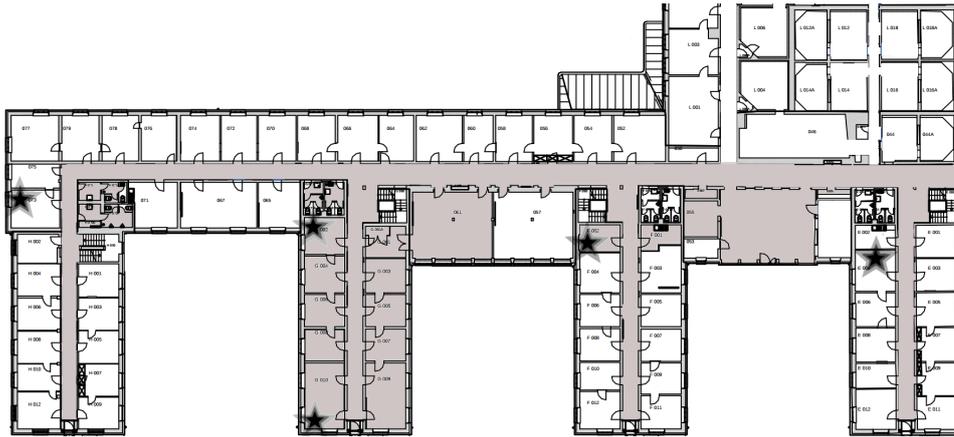


Abbildung 4.9: Veranschaulichung des Versuchsaufbaus: Für den grauen Bereich liegen Beobachtungen mit entsprechender Grundwahrheit vor. Die Sterne symbolisieren die Messknoten.

### Genauigkeit der deterministischen Positionierung und Pfadermittlung

In einem ersten Schritt soll die Genauigkeit der in Abschnitt 4.2.2 beschriebenen deterministischen Positionierung auf Basis des Versuchsaufbaus evaluiert werden. Die dabei gewonnenen Ergebnisse dienen als Vergleichspunkt für die probabilistischen Verfahren, die im späteren Verlauf untersucht werden. Zur Lokalisierung der vier Testgeräte wird eine Radiomap erstellt, die als Referenzpunkte die Positionen der Zustandsknoten enthält. In der Online-Phase werden die Beobachtungsvektoren, die an den 124 bekannten Positionen aufgenommen wurden mit der Radiomap verglichen. Die Positionsbestimmung erfolgt dabei über das gewichtete kNN-Verfahren mit  $k = 4$ .

Abbildung 4.10a zeigt für jedes Testgerät die Fehlerverteilung in Metern auf Basis der euklidischen Distanz zwischen dem Ergebnis der durchgeführten Lokalisierung und dem wahren Aufenthaltsort des Geräts. Dabei wird deutlich, dass die Fehler für alle Geräte ähnlich verteilt sind. Der Interquartilsabstand schwankt lediglich zwischen 5,95 (Nexus 4) und 7,75 m (iPhone). Die genutzte Hardware hat demnach keinen erkennbaren Einfluss auf die gemessenen Signalstärken. Insgesamt liegt das quadratische Mittel des Fehlers bei 7,37 m, was ein relativ schlechtes Ergebnis im Vergleich zu anderen Fingerprinting-Verfahren wie [16] darstellt. Dabei zeigt das Galaxy Note den niedrigsten Wert für den Median von 4,42 m und gleichzeitig die höchste Standardabweichung von 7,5 m. Zudem lassen sich für jedes der vier Geräte sehr große Ausreißer von bis zu 33,82 m beobachten. Zusammenfassend lässt sich festhalten, dass die Positionierung auf Basis von WLAN-Beobachtungen und einem herkömmlichen Fingerprinting-Verfahren wie erwartet zu ungenauen Ergebnissen führt. Im Folgenden soll untersucht werden, was dies für die Pfadermittlung bedeutet.

Aus diesem Grund wird im nächsten Schritt der Fehler über komplette Trajektorien auf Basis der ermittelten Aufenthaltsorte evaluiert. Hierfür werden

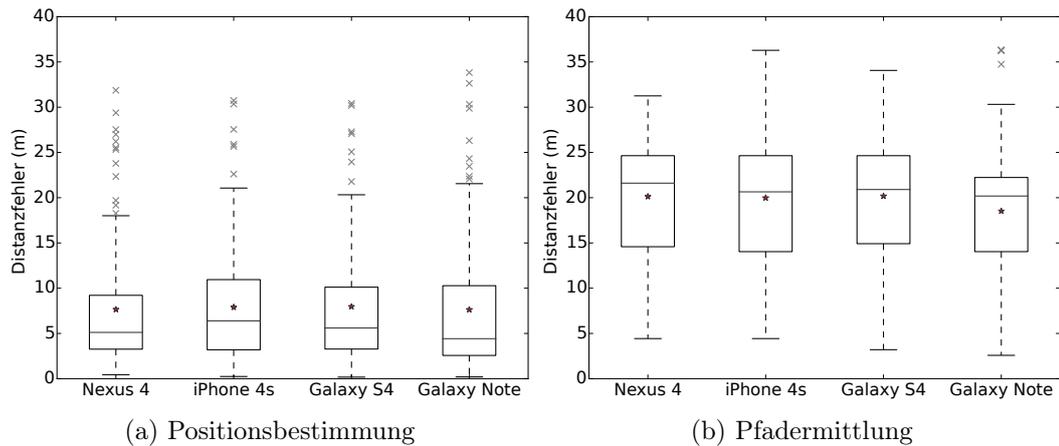


Abbildung 4.10: Fehlerverteilung bei der deterministischen Positionsbestimmung und Pfadermittlung.

zunächst ein Bewegungsgraph über die 124 bekannten Positionen erstellt und  $3 \times 57$  Pfade generiert, die von einer zufällig gewählten Startposition aus entweder ein, zwei, oder drei zufällige Ziele auswählen. Dabei liegen die resultierenden Beobachtungsvektoren in der entsprechenden Reihenfolge vor und repräsentieren den beobachteten Bewegungsablauf eines Nutzers. Auf Basis dieser Abfolgen werden die Positionen über das gewichtete kNN-Verfahren bestimmt und als Bewegungspfad miteinander verbunden. Zuvor müssen die ermittelten Positionen über orthogonale Projektion auf den Bewegungsgraphen projiziert werden, damit die Pfadermittlung mit den später diskutierten Methoden vergleichbar ist.

Zur Evaluation der Genauigkeit wird wieder die Verteilung der auftretenden Distanzfehler zwischen den ermittelten und den echten Pfaden herangezogen. Im Fall von Trajektorien bietet sich hierfür die diskrete Fréchet-Distanz an, die in Abschnitt 2.4.1 beschrieben wurde. Abbildung 4.10b zeigt das entsprechende Ergebnis auf Basis der 171 generierten Pfade und der vier Testgeräten. Auch hierbei zeigen alle Verteilungen ähnliche Muster unabhängig vom genutzten Smartphone, wobei das Galaxy Note mit einem mittleren Fehler von 18,52 m und einem geringeren Interquartilsabstand von 8,16 m noch am besten abschneidet. Auf Basis dieser Ergebnisse lässt sich nicht erkennen, dass die Hardware einen Einfluss auf die Beobachtung nimmt. Dies kann allerdings auch daran liegen, dass hier bewusst aktive Scans durchgeführt wurden, die bei allen mobilen Geräten mit einer vergleichbaren Sendeleistung realisiert werden.

Insgesamt liegt der durchschnittliche Fehler für den kompletten Versuch bei 19,98 m und einer Standardabweichung von 6,42 m. Dieser Wert ist für eine verlässliche Pfadermittlung und damit zur Analyse von Nutzerströmen ungenügend, was aufgrund der genannten Probleme erwartet wurde. Daher soll im Folgenden untersucht werden, inwieweit die probabilistischen Verfahren eine Verbesserung hinsichtlich der Genauigkeit erreichen können.

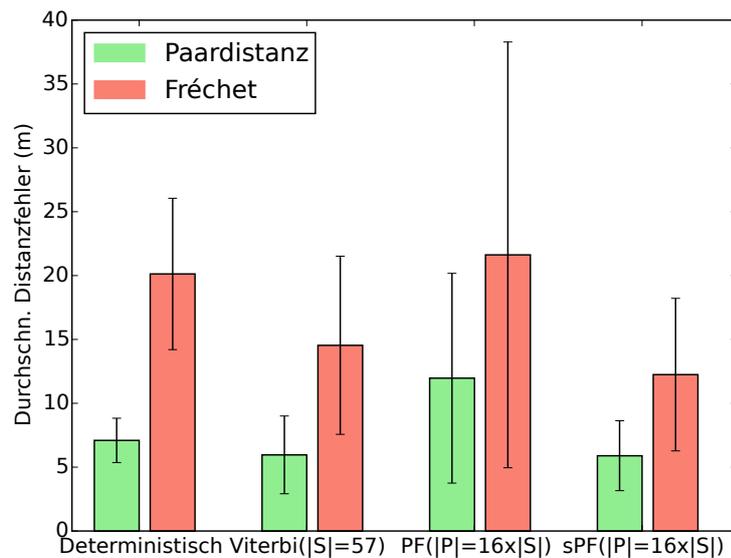


Abbildung 4.11: Vergleich der Genauigkeit aller Methoden anhand der durchschnittlich erreichten Distanzfehler.

### Genauigkeit der probabilistischen Pfadermittlung

Zur Vergleichbarkeit der Ansätze basieren die folgenden Versuche auf den gleichen 171 Pfaden und den entsprechenden Signalbeobachtungen, die ausgehend vom Nexus 4 gemacht wurden. Als Distanzmaß wird neben der diskreten Fréchet-Distanz auch die einfachere Paardistanz (engl.: *pairwise distance*) herangezogen. Diese berechnet sich über den Durchschnitt der paarweisen euklidischen Distanzen zwischen den diskreten Punkten auf dem ermittelten und dem tatsächlichen Bewegungspfad. Hierzu müssen die beiden Trajektorien über dieselbe Anzahl an Punkten verfügen.

Für einen ersten Vergleich der Genauigkeit zwischen allen Verfahren werden folgende Bestimmungen für die Parametereinstellung getroffen: Das Zustandsmodell des HMM besitzt aufgrund des gewählten Abstands von sechs Metern insgesamt  $|S| = 57$  Zustandsknoten. Die Anzahl an Partikeln für den Freiflächen-Partikel-Filter (PF) und dem State-Partikel-Filter (sPF) wird jeweils auf  $|P| = 16 \times |S|$  festgelegt. Die entsprechenden Ergebnisse sind in Abbildung 4.11 dargestellt, wobei die darin angedeuteten Konfidenzintervalle auf einem Konfidenzniveau von 95% beruhen. Es lässt sich offensichtlich feststellen, dass der PF, der im Gegensatz zu den anderen Verfahren keinerlei Einschränkungen bzgl. seiner Bewegung hat, im Durchschnitt die geringste Genauigkeit und die größte Streuung aufweist. Dies ist eine interessante Beobachtung und zeigt, dass eine Einschränkung des Lösungsraums für die Positionierung Vorteile bringen kann, vor allem dann, wenn eine höhere Genauigkeit ohnehin nicht erreicht werden kann. So bietet es sich im vorliegenden Fall an, den Lösungsraum auf ein realistisches Pfadmodell abzubilden, welches die Menge an möglichen Nutzerbewegungen beinhaltet, anstatt jede Koordinate

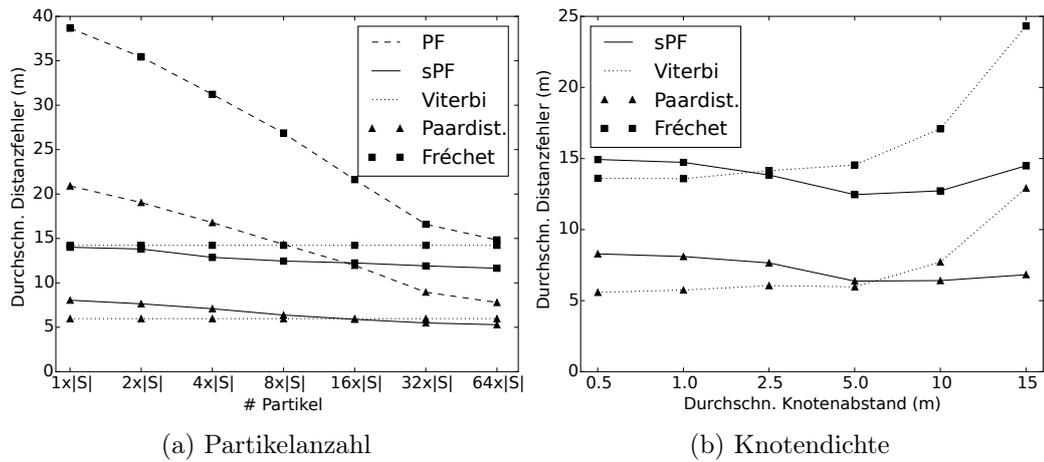


Abbildung 4.12: Einfluss der Partikelanzahl und der Knotendichte auf die Genauigkeit der jeweiligen Verfahren.

der Freifläche für die Positionierung zu berücksichtigen. Zudem lässt sich in Abbildung 4.11 erkennen, dass die beiden anderen probabilistischen Verfahren mit eingeschränktem Lösungsraum (Viterbi und sPF) jeweils eine Verbesserung gegenüber der deterministischen Pfadermittlung erreichen. Im Fall der Fréchet-Distanz sinkt der Fehler um 5,59 m bzw. 7,93 m. So schneidet hier der sPF im Durchschnitt mit einem Fehler von 12,25 m und einer Standardabweichung von 5,97 m am besten ab. Betrachtet man die mittlere Paardistanz beträgt die Verbesserung 1,11 m für Viterbi bzw. 1,19 m im Fall von sPF.

Aufgrund der Tatsache, dass die Partikelanzahl  $|P|$  bzw. die Dichte der Zustandsknoten einen Einfluss auf die Genauigkeit der einzelnen Verfahren nehmen, werden diese beiden Parameter nun in Bezug auf die resultierenden Distanzfehler für Fréchet- und Paardistanz evaluiert. Deshalb wird die Zustandsmenge  $|S| = 57$  zunächst beibehalten und  $|P|$  als Vielfaches von  $|S|$  variiert, wobei in Zweierpotenzen von  $1 \times |S|$  bis  $64 \times |S|$  iteriert wird. Dies wirkt sich nur auf das Ergebnis der beiden Partikel-Filter aus. Für  $|P| = 64 \times |S|$  übersteigt die Partikelanzahl die Mächtigkeit der Zustandsmenge im Quadrat, wodurch die höhere Skalierbarkeit gegenüber Viterbi verloren geht.

Abbildung 4.12a stellt die Ergebnisse dieses Versuchs graphisch dar. Dabei zeigt sich, dass der Fehler in jedem Fall mit steigender Partikelanzahl sinkt, was zu erwarten war. Interessanter ist die Beobachtung, dass im Fall der Fréchet-Distanz der PF konsequent schlechter und der sPF durchgehend besser abschneidet als der Viterbi-Ansatz. Unter Berücksichtigung der Paardistanz zeigt der sPF ab  $|P| = 16 \times |S|$  eine geringere Abweichung als Viterbi. Zudem scheint der sPF robuster zu sein, da die Distanzfehler mit steigender Partikelanzahl in geringerem Maße abnehmen, als beim PF. Die Genauigkeit von sPF wird also weniger durch den Parameter  $|P|$  beeinflusst und zeigt bereits für die geringste Partikelanzahl von  $|P| = |S|$  ähnliche Distanzfehler wie

Viterbi. Dies lässt auf eine hohe Skalierbarkeit für den sPF schließen. Im Gegensatz dazu benötigt der PF eine hohe Partikelanzahl, um eine vergleichbare Genauigkeit wie der Viterbi-Ansatz zu erreichen.

Als nächstes soll die Dichte der Zustandsknoten variiert und  $|P|$  konstant gehalten werden, um den Einfluss des Parameters  $|S|$  zu evaluieren, der sich zusammen mit der Knotendichte verändert. Dies hat wiederum nur eine Auswirkung auf das Ergebnis von Viterbi und sPF. Für die Partikelanzahl wird die Mitte des vorherigen Versuchs gewählt, also  $|P| = 8 \times |S| = 456$ . Die Ergebnisse sind wieder für beide Distanzmaße in Abbildung 4.12b dargestellt, wobei die Knotendichte über den durchschnittlichen Abstand aller benachbarten Zustandsknoten in Metern angegeben wird. Wie erwartet, verringert sich die Genauigkeit des Viterbi-Ansatzes mit geringer werdender Knotendichte, da der Lösungsraum für die Positionierung kleiner und grobgranularer wird. Im Gegensatz dazu zeigt der sPF den geringsten Fehler für beide Distanzmaße bei einem Knotenabstand von ca. 5,0 m. Werden die Zustandsknoten hingegen enger bzw. spärlicher gesetzt, steigt auch der Fehler. Der Grund liegt darin, dass eine sehr hohe Knotendichte zu mehr Bewegungsfreiheit der Partikel führt, was einer Annäherung an den Freiflächen-Partikel-Filter nahe kommt, der in den bisherigen Versuchen schlechter abgeschnitten hat. Eine zu geringe Knotendichte führt wiederum zu einem sehr restriktiven Bewegungsmodell, welches die wahren Nutzerpfade nicht mehr hinreichend abbilden kann. Es muss also ein geeignetes Mittelmaß zwischen den beiden Extrema gefunden werden. Zu diesem Zweck wurde in Abschnitt 4.2.2 eine Heuristik zur Bestimmung des durchschnittlichen Knotenabstands vorgestellt, welche für den vorliegenden Fall eine Distanz von  $d_{i,j} = 6,0$  m zurückliefert. Wie aus Abbildung 4.12b hervorgeht, wird in diesem Bereich auch die höchste Genauigkeit für beide Distanzmaße erreicht, was die Eignung der eingeführten Heuristik bestätigt.

Eine weitere Beobachtung ist, dass der Viterbi-Ansatz bei einer höheren Knotendichte einen geringeren Distanzfehler erreicht als der sPF. Dazu sei allerdings angemerkt, dass hierfür auch mehr Zustandsknoten verarbeitet werden müssen, womit die Laufzeit von Viterbi um die zusätzliche Anzahl an Knoten quadratisch zunimmt. Auf die Komplexität des sPF hat dies hingegen keine Auswirkung, da die Distanz- und Winkelmatrix komplett vorberechnet werden. Das bedeutet, dass bei einer höheren Anzahl an Zustandsknoten fairerweise mehr Partikel verwendet werden können, ohne dass der Skalierbarkeitsvorteil des sPF im Vergleich zu Viterbi geschmälert wird. Dadurch würde sich die Genauigkeit des sPF wiederum erhöhen, was in Abbildung 4.12b durch die festgelegte Anzahl an Partikel nicht abgebildet ist.

Zum Abschluss der Untersuchungen fasst Tabelle 4.2 die Ergebnisse nochmals zusammen, die im besten Fall bzgl. der Genauigkeit erreicht wurden. Dabei stellt die letzte Reihe die gewählten Parameterwerte für  $|S|$  und  $|P|$  dar, falls diese benötigt werden. Insgesamt zeigt sich, dass die probabilistischen Verfahren unter Berücksichtigung der Fréchet-Distanz im Durchschnitt genauere Ergebnisse erzielen als das reine deterministische Fingerprinting. Be-

Verfahren	Determ.	Viterbi	PF	sPF
Durchschn. Fréchet-Distanz	19,98	13,62	14,84	11,65
Standardabw. (Fréchet)	6,42	7,78	9,86	5,98
Durchschn. Paardistanz	7,10	5,58	7,80	5,29
Standardabw. (Paardistanz)	1,74	2,71	5,52	2,33
Gewählte Parameter ( $ S ; P $ )	– ; –	368; –	– ; 3648	72; 3648

Tabelle 4.2: Überblick über die erreichte Genauigkeit bei der Pfadermittlung im besten Fall (in Meter).

trachtet man die Paardistanz, so schneiden Viterbi und sPF immer noch besser ab, aber die Standardabweichung beim Fingerprinting-Verfahren fällt deutlich geringer aus. Die besten Resultate werden durch den sPF erzielt, der im Durchschnitt den geringsten Fehler bei der Pfadermittlung (Fréchet-Distanz) von 11,65 m und die niedrigste Standardabweichung von 5,98 m erreicht. Es sei darauf hingewiesen, dass die Tabelle 4.2 die besten Ergebnisse nur bzgl. der erreichten Genauigkeit darstellt. Daher wurden für  $|S|$  und  $|P|$  relativ hohe Werte gewählt, wodurch die Skalierbarkeit leidet. Deshalb sollen nun weitere Metriken für eine vollständige Bewertung der Verfahren untersucht werden.

### Evaluation weiterer Metriken

Nach Liu et al. [130] existieren neben der Genauigkeit als eines der wichtigsten Prädikate für Indoor-Positionierungssysteme weitere Metriken, wie Präzision, Robustheit oder Kosten, um solche Systeme umfassend zu bewerten. Letztere können im vorliegenden Fall vernachlässigt werden, da alle Verfahren das gleiche Tracking-System nutzen, welches keine Spezialhardware benötigt und generell mit geringen Kosten verbunden ist.

Die Präzision hingegen wird in der Literatur sehr häufig betrachtet. Sie wird meistens als die Standardabweichung des Positionsfehlers aufgefasst. Generell ist die Präzision aber ein Maß dafür, wie einheitlich ein Positionierungssystem arbeitet, weshalb Liu et al. die kumulative Verteilungsfunktion (CDF für engl. *cumulative distribution function*) der Abweichung betrachten. Analog dazu wird im vorliegenden Fall die CDF über die Abweichungen bzgl. der Fréchet-Distanz für alle Verfahren berechnet und zur Darstellung der Präzision in Abbildung 4.13a verwendet. Dabei zeigt sich, dass die probabilistischen Methoden insgesamt präziser sind als das deterministische kNN-Verfahren. Beschreibt man die Präzision als das 90. Perzentil der CDF, so beträgt sie für sPF 20,12 m, was den besten Wert im durchgeführten Experiment darstellt.

Zum Abschluss soll noch die Robustheit der Verfahren verglichen werden. Sie gilt als Maß dafür, wie stark das Ergebnis eines Systems durch fehlerhafte oder fehlende Messwerte beeinflusst wird. Aufgrund von verfälschten RSS-Messungen und willkürlich versendeten Probe Requests ist diese Metrik für die WLAN-basierte Standortverfolgung sehr bedeutend. Zur Evaluation der Ro-

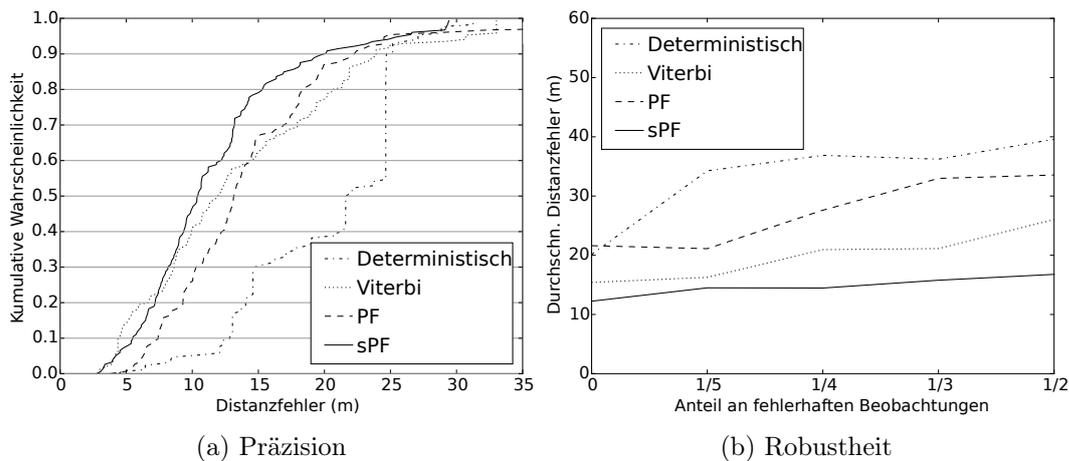


Abbildung 4.13: Vergleich der Verfahren in Bezug auf Präzision und Robustheit.

bustheit werden im Folgenden die Beobachtungen der 50 längsten Nutzerpfade herangezogen und die Abweichung zur Grundwahrheit für jedes Verfahren bestimmt. Anschließend wird dieser Schritt viermal wiederholt, wobei sukzessive ein immer größerer Anteil von  $1/5$  bis  $1/2$  der korrekten Beobachtungen verfälscht wird. Dies geschieht durch Manipulation der entsprechenden Anzahl zufällig ausgewählter Beobachtungsvektoren, deren Einträge auf  $-100$  dBm gesetzt werden, was einer *nicht-Beobachtung* gleichkommt.

In Abbildung 4.13b werden die Ergebnisse dieses Versuchs bzgl. der jeweils erreichten durchschnittlichen Fréchet-Distanz für alle vier Verfahren dargestellt. Darin lässt sich erkennen, dass die Abweichung zur Grundwahrheit erwartungsgemäß bei allen Verfahren ansteigt, wenn mehr und mehr fehlerhafte Beobachtungen vorliegen. Zur Bewertung der Stabilität muss allerdings die Steigung der Kurven berücksichtigt werden. Dabei gilt, je geringer der Anstieg, desto höher ist die Robustheit des Systems. Auf Basis dessen zeigt das deterministische kNN-Verfahren die geringste Robustheit, da bereits bei einem Anteil von  $1/5$  fehlerhafter Beobachtungen ein hoher Anstieg des Distanzfehlers von knapp  $15$  m vorliegt. Im Gegensatz dazu fällt die Steigung beim sPF mit einer maximalen Differenz von  $4,92$  m am geringsten aus, was einen guten Wert darstellt. Zum Vergleich, im Fall von Viterbi bzw. PF liegt die maximale Differenz der Steigung bei  $10,62$  m bzw.  $11,93$  m, womit die Robustheit der beiden Verfahren als moderat bezeichnet werden kann.

#### 4.2.5 Zusammenfassung und Diskussion

Die eben beschriebenen Ergebnisse, die im Rahmen der Evaluation erreicht wurden, legen nahe, dass eine WLAN-basierte passive Standortverfolgung immer noch mit vielen Schwierigkeiten verbunden ist, die zu ungenauen Positionsschätzungen und somit zu großen Abweichungen in der Pfadermittlung führen.

Verfahren	Determ.	Viterbi	PF	sPF
Genauigkeit	19,98	13,62	14,84	11,65
Präzision (90%)	24,69	23,91	22,29	20,12
Komplexität	Gering	Hoch	Moderat	Moderat
Skalierbarkeit	Sehr gut	Gering	Moderat	Gut
Robustheit	Gering	Moderat	Moderat	Gut

Tabelle 4.3: Zusammenfassung der Ergebnisse aller evaluierten Verfahren bzgl. der wichtigsten Metriken nach Liu et al. [130]

Es konnte dargelegt werden, dass deterministische Verfahren, wie der hier verwendete gewichtete kNN-Ansatz, zu einem hohen Distanzfehler sowie zu einer niedrigen Präzision und Robustheit führen. Die untersuchten probabilistischen Methoden zeigten allesamt bessere Ergebnisse, was zu der Schlussfolgerung führt, dass nur solche Verfahren für die WLAN-basierte Standortverfolgung in Betracht kommen. Lediglich die Skalierbarkeit ist beim deterministischen Fingerprinting-Ansatz höher, da die Komplexität in der Online-Phase nach Bildung der Radiomap gering ausfällt. Tabelle 4.3 fasst die wichtigsten Ergebnisse der untersuchten Ansätze und Metriken nochmals zusammen. Darin zeigt sich, dass der hier entwickelte sPF das jeweils beste Resultat in Bezug auf Genauigkeit, Präzision und Robustheit liefert. Zudem werden viel weniger Partikel benötigt als beim herkömmlichen SIR-Partikel-Filter, um eine vergleichbare Genauigkeit zu erreichen. Dies spricht für eine gute Skalierbarkeit des sPF, der sich somit auch in größeren Gebäuden adäquat einsetzen lässt.

Insgesamt hat sich das Konzept eines eingeschränkten Lösungsraums bei der Evaluation bewährt. So zeigte der PF, der sich frei bewegen kann, eine höhere Abweichung von der Grundwahrheit als Viterbi und sPF, die ein diskretes Zustandsmodell verwenden. Erst durch eine sehr hohe Partikelanzahl konnte der hohe Distanzfehler des PF kompensiert werden, was zu einer verringerten Skalierbarkeit führt. Somit ist festzuhalten, dass die Berücksichtigung des kompletten Lösungsraums hier keinen Mehrwert bietet und eher kontraproduktiv ist, da eine Lokalisierung auf Basis von WLAN-Beobachtungen ohnehin nicht die Genauigkeiten für eine exakte Positionierung im Gebäude erreicht.

Zusammenfassend ist festzustellen, dass der sPF unter den evaluierten Methoden am besten abschneidet, auch wenn die absoluten Fehlerdistanzen im Vergleich zu existierenden Indoor-Navigationssystemen insgesamt hoch sind. Die Gründe hierfür werden vor allem in der modellbasierten Radiomap gesehen, die ungenauer als eine empirische Fingerprint-Datenbank ist [16]. Zudem verursachten einige Orte innerhalb der Versuchsumgebung einen erhöhten Positionsfehler. Die Abweichungen der Fréchet-Distanz ergaben sich aufgrund der Gebäudestruktur und der Tatsache, dass einzelne Abweichungen einen starken Einfluss auf den Distanzwert nehmen, bspw. wenn eine fehlerhafte Positionsschätzung zur Auswahl des falschen Korridors führt. Zum Vergleich der Ergebnisse sollten die Versuche für ein komplexeres Gebäude wiederholt werden.

### 4.3 Zusammenfassung

In diesem Kapitel wurde die Extraktion der Ortsinformation aus aufgezeichneten WLAN-Beobachtungen eingehend untersucht. Da die aktuelle Nutzerposition als eine der wichtigsten und primären Kontextinformationen angesehen wird, stellen die hier analysierten und entwickelten Verfahren auch einen Kernbereich der vorliegenden Arbeit dar.

Hierunter zählen zunächst die drei geometrischen Ansätze zur Identifizierung von Hotspots, die im ersten Teil dieses Kapitels vorgestellt wurden, um eine annähernd optimale Platzierung der notwendigen Messstationen zu erreichen. Bei der Evaluation zeigte sich, dass der Trajektorien-basierte Ansatz das beste Ergebnis der Brute-Force-Methode übertreffen konnte, während die anderen Verfahren knapp darunter lagen. Ebenso ist das jeweilige Ergebnis immer abhängig von den Parameterwerten und dem zugrunde liegenden Szenario, wie die durchgeführten Untersuchungen darlegten. Dies erfordert zwar eine geeignete Parameterwahl, schafft aber gleichzeitig eine flexible Einsatzfähigkeit der drei Ansätze für beliebige Umgebungen. Abschließend sei angemerkt, dass zwar nicht das optimale Ergebnis erreicht wird, was aufgrund des NP-harten Problems auch nicht zu erwarten ist, jedoch eine Grundlage geschaffen wird, um geeignete Positionen für Messstationen zur effizienten Erfassung von Nutzerströmen zu finden.

Diese Grundlage wurde im zweiten Teil des Kapitels genutzt, um in einem realen Szenario Messstationen effizient zu platzieren und eine Pfadermittlung auf Basis der aufgezeichneten WLAN-Signale durchzuführen. Neben einem deterministischen WLAN-Fingerprinting-Ansatz basierend auf dem gewichteten kNN-Verfahren wurden hierzu zwei gängige probabilistische Methoden untersucht. Zum einen der Viterbi-Algorithmus, der auf einem HMM mit diskreten Zustandsknoten arbeitet. Zum anderen ein herkömmlicher SIR-Partikel-Filter, welcher ein Bewegungsmodell für Freiflächen nutzt. Zudem wurde der State-Partikel-Filter als eine neuartige hybride Methode dieser beiden Verfahren entwickelt, welcher das diskrete Zustandsmodell mit dem Bewegungsmodell für Freiflächen kombiniert. Alle vier Verfahren wurden unter realen Bedingungen evaluiert. Dabei zeigte sich ganz deutlich, dass ein eingeschränkter Lösungsraum, der sich auf einen vordefinierten Bewegungsgraphen bezieht, für die WLAN-basierte Standortverfolgung besser geeignet ist, als eine Berücksichtigung aller möglichen Positionen. So liefert der hier entwickelte State-Partikel-Filter in Bezug auf Genauigkeit, Präzision und Robustheit die besten Ergebnisse im Vergleich zu den anderen Verfahren. Zudem benötigt dieser deutlich weniger Partikel, was die Skalierbarkeit erhöht und dadurch einen großen Vorteil bei der Erfassung von Nutzerströmen darstellt. Aus diesen Gründen wird der State-Partikel-Filter bevorzugt und auch im folgenden Kapitel bei der Klassifizierung von Nutzerrollen eingesetzt.

## 5 Extraktion der Aktivitäts- und Identitätsinformation

Nach Abowd et al. [2] zählen neben der Zeit und dem Ort bekanntlich auch die Aktivität und die Identität eines Nutzers zu den primären Typen der Kontextinformationen. Da sich das vorangegangene Kapitel bereits eingehend mit der Bestimmung des Aufenthaltsorts und mit der Berechnung von räumlich-zeitlichen Trajektorien beschäftigt hat, wird im Folgenden die Extraktion von Aktivitäts- und Identitätsinformationen genauer untersucht. Dadurch soll es möglich werden, Nutzerströme nicht nur auf einer räumlich-zeitlichen, sondern auch auf einer semantischen Ebene betrachten und analysieren zu können.

Die Aktivitätserkennung selbst stellt ein großes und aktuelles Forschungsgebiet dar, in welchem versucht wird, verschiedene Arten von Nutzeraktivitäten mit einer hohen Genauigkeit zu bestimmen [191]. Neben einfachen Aktivitäten, wie *Stehen*, *Gehen* oder *Laufen* [192] werden auch komplexere bzw. feingranularere Zusammenhänge wie bspw. *Zähneputzen* oder *Staubsaugen* [157] erkannt. Allerdings greifen die meisten Verfahren dabei direkt auf Sensordaten von Smartphones zurück. Dies erfordert wiederum das Einverständnis bzw. die aktive Beteiligung der Nutzer, was außerhalb der verfolgten Ziele liegt. Daher können solche Verfahren im Rahmen der vorliegenden Arbeit nicht angewendet werden. Die Identitätserkennung hingegen erfolgt meist über visuelle Sensoren wie bspw. Kameras und die entsprechenden Algorithmen, welche bestimmte Objekte oder bekannte Gesichter in Bildern wiedererkennen [8, 155]. Auch hierfür sollen im Folgenden andere Möglichkeiten entwickelt werden.

Deshalb wird in diesem Kapitel untersucht, in wie weit sich die beiden Kontextinformationstypen *Aktivität* und *Identität* aus aufgezeichneten WLAN-Signalen extrahieren lassen. Dabei ist es zunächst ausreichend, wenn der aktuelle Bewegungsstatus eines Nutzers als einfache Aktivität zuverlässig erkannt werden kann. Zu dem Zweck werden im ersten Teil bestehende Ansätze untersucht, die eine Unterscheidung zwischen Ruhe- und Bewegungsphasen auf Basis kontinuierlicher RSS-Werte erlauben. Im Anschluss daran werden Erweiterungen zu verwandten Arbeiten vorgeschlagen, welche die Genauigkeit und Generalisierbarkeit der existierenden Methoden verbessern sollen. Die Evaluation erfolgt schließlich auf Basis der in Abschnitt 3.2 vorgestellten Simulationsumgebung und vier Nutzergruppen mit unterschiedlichem Verhalten.

Diese Gruppen werden im zweiten Teil des Kapitels genutzt, um eine Klassifizierung von Nutzerrollen auf Basis der bisher extrahierten Kontextinformationen (Zeit, Ort und Aktivität) durchzuführen. Hierzu werden zunächst

statistische Eigenschaften (Features) aus den vorhandenen Informationen berechnet und gängige Verfahren des maschinellen Lernens zur Klassifizierung untersucht. Die Evaluation erfolgt auf Basis der simulierten Nutzergruppen, wobei die daraus gewonnenen Ergebnisse anhand realer Daten abschließend diskutiert werden. Es sei angemerkt, dass Nutzerrollen in diesem Zusammenhang bestimmte Rollen, wie bspw. Student, Arbeiter oder Besucher bezeichnen, die mit gewissen Zielen und Aktivitäten eng verknüpft sind. Nach Fearon [59] stellt die Zugehörigkeit zu einer bestimmten Rolle mit den jeweiligen Eigenschaften und Aktivitäten die Identität einer Person dar. Deshalb kann die Bestimmung von Nutzerrollen als Extraktion der Identitätsinformation aufgefasst werden. Die Hauptinhalte und alle Abbildungen, die in diesem Kapitel vorgestellt werden, wurden bereits in [180] publiziert. Wie aus Abschnitt 1.3 hervorgeht, stammen dabei alle inhaltlichen Schwerpunkte vom Autor der vorliegenden Arbeit.

## 5.1 Bestimmung von Ruhe- und Bewegungsphasen

Die Bestimmung des Bewegungsstatus wird im Rahmen der klassischen Aktivitätserkennung sehr häufig untersucht [12]. Zum einen, weil die bloße Unterscheidung zwischen Ruhe- und verschiedenen Bewegungsphasen eine Basisinformation zur Bewertung der gegenwärtigen Situation eines Nutzers liefert. Zum anderen kann der Bewegungsstatus über verschiedene Smartphone-Sensoren wie bspw. Accelerometer und Gyroskop relativ einfach und zuverlässig erkannt werden. Dabei wäre es theoretisch auch möglich, solche eindeutigen Aktivitäten wie *Gehen* oder *Stehen* über kontinuierliche Positionsschätzungen zu ermitteln. In der Praxis scheitert dies jedoch häufig an den vorhandenen Positionierungssystemen, welche nicht die erforderliche Genauigkeit aufweisen.

Wie im vorherigen Kapitel 4 gezeigt, trifft dies auch bei der probabilistischen Pfadermittlung auf Basis von WLAN-Beobachtungen zu. So ist die dort erreichte Lokalisierungsgenauigkeit nicht ausreichend, um den Bewegungsstatus einer Person zu einem bestimmten Zeitpunkt verlässlich bestimmen zu können. Aus diesem Grund wird im vorliegenden Abschnitt ein anderes Verfahren zur Bestimmung von Ruhe- und Bewegungsphasen vorgeschlagen, das auf den reinen Rohdaten der WLAN-Beobachtungen basiert. Dabei werden die Zeitserien der aufgezeichneten RSS-Werte über ein Zeitfensterverfahren (engl.: *sliding window*) auf vorhandene Varianzen überprüft. Ziel ist es, klar abgegrenzte Bereiche von Phasen mit hoher und niedriger Varianz zu entdecken, die eine Schlussfolgerung über den Bewegungsstatus des Nutzers in diesen Zeiträumen zulassen. Eine Serie von RSS-Werten mit einer hohen Streuung gilt dabei als Indikator für eine stattfindende Bewegung im Zeitverlauf, während beständigere RSS-Werte auf eine entsprechende Ruhephase hindeuten. Die Idee dahinter ist, dass sich die beobachteten Empfangssignalstärken während einer Bewe-

gung stärker verändern müssen als wenn das entsprechende Gerät auf seiner Position verweilt. Allerdings ist dies kein grundlegend neuer Ansatz, wie im folgenden Abschnitt aufgezeigt wird.

### 5.1.1 Verwandte Arbeiten

Als eines der ersten Systeme, bei denen der Bewegungsstatus auf Basis von WLAN-Beobachtungen ermittelt wurde, ist LOCADIO von Krumm und Horvitz [116] zu nennen, das im Jahr 2004 vorgestellt wurde. Anders als in der vorliegenden Arbeit wird dabei lediglich das Signal desjenigen Access Points betrachtet, welches von einem mobilen Gerät zu einem bestimmten Zeitpunkt am stärksten empfangen wird. Auf Basis dessen wird ähnlich wie in diesem Kapitel ein Zwei-Zustands-HMM definiert und über die Varianz des Signals eine Entscheidung bzgl. des Bewegungsstatus getroffen. Die gewonnenen Informationen gehen in ein weiteres HMM ein, welches zur Lokalisierung verwendet wird (vgl. Abschnitt 4.2.1). Nach Angaben der Autoren wird der Bewegungsstatus mit einer Klassifizierungsgenauigkeit von 87% erreicht.

Lee und Chen [120] greifen ebenfalls den oben beschriebenen Ansatz zur Bestimmung von Bewegungsphasen auf und nutzen diese Information zur deterministischen Lokalisierung, wenn eine Menge an nicht unterscheidbaren Kandidaten zur Positionierung vorliegt. Ähnlich zur vorliegenden Arbeit werden die Zeitserien der RSS-Werte in einem Sliding-Window-Verfahren verarbeitet und geglättet, um eindeutige Veränderungen im zeitlichen Verlauf der Empfangssignalstärken zu erkennen. Allerdings betrachten die Autoren wieder nur einzelne Signalkurven und nutzen zur Glättung den einfachen kumulativen Mittelwert. Ein dazu sehr ähnlicher Ansatz wird von Shen et al. [188] vorgestellt, der ebenfalls den Bewegungsstatus ermittelt, um ambivalente Kandidaten für eine raumgenaue Positionierung zu unterscheiden. Das Verfahren läuft analog über die Betrachtung der zeitlichen Verläufe der Empfangssignalstärken ab. Mit der Information über den Bewegungsstatus erreichen die Autoren eine Verbesserung ihres Positionierungssystems von insgesamt 10% im Vergleich zu herkömmlichen Methoden.

Muthukrishnan et al. [147] untersuchen ebenfalls verschiedene Methoden, um Ruhezeiten auf Basis aufgezeichneter WLAN-Beobachtungen zu bestimmen. Analog zu den beiden vorherigen Ansätzen sollen diese Informationen dazu dienen, klassische signalbasierte Lokalisierungsverfahren wie WLAN-Fingerprinting zu verbessern. Dabei erreichen die Autoren ein Klassifizierungsergebnis von über 90% für Präzision und Sensitivität, wobei probabilistische Ansätze nicht betrachtet werden. King und Kjærgaard [106] präsentieren ein System, welches je nach Bewegungsstatus des Nutzers adaptiv zwischen aktiven und passiven WLAN-Scans wechselt, um Interferenzen bei der Signalübertragung gering zu halten. Zur Bestimmung von Ruhe- und Bewegungsphasen untersuchen die Autoren zunächst verschiedene Einflussfaktoren wie den Hardwareeinsatz oder die Umgebung. Zudem werden wie in der vorliegenden Arbeit

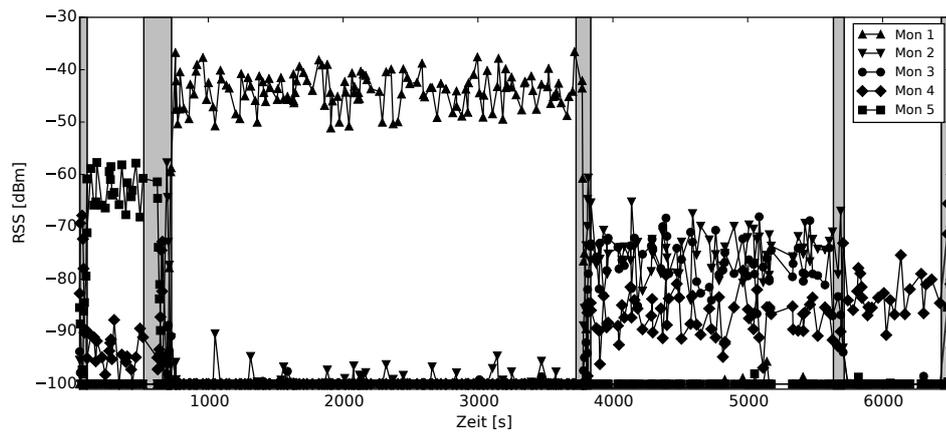
diverse Methoden evaluiert, welche die Eigenschaften der vorliegenden Empfangssignalstärken ausnutzen. Die Autoren verwenden außerdem ein zu LOCA-DIO vergleichbares HMM, welches die beste Option aus ihrer Sicht darstellt. So erreichen sie unter realen Bedingungen eine Richtig-Positiv-Rate von 78,13%. Anders als in der vorliegenden Arbeit werden allerdings nur RSS-Werte von einem Access Point betrachtet.

Wind et al. [230] analysieren den Bewegungsstatus über bekannte Aufenthaltsorte, an denen der Nutzer für eine bestimmte Zeit stehen geblieben ist. Zu diesem Zweck wird in regelmäßigen Abständen aktiv nach Access Points gesucht und die Start- bzw. Endzeit für die entsprechenden Orte ermittelt. Über diese Information lassen sich dann Ruhephasen für eine ganze Umgebung bzw. über einen kompletten Tag herleiten. Andere Arbeiten verwenden aufgezeichnete WLAN-Beobachtungen für weiterführende Aktivitätserkennungen. So stellen Quin et al. [165] das *Mo-Fi System* vor, welches vier definierte Präsenzaktivitäten (*draußen*, *vorbeigehen*, *springen* und *beschäftigt*) mit einer Erkennungsrate von insgesamt 87.4% aus den gesammelten WLAN-Daten korrekt extrahiert. Ruiz-Ruiz et al. [171] analysieren verschiedenen Aktivitäten und auch Rollen (vgl. Abschnitt 5.2.1) für eine Krankenhausumgebung, wobei die Unterscheidung zwischen Ruhe- und Bewegungsphasen nicht im Detail untersucht wird.

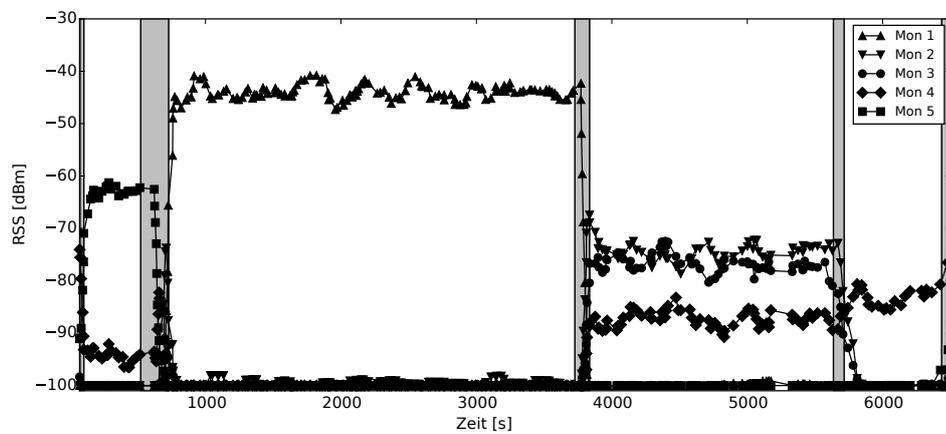
Wie die erwähnten Arbeiten zeigen, wurden bereits mehrere Verfahren untersucht, um den Bewegungsstatus eines Nutzers über WLAN-Beobachtungen zu ermitteln. Im Folgenden wird ein ähnlicher Ansatz vorgestellt, welcher sich jedoch in bestimmten Aspekten von den bestehenden Methoden unterscheidet. Zudem werden weitere Distanzfunktionen sowie deterministische und probabilistische Methoden zur Differenzierung zwischen Ruhe- und Bewegungsphasen evaluiert. Nach heutigem Kenntnisstand des Autors der vorliegenden Arbeit wurden derartige Untersuchungen bisher nicht durchgeführt.

### 5.1.2 Beiträge zur Bestimmung des Bewegungsstatus

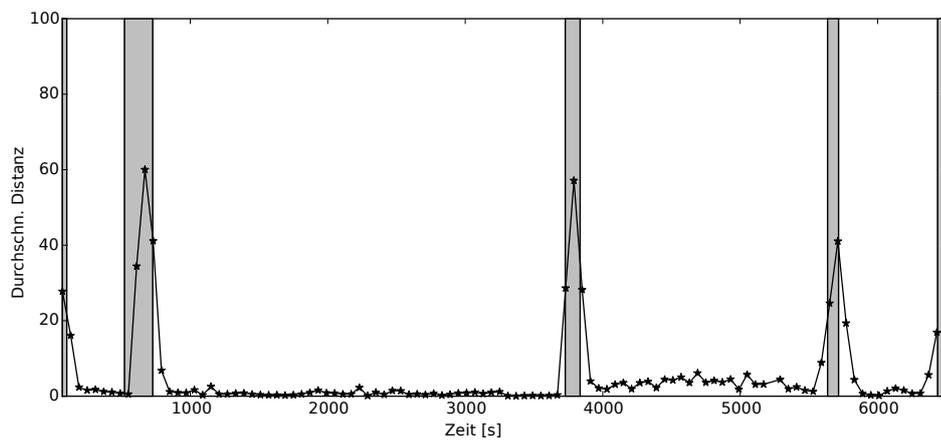
Der vorliegende Ansatz verfolgt das Ziel, eindeutige Veränderungen in einer Zeitserie von WLAN-Beobachtungen zu erkennen, um letztendlich zwischen Ruhe- und Bewegungsphasen eines Nutzers möglichst genau unterscheiden zu können. Hierzu werden die gemessenen Rohdaten zunächst in zwei Schritten weiterverarbeitet: Als erstes findet eine Kurvenglättung zur Elimination von Ausreißern statt, und anschließend erfolgt eine Distanzberechnung zur Bestimmung, wie stark sich die RSS-Werte im Zeitverlauf ändern. Zur besseren Veranschaulichung des Verfahrens werden die Rohdaten und das jeweilige Resultat der beiden Verarbeitungsschritte in Abbildung 5.1 an einem Beispiel illustriert. Hierzu wurden an fünf Monitoren (Mon 1-5) Probe Requests eines Nutzers aufgezeichnet, der sich für ca. zwei Stunden im überwachten Bereich aufgehalten hat und dabei in vier Räumen unterschiedlich lange geblieben ist. Die grau hinterlegten Bereiche stellen die tatsächlichen Bewegungsphasen des Nutzers



(a) Rohdaten



(b) Nach Kurvenglättung



(c) Nach Distanzberechnung

Abbildung 5.1: Darstellung des Verfahrens zur Bestimmung von Ruhe- und Bewegungsphasen anhand eines Beispiels mit fünf Messstationen. Graue Bereiche markieren die realen Bewegungsphasen, weiße Bereiche die Ruhezeiten des beobachteten Nutzers.

dar, während die weißen Flächen die echten Ruhezeiten symbolisieren.

Anders als in verwandten Arbeiten, kommt für die Kurvenglättung im vorliegenden Fall der Savitzky-Golay-Filter [174] zum Einsatz. Der Grund dafür ist, dass dieser Filter die hier wichtigen Eigenschaften wie die Verteilung von Maxima und Minima während der Glättung beibehält. Andere gängige Verfahren wie der gleitende Mittelwert oder auch der Kalman-Filter haben in ersten Versuchen auf den vorhandenen Testdaten zu mehr verfälschten Werten und damit zu schlechteren Ergebnissen geführt. Dieser Schritt eliminiert vorhandene Ausreißer und minimiert letztendlich das Rauschen in den RSS-Werten der ursprünglichen Zeitserie. Dadurch lassen sich die tatsächlichen Änderungen im Zeitverlauf, die durch eine Bewegung des Nutzers hervorgerufen werden, deutlicher hervorheben. Dieser Effekt kann durch den Vergleich der beiden Abbildungen 5.1a und 5.1b beispielhaft beobachtet werden.

Anschließend erfolgt die Berechnung einer Distanzkurve auf Basis der geglätteten Zeitserie. Hierzu wird ein klassischer Sliding-Window-Ansatz genutzt, welcher über eine Fenstergröße  $t_w$  und einer Schrittweite  $s = t_w/2$  zu überlappenden Bereichen führt. Für jeden dieser Fensterbereiche wird die durchschnittliche Distanz als Maß für die Streuung der darin vorhandenen RSS-Werte mittels einer gewählten Distanzfunktion errechnet. Dabei gilt, höhere Distanzen stellen letztendlich eine erhöhte Varianz dar und lassen eine echte Bewegung des Nutzers vermuten. Das Ergebnis dieses Schritts ist eine Distanzkurve, wie sie in Abbildung 5.1c für das gegebene Beispiel dargestellt ist. Darin lässt sich erkennen, dass hohe Distanzwerte um die grau hinterlegten Bereiche herum erzielt werden, was ein adäquates Ergebnis repräsentiert. Allgemein gilt, je genauer und eindeutiger die tatsächlichen Bewegungsphasen durch erhöhte Distanzwerte repräsentiert werden, desto leichter ist anschließend die Unterscheidung zwischen Ruhe- und Bewegungsphasen und desto besser fällt das Klassifizierungsergebnis aus. Aus diesem Grund werden im vorliegenden Fall alle Signalstärken, die an den vorhandenen Monitoren gemessen wurden, zur Berechnung der Distanzkurve herangezogen und nicht wie bei LOCADIO [116] nur das jeweils stärkste Signal. Laut King und Kjærgaard [106] sollte dies zu eindeutigeren und stabileren Mustern führen. Außerdem werden im Folgenden mehrere Distanzfunktionen für die Bestimmung des Bewegungsstatus untersucht und nicht nur die in verwandten Arbeiten gebräuchliche Varianz der Empfangssignalstärken.

## Distanzfunktionen

Zur Berechnung der Distanzkurve kann prinzipiell jede Funktion verwendet werden, die einen Skalar als Ähnlichkeitswert zurückliefert. Insbesondere bieten sich die gängigen Ähnlichkeitsmaße für Vektoren wie bspw. die Euklidische Distanz, der Cosinus- bzw. Dice-Koeffizient oder der Jaccard-Index an [79]. Unter allen Möglichkeiten haben sich die folgenden vier Funktionen als besonders geeignet herausgestellt, da hiermit bessere Ergebnisse erzielt wurden:

- **Kosinus-Ähnlichkeit:** Repräsentiert die Ähnlichkeit zweier  $n$ -elementiger Beobachtungsvektoren  $\vec{u}$  und  $\vec{v}$ , indem der Kosinus des Winkels zwischen den beiden wie folgt berechnet wird:

$$\cos(\vec{u}, \vec{v}) = \frac{\sum_{i=1}^n \vec{u}_i \cdot \vec{v}_i}{\sqrt{\sum_{i=1}^n \vec{u}_i^2} \cdot \sqrt{\sum_{i=1}^n \vec{v}_i^2}} \quad (5.1)$$

Damit wird ein Ähnlichkeitswert bzgl. der Richtung zweier Vektoren beschrieben, welcher zwischen null (orthogonal, also komplett verschieden) und eins (gleichgerichtet) liegt. Dieses Maß wurde bereits in einer früheren Arbeit [133] zur Berechnung der Übereinstimmung erfolgreich eingesetzt. Im vorliegenden Fall wird der Durchschnitt der Kosinus-Ähnlichkeit aller Paare von Beobachtungsvektoren berechnet, die sich in einem Fensterbereich befinden.

- **Euklidische Distanz:** Berücksichtigt zur Bestimmung der Ähnlichkeit den elementweisen Unterschied der zwei  $n$ -elementigen Beobachtungsvektoren  $\vec{u}$  und  $\vec{v}$  durch Berechnung der Distanz  $\delta(\vec{u}, \vec{v}) = \sqrt{\sum_{i=1}^n (\vec{u}_i - \vec{v}_i)^2}$ . Im Gegensatz zu Muthukrishnan et al. [147], die lediglich den Euklidischen Abstand zwischen der ersten und letzten Beobachtung in einem Fensterbereich betrachten, wird im vorliegenden Fall jeweils die durchschnittliche Euklidische Distanz zwischen allen Paaren von Beobachtungsvektoren ermittelt, was einen robusteren Wert liefert.
- **Spannweite:** Ist ein einfaches Maß für die Streuung und berechnet sich aus den maximalen Differenzen der RSS-Werte, die von jedem Monitor innerhalb eines Zeitfensters gemessen werden. Im vorliegenden Fall wird die Summe der einzelnen Spannweiten  $\sum_{i=1}^m (x_{i_{\max}} - x_{i_{\min}})$  für  $m$  Monitore berechnet. Im Gegensatz zu den vorherigen Distanzfunktionen werden hier nicht mehr die Beobachtungsvektoren, sondern die Messwerte der einzelnen WLAN-Monitore separat betrachtet.
- **Varianz:** Repräsentiert ebenfalls die Streuung der RSS-Werte, die von jedem Monitor innerhalb eines Zeitfensters gemessen werden. Im vorliegenden Fall wird der Durchschnitt der einzelnen  $m$  Varianzen für ein Zeitfenster errechnet. Die Varianz wird sehr häufig in der Literatur zur Bestimmung des Bewegungsstatus eingesetzt und soll daher als Vergleichsmetrik dienen. Gelegentlich werden auch andere ähnliche Streumaße wie bspw. die Standardabweichung verwendet [147].

Die Evaluation dieser vier Distanzfunktionen wird im späteren Verlauf (s. Unterkapitel 5.1.3) genauer vorgestellt. Um eine feingranulare Distanzkurve zu erhalten, muss die Fenstergröße  $t_w$  entsprechend kurz gewählt und ein Distanzwert pro Zeitfenster berechnet werden. Da WLAN-Beobachtungen in der Realität aufgrund der niedrigen Probe-Frequenz im Durchschnitt nur alle 45

Sekunden vorliegen (vgl. Abschnitt 2.1.2), sind in kürzeren Zeitfenstern oftmals keine Beobachtungen vorhanden. Die fehlenden Werte müssen dann entsprechend über die angrenzenden Distanzwerte interpoliert werden, um eine konstant hohe Abtastrate entsprechend der Schrittweite von  $s = t_w/2$  zu gewährleisten. Liegt eine solche Distanzkurve vor, muss auf Basis des Kurvenverlaufs entschieden werden, ob der Nutzer an einem bestimmten Zeitpunkt steht oder sich bewegt. Diese binäre Entscheidung kann entweder deterministisch oder probabilistisch erfolgen.

### Deterministische Entscheidung

Für eine deterministische Entscheidung muss ein eindeutiger Schwellwert  $\theta$  gewählt werden. Liegt ein Distanzwert für einen Zeitpunkt  $t$  oberhalb von  $\theta$ , so wird der Nutzer als *gehend* und im anderen Fall als *stehend* klassifiziert. Ein solcher Ansatz ist demnach sehr leicht umzusetzen und wird bspw. auch in [147] und [188] evaluiert. Allerdings ist die Wahl eines geeigneten Schwellwerts nicht trivial und auch als kritisch anzusehen, da das Klassifizierungsergebnis direkt von  $\theta$  abhängt. Im Rahmen der vorliegenden Arbeit werden deshalb zwei Varianten untersucht, um  $\theta$  möglichst adäquat zu bestimmen. Zum einen über empirische Versuche, indem auf Basis einer größeren Menge von Nutzerbewegungen mit bekannter Grundwahrheit der optimale statische Schwellwert  $\theta$  ermittelt wird, welcher das beste Klassifizierungsergebnis zurückliefert. Zum anderen wird ein dynamischer Schwellwert  $\theta_{\text{dyn}}$  genutzt, der über den Durchschnitt der vorliegenden lokalen Maxima in der jeweiligen Distanzkurve ermittelt wird. Dadurch soll eine verbesserte Generalisierbarkeit erreicht werden, die bei einem empirisch ermittelten Schwellwert kaum gegeben ist. Die Evaluation der beiden Varianten wird später in Abschnitt 5.1.3 vorgestellt. Ganz ohne Schwellwert kommt hingegen ein probabilistisches Verfahren aus.

### Probabilistische Entscheidung

Zur Umsetzung der probabilistischen Entscheidung wird in Anlehnung an das LOCADIO System [116] ein HMM  $\lambda = (S; O; A; B; \pi)$  mit den beiden Zuständen  $S = \{s, b\}$  erstellt, wobei  $s$  den Status *Stehen* und  $b$  den Zustand *Bewegung* symbolisiert. Im Gegensatz zu LOCADIO wird im vorliegenden Fall die berechnete Distanzkurve als Zeitserie der Beobachtungen  $O$  herangezogen. Außerdem wird eine höhere Wahrscheinlichkeit für Zustandsänderungen in realen Szenarien angenommen<sup>1</sup>:

$$A = \begin{pmatrix} a_{s,s} & a_{s,b} \\ a_{b,s} & a_{b,b} \end{pmatrix} = \begin{pmatrix} 0.99 & 0.01 \\ 0.01 & 0.99 \end{pmatrix}$$

---

<sup>1</sup>Zum Vergleich: Bei LOCADIO wird zur Bestimmung der Matrix  $A$  von einem Arbeiter ausgegangen, der an einem 8-Stunden-Tag zehn Ortswechsel durchführt.

Dadurch wird ein Wechsel des Bewegungszustands von  $s$  nach  $b$  und andersherum leichter möglich. Dies erscheint realistischer und erhöht die Generalisierbarkeit, wodurch sich auch andere Verhaltensweisen von Personen besser klassifizieren lassen.

Zur Berechnung der Beobachtungswahrscheinlichkeiten  $B$  wird im vorliegenden Fall der beobachtete Distanzwert als normalverteilte Zufallsgröße  $X_\rho$  für einen Zustand  $\rho \in S$  angenommen:  $X_\rho \sim \mathcal{N}(\mu_\rho, \sigma_\rho^2)$ . Der Mittelwert  $\mu_\rho$  und die Standardabweichung  $\sigma_\rho$  werden empirisch auf den vorhandenen Daten ermittelt, indem separat für beide Zustände die Verteilung der resultierenden Distanzwerte betrachtet wird. Wie erwartet, liegen die Werte für  $\mu_b$  und  $\sigma_b$  (während der Bewegung eines Nutzers) deutlich höher als für  $\mu_s$  und  $\sigma_s$ , was ein plausibles Ergebnis darstellt. Darauf aufbauend lässt sich über die WDF  $f_{X_\rho}$  für eine Beobachtung  $\text{obs} \in O$  unter der Bedingung, dass sich der Nutzer im Zustand  $\rho$  befindet, die Wahrscheinlichkeit  $p(\text{obs} | \rho)$  angeben und es gilt:

$$p(\text{obs} | \rho) = f_{X_\rho}(\text{obs}; \mu_\rho, \sigma_\rho^2) \quad (5.2)$$

Insgesamt ergeben sich dadurch für alle Zustände und Beobachtungen die Einträge der Matrix  $B$ . Als Anfangswahrscheinlichkeit wird hier  $\pi_b = 0,9$  und  $\pi_s = 0,1$  festgesetzt. Der Grund ist, dass ein Nutzer zu Beginn höchstwahrscheinlich in Bewegung sein muss, weil er oder sie gerade in den beobachteten Bereich einer Messstation gelangt ist. Aufgrund der geringen Beobachtungsfrequenz besteht allerdings eine geringe Restwahrscheinlichkeit, dass der Nutzer bereits zur Ruhe gekommen ist, ehe er von einem WLAN-Monitor erfasst wird. Bei LOCADIO liegen diese Annahmen nicht vor, weshalb dort die Anfangswahrscheinlichkeit  $\pi_b = \pi_s = 0,5$  beträgt.

Auf Basis des eben vorgestellten Modells lässt sich wieder der Viterbi-Algorithmus anwenden, um die wahrscheinlichste Sequenz an verborgenen Zuständen zu bestimmen. Im Gegensatz zur bereits durchgeführten Pfadermittlung (vgl. Abschnitt 4.2.2), bei der mittels Viterbi die wahrscheinlichste Nutzertrajektorie zurückgeliefert wurde, repräsentiert der Viterbi-Pfad im vorliegenden Fall die wahrscheinlichste Abfolge der Bewegungszustände eines Nutzers. Dadurch lässt sich für jeden Zeitpunkt abschätzen, ob sich die entsprechende Person in einer Ruhe- oder Bewegungsphase befindet.

### 5.1.3 Evaluation und Ergebnisse der Aktivitätsbestimmung

Im vorliegenden Abschnitt werden die vorgestellten Beiträge zur Bestimmung des Bewegungsstatus evaluiert und mit existierenden Verfahren verglichen. Hierzu wird auf simulierte Daten zurückgegriffen, da diese die notwendige Grundwahrheit beinhalten, um die Verfahren hinsichtlich ihrer Klassifizierungsgenauigkeit bewerten zu können. Zudem soll untersucht werden, ob die extrahierte Aktivitätsinformation das Bewegungsmodell des in Abschnitt 4.2.3

vorgestellten State-Partikel-Filters adaptiv unterstützen kann, um genauere Trajektorien zu erhalten. Dies ist nach Kenntnisstand des Autors ein neuwertiger Ansatz, da in der betrachteten Literatur bisher lediglich gezeigt wurde, dass deterministische Verfahren wie WLAN-Fingerprinting durch Hinzunahme der Aktivitätsinformation verbessert werden konnten [116, 147, 188].

### Versuchsaufbau und Durchführung

Damit die verwendete Simulation ein möglichst realistisches Szenario abbildet, sollen sich Nutzer in einem realen Gebäude auf natürliche Weise bewegen und dabei ein unterschiedliches Verhalten an den Tag legen, welches ihrer angedachten Rolle entspricht. Zu diesem Zweck werden zunächst vier typische Nutzergruppen für das in Abschnitt 3.2.4 und 4.2.4 betrachtete Universitätsgebäude erstellt:

1. **Mitarbeiter:** Arbeitet täglich im entsprechenden Gebäude und kommt zwischen 8:00 und 10:00 Uhr in sein Büro. Während des Tages wechselt der Mitarbeiter zufällig oft in andere Büros, um dort eine lokale Aufgabe für eine begrenzte Zeitspanne von bis zu 40 Minuten zu erledigen. Zwischen 17:00 und 19:00 Uhr verlässt er das Gebäude. Diese Nutzergruppe simuliert damit mehrere Ruhephasen von mittlerer bis langer Dauer.
2. **Student:** Kommt zwischen 10:00 und 16:00 Uhr in das Gebäude, um eine oder mehrere Vorlesungen zu hören. Der Student geht also in einen Hörsaal und bleibt dort für eine Unterrichtsdauer von 80 bis 120 Minuten. Anschließend verlässt er das Gebäude und erscheint ggf. später zu einer anderen Vorlesung wieder. Diese Nutzergruppe simuliert damit wenige Ruhephasen von langer Dauer, welche sich daher gut spezifizieren lassen.
3. **Reinigungskraft:** Muss alle Räume im vorliegenden Gebäude am Vormittag reinigen. Daher kommt die Reinigungskraft zwischen 7:00 und 12:00 Uhr an, geht in jeden Raum und verweilt dort für eine kurze Zeitspanne von einer bis zu fünf Minuten. Diese Nutzergruppe simuliert damit sehr viele Ruhephasen von kurzer Dauer.
4. **Zufall:** Verhält sich unspezifisch. Eine Person dieser Gruppe kommt zwischen 7:00 und 18:00 Uhr im Gebäude an, geht während seiner zufällig langen Aufenthaltsdauer in mehrere Räume und verweilt dort für eine Zeitspanne zwischen einer und 120 Minuten. Diese Nutzergruppe simuliert damit ein unvorhersehbares Verhalten, was aufgrund des komplexen Bewegungsverhaltens von Menschen realistisch erscheint. Zugleich werden damit alle nicht betrachteten Nutzerrollen wie bspw. der Postbote oder die Besucher des vorliegenden Gebäudes repräsentiert.

Unter Verwendung der in Unterkapitel 3.2 vorgestellten Simulationsumgebung werden insgesamt 480 Personen (also 120 Nutzer pro Gruppe) erstellt, die

sich gemäß ihrem definierten Verhalten im vorliegenden Universitätsgebäude (s. Abbildung 3.7) bewegen. Die daraus resultierenden Beobachtungen werden zusammen mit der bekannten Grundwahrheit zur Evaluation verwendet.

Für den Sliding-Window-Ansatz wird eine Fenstergröße von  $t_w = 90$  Sekunden gewählt, da diese Zeitspanne lang genug ist, um im Durchschnitt eine WLAN-Beobachtung pro Zeitfenster zu machen und klein genug ist, um kurze Bewegungsänderungen erfassen zu können. Auf Basis der errechneten Distanzkurven wird im Zeitverlauf alle fünf Sekunden eine Klassifizierung des Bewegungszustands durchgeführt und mit der Grundwahrheit verglichen. Als Evaluationsmetriken werden dabei die Relevanz, die Sensitivität und das F-Maß (engl.: *Precision*, *Recall*, *F-score*) genutzt, welche typischerweise zur Leistungsbewertung eingesetzt werden [71]. Die drei Metriken beschreiben die Genauigkeit eines Klassifizierungsergebnisses und sind über das Verhältnis von *richtig positiv* ( $r_p$ ), *falsch positiv* ( $f_p$ ) bzw. *falsch negativ* ( $f_n$ ) wie folgt definiert:

$$\text{Relevanz} := \frac{r_p}{r_p + f_p} \quad \text{Sensitivität} := \frac{r_p}{r_p + f_n}$$

$$\text{F-Maß} := 2 \times \frac{\text{Relevanz} \times \text{Sensitivität}}{\text{Relevanz} + \text{Sensitivität}}$$

Das F-Maß stellt dementsprechend das gewichtete harmonische Mittel von Relevanz und Sensitivität dar. Da es im vorliegenden Fall auf gleiche Weise wichtig ist, beide Bewegungszustände korrekt zu klassifizieren, wird die Relevanz und Sensitivität jeweils als ungewichtetes Mittel über beide Zustände berechnet. Dies soll verhindern, dass die Bewertung eines dominanten Zustands das Klassifizierungsergebnis verfälscht, was vor allem dann der Fall ist, wenn das Verhältnis zwischen Ruhe- und Bewegungsphasen sehr unterschiedlich ausfällt.

### Klassifizierungsgenauigkeit bei deterministischer Entscheidung

Als erster Schritt soll die durchschnittliche Klassifizierungsgenauigkeit unter Berücksichtigung der beschriebenen Distanzfunktionen evaluiert werden, wenn eine deterministische Entscheidung getroffen wird. Hierzu wird die Verwendung eines statischen und eines dynamischen Schwellwerts getrennt voneinander untersucht und anhand der drei Metriken (Relevanz, Sensitivität und F-Maß) auf Basis des gesamten Datensatzes verglichen.

Abbildung 5.2 stellt die Ergebnisse der durchgeführten Versuche vor. Es zeigt sich, dass die Relevanz im Fall eines statischen Schwellwerts für die ersten drei Distanzfunktionen im Durchschnitt über 0,8 liegt und damit sehr hoch ist. Allerdings werden dabei gleichzeitig geringe Werte ( $< 0,58$ ) für die Sensitivität beobachtet, was die Folge einer hohen Anzahl an falsch negativen Entscheidungen ist. Der festgelegte Schwellwert führt also im Durchschnitt dazu, dass der tatsächliche Bewegungsstatus häufig nicht korrekt klassifiziert wird. Lediglich im Fall der Varianz hat sich der statische Schwellwert im Durchschnitt besser bewährt, was sich beim Vergleich aller erreichten Werte für das F-Maß ablesen

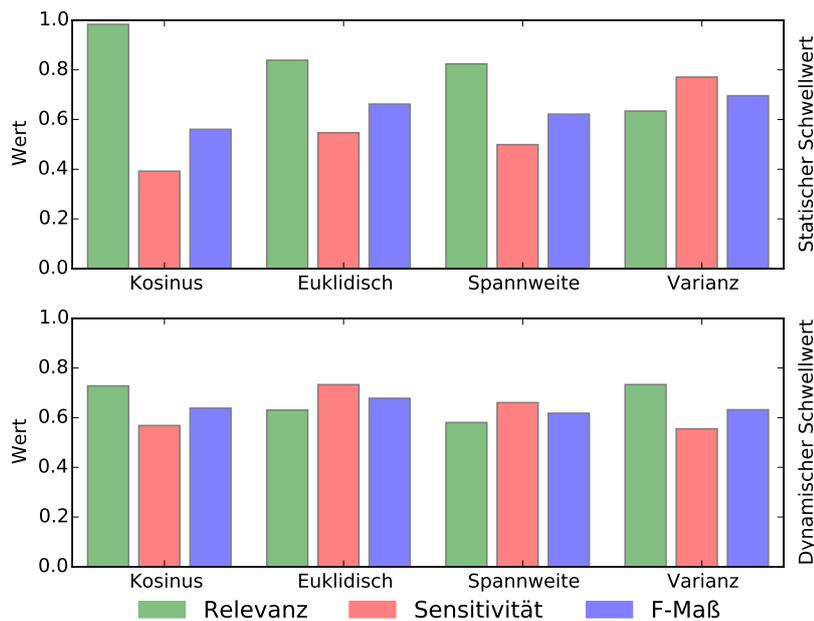


Abbildung 5.2: Evaluation der Klassifizierungsgenauigkeit bei Verwendung eines statischen bzw. dynamischen Schwellwerts und verschiedenen Distanzfunktionen.

lässt. Dadurch zeigt sich deutlich, dass eine deterministische Entscheidung auf Basis eines empirisch ermittelten Schwellwerts nicht generisch anwendbar ist und nur unter bestimmten Bedingungen ein verbessertes Klassifizierungsergebnis im Gesamtdurchschnitt (hier ein F-Maß von knapp 0,7) erreicht.

Im Unterschied dazu führt die Anwendung eines dynamischen Schwellwerts bei den untersuchten Distanzfunktionen zu ausgeglicheneren Ergebnissen für Relevanz und Sensitivität, wie im unteren Teil der Abbildung 5.2 zu sehen ist. Die Werte des F-Maßes sind ähnlich wie zuvor und liegen für alle Distanzfunktionen über 0,6. Der beste Wert wird hier für die Euklidische Distanz bei 0,68 beobachtet, was im Durchschnitt eine geringe Klassifizierungsgenauigkeit darstellt. Daher lässt sich zusammenfassend festhalten, dass eine deterministische Entscheidung unabhängig von der verwendeten Distanzfunktion keine verlässliche Schätzung über den Bewegungszustand eines Nutzers zurückliefert. Der Grund ist, dass die Verwendung eines Schwellwerts zu unflexibel ist, um auf die großen Unterschiede im komplexen Bewegungsverhalten verschiedener Nutzer adäquat eingehen zu können. Ein probabilistischer Ansatz kann mit solchen Unsicherheiten besser umgehen und höhere Genauigkeiten erreichen, da dieser eine wahrscheinlichkeitstheoretische Entscheidung über den Bewegungsstatus eines Nutzers trifft.

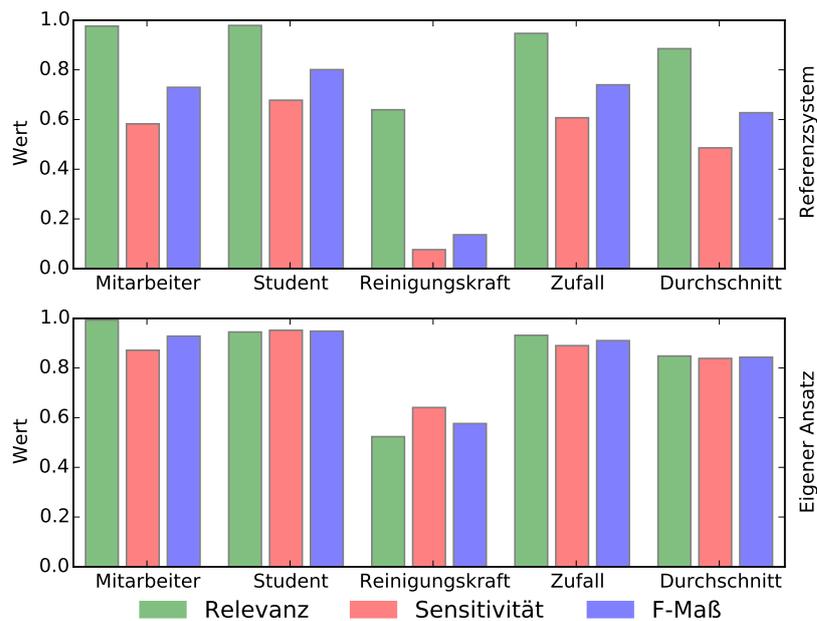


Abbildung 5.3: Evaluation der Klassifizierungsgenauigkeit bei Verwendung von LOCADIO [116] als Referenzsystem und dem eigenen Ansatz unter Berücksichtigung der erzeugten Nutzergruppen.

### Klassifizierungsgenauigkeit bei probabilistischer Entscheidung

Zur Evaluation des probabilistischen Ansatzes wird die Euklidische Distanz verwendet, da diese beim vorherigen Versuch das beste Klassifizierungsergebnis (knapp vor der Varianz) im Gesamtdurchschnitt erreicht hat. Anders als zuvor, werden nicht die verschiedenen Distanzfunktionen, sondern die vier unterschiedlichen Nutzergruppen separat untersucht. Für einen besseren Vergleich mit bereits existierenden Verfahren wird der gleiche Versuch zusätzlich mit einem Referenzsystem durchgeführt. Hierzu wurde LOCADIO von Krumm und Horvitz [116] nachimplementiert und auf die vorhandenen Daten angewendet. Es sei nochmals darauf hingewiesen, dass sich der eigene Ansatz in Bezug auf die verwendete Distanzfunktion und dem eingesetzten HMM in den Parametern  $O, A, B$  und  $\pi$  von LOCADIO unterscheidet.

Abbildung 5.3 stellt die erreichten Klassifizierungsergebnisse für beide Systeme auf Basis der drei Metriken Relevanz, Sensitivität und F-Maß dar. Neben den einzelnen Nutzerrollen wird zusätzlich die durchschnittliche Genauigkeit angegeben, welche mit den Werten für die Euklidische Distanz aus dem vorherigen Versuch in Abbildung 5.2 direkt verglichen werden kann. Hierbei zeigt sich, dass der eigene Ansatz in Hinblick auf Relevanz und Sensitivität genauer und ausgeglichener ist. Mit Ausnahme der Personen, die als Reinigungskraft modelliert sind, wird ein deutlich höheres F-Maß von über 0,91 sogar für die Zufallsgruppe erreicht. Der durchschnittliche Wert beträgt immerhin 0,84 und repräsentiert damit ein adäquates Ergebnis für den generischen Fall. Dies stellt

eine Verbesserung zum vorherigen Versuch von mehr als 0,14 im F-Maß dar.

Die Gruppe der Reinigungskräfte führt bei beiden Verfahren zu einer geringeren Klassifizierungsgenauigkeit. Dies war allerdings zu erwarten, da hier eine hohe Frequenz an Zustandsänderungen simuliert wird, die schwer zu erfassen ist. Offensichtlich kann der eigene Ansatz damit besser umgehen (F-Maß=0,57), als das Referenzsystem (F-Maß=0,13), bei welchem ein stärkerer Rückgang der relevanten Werte zu verzeichnen ist. Dies ist vor allem auf die erwähnten Unterschiede in der Matrix  $A$  und somit auf die gewählten Übergangswahrscheinlichkeiten zurückzuführen, die den eigenen Ansatz flexibler und damit robuster für unterschiedliche Verhaltensmuster machen. Für die anderen Gruppen erreicht das Referenzsystem ein F-Maß von über 0,73 und im Durchschnitt 0,63 – wegen des schlechten Abschneidens im Fall der Reinigungskräfte. Damit ist es im Gesamtdurchschnitt sogar etwas ungenauer als die deterministische Entscheidung mit dynamischem Schwellwert und Euklidischer Distanz (vgl. Abbildung 5.2). Allerdings wird für die Gruppe der Studenten wiederum ein hohes F-Maß von 0,8 erreicht, was den dargestellten Ergebnissen von Krumm und Horvitz nahe kommt.

Insgesamt gilt es jedoch festzuhalten, dass die Nachimplementierung des Referenzsystems keine verlässliche Klassifizierungsgenauigkeit für den generischen Fall liefert. Im Vergleich dazu wird durch den eigenen Ansatz eine durchschnittliche Verbesserung von 0,21 im F-Maß erreicht. Es wird vermutet, dass diese Verbesserung hauptsächlich durch die Verwendung aggregierter WLAN-Beobachtungen über mehrere Monitore erzielt wird. LOCADIO berücksichtigt hier lediglich die Varianz des stärksten empfangenen Signals. Damit zeigt der eigene Ansatz insgesamt die besten Klassifizierungsergebnisse, weshalb er im nächsten Schritt zur Bereitstellung der Aktivitätsinformation bei der Pfadermittlung zum Einsatz kommen soll.

### **Integration der Aktivitätsinformation bei der Pfadermittlung**

Es wird nun überprüft, ob sich mithilfe des klassifizierten Bewegungszustands die probabilistische Pfadermittlung, welche in Abschnitt 4.2.4 evaluiert wurde, verbessern lässt. Hierzu wird das Bewegungsmodell des State-Partikel-Filters leicht verändert und mit der extrahierten Aktivitätsinformation adaptiv unterstützt. Ein möglicher Nachteil des Filter-Verfahrens ergibt sich aus dem eingeschränkten Lösungsraum, der durch den zugrunde liegenden Zustandsgraphen definiert wird. Dadurch kann der Positionsfehler insbesondere bei längeren Ruhephasen des Nutzers durch falsche Positionsschätzungen stetig anwachsen, was zu einer verringerten Genauigkeit in der Pfadermittlung führt. Durch die Integration der Aktivitätsinformation in das Bewegungsmodell soll dieses Problem des State-Partikel-Filters minimiert werden.

Dabei ist die Idee wie folgt: Solange der Nutzer als *in Bewegung* klassifiziert wird, bleibt der Filter unverändert und verhält sich genau wie zuvor. Wenn sich der Nutzer aber laut der durchgeführten Aktivitätserkennung in einer Ruhephase befindet, wird automatisch das Freiflächen-Bewegungsmodell von

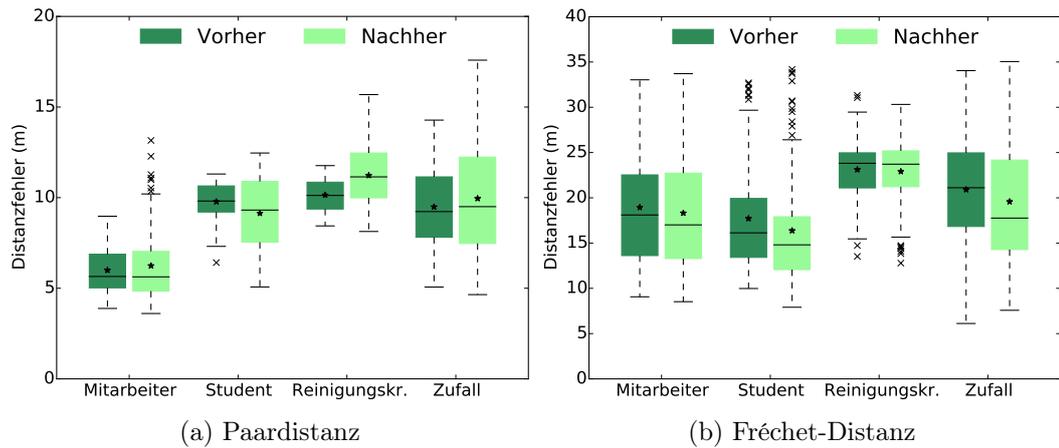


Abbildung 5.4: Verteilung der Lokalisierungsfehler vor und nach der Modifikation des Bewegungsmodells.

Widyawan et al. [221] verwendet und die angenommene Geschwindigkeit der Partikel auf 0 bis 1 m/h herabgesetzt. Durch diese Veränderungen wird verhindert, dass die ermittelte Position des Nutzers während einer klassifizierten Ruhephase durch ungenaue Positionsschätzungen stark verfälscht wird. Trotzdem können noch kleinere Positionskorrekturen im freien Raum um den letzten ermittelten Standort durchgeführt werden. Es sei angemerkt, dass hierfür akkurate Informationen über den Bewegungszustand für jeden Vorhersage-Schritt des Filters vorliegen müssen.

Ob ein solches Verfahren eine Verbesserung der Positionierungsgenauigkeit bewirken kann, wird auf Basis der vier Nutzergruppen und des vorgestellten Ansatzes zur Bestimmung des Bewegungsstatus evaluiert. Hierzu wird wie in Abschnitt 4.2.4 die durchschnittliche Paar- bzw. die diskrete Fréchet-Distanz betrachtet und der entstandene Lokalisierungsfehler analysiert. Dies geschieht vor und nach der Modifikation des Bewegungsmodells separat für jede der vier Nutzergruppen. In Abbildung 5.4 werden die entsprechenden Ergebnisse als Boxplot-Diagramm zur Veranschaulichung der Fehlerverteilungen dargestellt. Dabei fällt unabhängig der betrachteten Distanz auf, dass im Fall der Reinigungskraft keine Verbesserung erzielt wurde, während für die Gruppe der Studenten jeweils ein geringerer durchschnittlicher Fehler nach der Modifikation gemessen werden konnte. Vergleicht man dies mit dem Ergebnis für den oben evaluierten Ansatz zur Aktivitätserkennung (s. Abbildung 5.3), so zeigt sich ein Zusammenhang zwischen der erreichten Klassifizierungsgenauigkeit und der beobachteten Verbesserung bei der Pfadermittlung. Demnach gilt, dass die Verwendung zusätzlicher Aktivitätsinformationen das zugrunde liegende Bewegungsmodell des State-Partikel-Filters verbessern kann, aber nur dann, wenn diese Information verlässlich ist. Im vorliegenden Versuch ist dies bei einem F-Maß von ca. 0,92 der Fall.

Zudem fallen Unterschiede bzgl. der betrachteten Distanz auf. Während bei

der durchschnittlichen Paardistanz (vgl. Abbildung 5.4a) nach der Modifikation Verteilungen mit gleichem und höherem Fehler entstehen, werden für die Fréchet-Distanz (vgl. Abbildung 5.4b) geringere Fehler und somit eine Verbesserung in der Pfadermittlung beobachtet. Diese liegt zwischen 1 bis 1,5 m für den Durchschnitt der betrachteten Pfade und fällt damit eher gering aus. Das wiederum spricht dafür, dass das ursprüngliche Bewegungsmodell bereits die menschliche Bewegung adäquat und generisch abbildet. Trotzdem lässt sich festhalten, dass die Reduzierung der Partikelgeschwindigkeit und die Korrektur der Positionsschätzung während einer erkannten Ruhephase nachweislich einen positiven Effekt auf den Gesamtfehler der Pfadermittlung hat. Allerdings zeigen die erhöhten Fehler in der Paardistanz, dass sich viele einzelne Positionsupdates im Vergleich zur Grundwahrheit verschlechtern. Dies ist vor allem dann der Fall, wenn der erkannte Bewegungszustand falsch ist, bzw. wenn zwischen zwei Bewegungszuständen gewechselt wird. Dennoch ist insgesamt eine Integration der Aktivitätsinformation zu empfehlen, da der Gesamtfehler bei der Pfadermittlung reduziert werden konnte.

#### 5.1.4 Zusammenfassung und Diskussion

Die eben gewonnenen Erkenntnisse legen nahe, dass eine zuverlässige Erkennung von Ruhe- und Bewegungsphasen rein auf Basis passiver WLAN-Beobachtungen möglich ist. Hieraus lassen sich entsprechende Aktivitätsinformationen gewinnen, die aus bloßen Ortsschätzungen nicht in dieser Genauigkeit extrahiert werden können. Es wurde gezeigt, dass deterministische Verfahren über einen zu bestimmenden Schwellwert nicht die nötige Flexibilität besitzen, um das komplexe Bewegungsverhalten von Personen auf verlässliche Weise abzubilden. So führte in der Evaluation weder ein statischer noch ein dynamischer Schwellwert zu einer adäquaten Klassifizierungsgenauigkeit. Der höchste Wert, der dabei auf Basis simulierter Nutzer im Gesamtdurchschnitt erreicht wurde, lag bei einem F-Maß von knapp 0,7. Dies legt einen relativ hohen Anteil an falschen Klassifizierungen nahe.

Im Gegensatz dazu zeigte der hier entwickelte Ansatz, der eine probabilistische Entscheidung auf Basis eines eigenen HMMs trifft, die eindeutig besseren Ergebnisse. Das implementierte Referenzsystem schnitt allerdings im Gesamtdurchschnitt etwas schlechter ab als das deterministische Verfahren mit dynamischem Schwellwert. Der Grund hierfür wird vor allem darin gesehen, dass das Referenzsystem nur die jeweils stärkste empfangbare Signalstärke als Beobachtung nutzt und nicht wie im vorliegenden Fall einen Beobachtungsvektor aus mehreren RSS-Werten. Der eigene Ansatz geht zudem von anderen Übergangswahrscheinlichkeiten in der Matrix  $A$  aus, so dass ein Wechsel des Bewegungszustands leichter möglich wird. So werden verschiedene Verhaltensmuster generisch abgebildet, auch wenn dadurch der Fehler für eine spezifizierte Gruppe steigt. Außerdem wurde die Anfangswahrscheinlichkeit und die Beobachtungswahrscheinlichkeit im Vergleich zum Referenzsystem geändert, um die

Realitäten im vorliegenden Szenario exakter zu modellieren. Bspw. wird dabei die Annahme umgesetzt, dass ein Nutzer sich am Anfang in den beobachteten Bereich hinein bewegt.

Insgesamt stellt der eigene Ansatz eine Verbesserung von 0,21 im F-Maß gegenüber dem Referenzsystem und von 0,16 gegenüber dem deterministischen Verfahren mit dynamischem Schwellwert dar. Dazu trägt die Tatsache bei, dass selbst die schwer zu klassifizierenden Bewegungszustände in der Gruppe der Reinigungskräfte häufig (mit einem F-Maß von 0,57) korrekt zugeordnet werden konnten. Zudem stellen die Werte bzgl. Relevanz und Sensitivität, die bei den anderen Nutzergruppen erreicht wurden, ein robustes und akkurates Klassifizierungsergebnis dar. Dieses ist verlässlich genug, um den Lokalisierungsfehler, der bei der Pfadermittlung entstand, mithilfe der extrahierten Aktivitätsinformation zu verringern.

Allerdings gilt es zu beachten, dass die hier gewonnenen Ergebnisse rein auf Simulationen beruhen, welche bestimmte Einstellungen erfordern und daher gewissen Einschränkungen unterliegen können. Obwohl versucht wurde, ein möglichst realistisches und komplexes Bewegungsverhalten in einer realen Umgebung mit Vorwissen über die Domäne zu realisieren, kann trotzdem eine systematische Verzerrung in den simulierten Daten vorliegen. Um einen solchen Fehler ausschließen zu können, müssen die hier evaluierten Ansätze in einer zukünftigen Studie unter realen Bedingungen nochmals untersucht und validiert werden. Hierfür ist allerdings wieder die genaue Grundwahrheit erforderlich, die oftmals schwer zu erfassen ist.

## 5.2 Klassifizierung von Nutzerrollen

Wie zu Beginn dieses Kapitels erwähnt, kann nach Fearon [59] die Identität einer Person ermittelt werden, indem diese Person einer bestimmten Gruppe zugeordnet wird. Daher wird hier die Klassifizierung von Nutzerrollen als Extraktion der Identitätsinformation aufgefasst. Eine Nutzerrolle ist die Funktion, die eine Person beim Betreten des Gebäudes inne hat. Mit dieser sind meistens auch die Aktivität und das Ziel verknüpft, welches in dem Gebäude verfolgt wird. Bspw. betritt ein Angestellter das Büro zum Arbeiten oder der Student das Universitätsgebäude für die Vorlesung. Aufgrund dieser Überlegungen liegt die Idee nahe, dass sich mithilfe der bereits extrahierten Kontextinformationen wie Zeit, Ort und Aktivität nun auch die Identität als letzte fehlende Schlüsselinformation gewinnen lässt. Insgesamt wird dadurch eine vollständige Extraktion von primären Kontextinformationen auf Basis aufgezeichneter WLAN-Beobachtungen erreicht.

Im folgenden Unterkapitel gilt es herauszufinden, welcher Informationstyp (bzw. eine Kombination daraus) die essentiellen Eigenschaften für die Bestimmung der korrekten Nutzerrolle beinhaltet und welcher gebräuchliche Klassifikator aus dem Bereich des maschinellen Lernens sich dazu besonders eignet. Als Basis des Klassifizierungsschritts dienen statistische Features, welche zu-

vor aus den gesammelten Daten und den bereits extrahierten Kontextinformationen für jeden Nutzer errechnet werden. Die Evaluation findet auf den simulierten Nutzergruppen aus Abschnitt 5.1.3 statt, wobei die Ergebnisse im Anschluss anhand realer Beobachtungen diskutiert werden. Da das prinzipielle Vorgehen zum Teil gängige Praxis bei Klassifizierungsproblemen darstellt, lohnt zunächst ein kurzer Blick auf verwandte Arbeiten, um die wissenschaftliche Relevanz und die Neuwertigkeit des hier vorgestellten Konzepts zu erörtern.

### 5.2.1 Verwandte Arbeiten

Eine umfassende Arbeit zur Extraktion und Analyse von Informationen aus WLAN-Beobachtungen stellt [171] von Ruiz-Ruiz et al. dar. Darin versuchen die Autoren zunächst verschiedene spatio-temporale Features zu bestimmen und anschließend vorhandene Nutzer auf Basis festgelegter Regeln in definierte Nutzergruppen einzuteilen. Zur Klassifizierung wird neben einem Bayesschen Netz aus dem Bereich des maschinellen Lernens ein eigener Rollen-basierter Klassifikator eingesetzt. Die Evaluation erfolgt unter realen Bedingungen in einer Krankenhausumgebung, wobei die Ergebnisse wegen der fehlenden Grundwahrheit anhand von Statistiken überprüft werden. Im Gegensatz zur vorliegenden Arbeit werden zur Klassifizierung von Nutzerrollen keine statistischen Features aus extrahierten Kontextinformationen herangezogen. Außerdem werden andere Feature-Gruppen und Klassifikatoren eingesetzt.

Insgesamt existieren nur wenige Publikationen, die sich mit der Klassifizierung von Nutzerrollen aus aufgezeichneten WLAN-Daten beschäftigen. Muthukrishnan et al. [147] bspw. diskutieren lediglich am Ende die Möglichkeit, durch kombinierte Feature-Gruppen weiterführende Aktivitäten wie *Fahrrad*- oder *Autofahren* zu bestimmen. Millonig et al. [139] geben einen generellen Überblick über Methoden zur Bestimmung von spatio-temporalem Verhalten von Fußgängern. Neben der Auflistung von technischen Möglichkeiten zur Datengewinnung werden experimentelle Versuche beschrieben. Dabei werden u.a. Personen in einem Einkaufszentrum anhand ihrer Gehgeschwindigkeit, den besuchten Orten und ihrem Kleidungsstil in drei Rollen eingeteilt.

Häufiger werden Personen in der Literatur anhand von visuellen Daten klassifiziert, die bspw. mithilfe von Überwachungskameras innerhalb von Gebäuden aufgenommen wurden. Munder und Gavrilla [144] präsentieren dazu eine umfangreiche experimentelle Studie zur Klassifikation von Fußgängern. Dabei definieren sie ebenfalls verschiedene Feature-Gruppen wie globale, lokale, adaptive und nicht-adaptive und untersuchen gebräuchliche Klassifikatoren wie die Support-Vector-Machine, neuronale Netze oder den k-Nächsten-Nachbarn-Klassifikator.

Wie sich anhand der aufgeführten Literatur zeigt, ist das prinzipielle Vorgehen bei der Klassifizierung häufig sehr ähnlich: Zunächst werden Features aus den vorhandenen, meist bereinigten Daten extrahiert und teilweise selektiert. Die relevanten Features werden anschließend einzeln oder in Kombination als

Eingangsparameter für den gewählten Klassifikator verwendet. Auch in der vorliegenden Arbeit werden diese Schritte durchlaufen. Allerdings ist sowohl die Extraktion als auch die Kombination der hier verwendeten Features als neuwertig zu betrachten, da nach heutigem Kenntnisstand keine Arbeit existiert, die ein vergleichbares Konzept zur Klassifizierung von Nutzerrollen präsentiert. Dieses wird im Folgenden ausführlich vorgestellt.

### 5.2.2 Konzept zur Klassifizierung von Nutzerrollen

Das hier vorgeschlagene Konzept zur Klassifizierung von Nutzerrollen basiert auf den bisher vorgestellten Methoden zur Datenerfassung und den entwickelten Ansätzen zur Extraktion von Orts- und Aktivitätsinformationen. Zur besseren Veranschaulichung stellt Abbildung 5.5 einen schematischen Überblick über das Gesamtkonzept dar. Dieses umfasst die folgenden vier Schritte:

1. **Datenerfassung:** Messstationen zeichnen IEEE 802.11 Probe-Requests, die von mobilen Geräten in Empfangsreichweite ausgesendet werden, in einem Bereich auf und aggregieren diese zu Beobachtungsvektoren.
2. **Informationsgewinnung:** Auf Basis der zuvor entwickelten Verfahren (vgl. Unterkapitel 4.2 und 5.1) werden Orts- und Aktivitätsinformationen aus den aufgezeichneten WLAN-Daten extrahiert. Zusätzlich werden direkt aus den Beobachtungsvektoren sogenannte Basis-Informationen wie Zeitstempel und Empfangssignalstärken ausgelesen.
3. **Feature-Berechnung:** Über die extrahierten Basis-, Aktivitäts- und Ortsinformationen werden statistische Features berechnet und in entsprechenden Feature-Gruppen (FG) zusammengefasst.
4. **Klassifizierung:** Auf Grundlage der vorliegenden Feature-Gruppen werden im letzten Schritt die Nutzerrollen über gängige Methoden des maschinellen Lernens klassifiziert. Hierzu werden die berechneten Features teilweise selektiert und anschließend skaliert, um Effekte wie *Overfitting* zu vermeiden.

Im Folgenden werden die neu eingeführten Aspekte im Detail behandelt. Diese sind in der Abbildung 5.5 farbig illustriert und umfassen dementsprechend die beiden letzten Schritte des Gesamtkonzepts, nämlich die Feature-Berechnung und die anschließende Klassifizierung.

#### Feature-Berechnung

In diesem Schritt liegen bereits die Basis-, Aktivitäts- und Ortsinformationen vor, welche zuvor aus den Daten extrahiert wurden. Sie repräsentieren drei der vier primären Typen von Kontextinformationen eines Nutzers. Für jede der drei Kategorien werden nun statistische Features bestimmt, die später zur

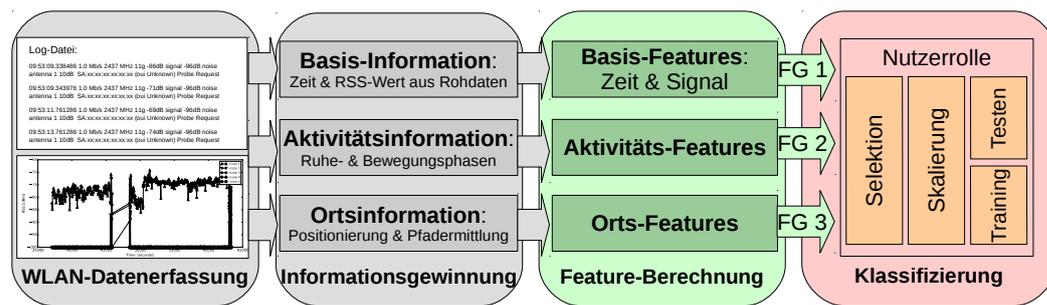


Abbildung 5.5: Schematische Darstellung des Gesamtkonzepts zur Klassifizierung von Nutzerrollen. Die bereits vorgestellten Methoden sind grau und die neuen Aspekte farbig markiert.

Klassifizierung von Nutzerrollen und damit zur Extraktion der Identitätsinformation dienen. Ziel dabei ist es, dass die berechneten Features die wesentlichen Eigenschaften einer MAC-spezifischen Zeitserie innerhalb einer Kategorie mithilfe numerischer Werte möglichst exakt beschreiben. Zu diesem Zweck werden in Anlehnung an [132] die folgenden statistischen Maße zur Feature-Berechnung herangezogen: Maximum und Minimum sowie der Mittelwert, der Median, die Standardabweichung, die Entropie, die mittlere quadratische Abweichung und das 75-Prozent-Quantil. Diese Maße werden in jeder der drei Kategorien für die vorliegende Information berechnet. Zusätzliche können noch andere gruppenspezifische Features definiert werden. Dadurch ergibt sich die Zusammenstellung der folgenden drei Feature-Gruppen:

- **Basis-Features (FG 1):** In dieser Gruppe werden die statistischen Maße für die Menge an RSS-Werten berechnet, welche sich aus der Zeitserie von Beobachtungsvektoren pro MAC-Adresse ablesen lassen. Als zusätzliche Features werden die Zeitpunkte der ersten und letzten Beobachtung sowie der Beobachtungszeitraum, die durchschnittliche Beobachtungsfrequenz und der Anteil an fehlenden Beobachtungen innerhalb des Beobachtungszeitraums betrachtet. Somit wird die Gruppe durch 49 Features repräsentiert.
- **Aktivitäts-Features (FG 2):** Aus der extrahierten Aktivitätsinformation für jeden Nutzer werden zunächst die Zeitspannen der einzelnen Ruhe- und Bewegungsphasen separat ermittelt. Daraufhin werden die statistischen Maße auf diese Zeitangaben angewendet. Als zusätzliche Features wird die akkumulierte Anzahl der einzelnen Bewegungszustände betrachtet, wodurch sich insgesamt 18 Features für diese Gruppe bestimmen lassen.
- **Orts-Features (FG 3):** Aus der extrahierten Ortsinformation für jeden Nutzer werden die Distanzen zwischen zwei aufeinanderfolgende Positionsschätzungen errechnet und die statistischen Maße auf diese Distanzwerte angewendet. Als zusätzliche Features werden die Anzahl an Positionsschätzungen sowie die Gesamtlänge der zurückgelegten Trajektorie aus der Pfad-

ermittlung herangezogen. Dadurch wird diese Gruppe durch insgesamt zehn Features repräsentiert.

Mithilfe der vorgestellten Feature-Gruppen ist es möglich, die vorhandenen Kategorien an Informationen zu beschreiben. Diese lassen sich entweder einzeln oder in Kombination für den Klassifizierungsschritt verwenden. Dadurch soll untersucht werden, welche Kategorie entscheidend ist, um die beobachteten Nutzer bzgl. ihrer Rolle korrekt klassifizieren zu können. Insgesamt stehen hierfür, bei einer Kombination aller drei Gruppen, 77 Features zur Verfügung.

## Klassifizierung

Wie in Abbildung 5.5 dargestellt, besteht der letzte Schritt des Gesamtkonzepts aus den vier Bereichen: Selektion und Skalierung der Feature-Werte sowie dem Trainieren und Testen des gewählten Klassifikators. Das Training erfolgt auf Basis sogenannter Trainingsdaten, die den vorhandenen Klassen bereits korrekt zugeordnet wurden. Mit deren Hilfe kann der Klassifikator eine Funktion erlernen, welche den gewünschten Zusammenhang zwischen Ein- und Ausgabe abbildet. Beim Testen, auch Validierung genannt, wird das trainierte Modell anhand von Test- bzw. Validierungsdaten überprüft. Dabei sollten Trainings- und Testdaten zwei disjunkte Mengen darstellen, um die Generalisierbarkeit des Modells evaluieren zu können [114].

Ziel der Klassifizierung ist es, die vorhandenen Feature-Vektoren der bisher unbekanntem Nutzer (*Samples*) möglichst akkurat der jeweils korrekten Nutzerrolle (*Label*) zuzuordnen. Um generalisierbare und damit brauchbare Ergebnisse zu erhalten, sollten Effekte wie Überanpassung (engl.: *Overfitting*) vermieden werden. Diese entstehen vor allem dann, wenn das Modell durch die verwendeten Features zu spezifisch auf die Trainingsdaten hin trainiert wird. Daher ist es notwendig, zunächst eine geeignete Selektion aus der Menge der berechneten Features vorzunehmen. Zu diesem Zweck werden im Rahmen der vorliegenden Arbeit die folgenden Methoden untersucht:

- **Varianz-Schwellwert:** Es werden diejenigen Features aus der Gesamtmenge entfernt, die eine geringere Varianz besitzen als ein festgelegter Schwellwert vorgibt. Im vorliegenden Fall wird dazu ein gängiger Wert von 1,0 verwendet.
- **$k$ -Best-Selektion:** Es werden nur die  $k$  besten Features anhand einer Bewertungsfunktion für die Klassifizierung herangezogen. Eine solche Funktion kann bspw. die stochastische Unabhängigkeit der Features prüfen, wodurch sich die  $k$  eindeutigsten Merkmale auswählen lassen. Hierfür eignen sich bspw. T-Tests oder der Chi-Quadrat-Test, wobei Letzterer keine Normalverteilung voraussetzt und daher im vorliegenden Fall zur Anwendung kommt.
- **Hauptkomponentenanalyse:** Durch die Hauptkomponentenanalyse (PCA für engl.: *Principal Component Analysis*) lässt sich ein multivariater Daten-

satz auf eine Teilmenge orthogonaler Komponenten reduzieren, welche zueinander linear unabhängig sind und somit die größtmögliche Varianz aufweisen. Damit wird wieder das Ziel verfolgt, die eindeutigsten  $k$  Merkmale zu identifizieren, um diese für die Klassifizierung verwenden zu können. Analog zur vorherigen Methode wird demnach ein Wert für  $k$  benötigt, der angibt, wie viele Features ausgewählt werden sollen.

Im Anschluss an die Selektion folgt die Skalierung der ausgewählten Features. Dabei wird eine Standardisierung des selektierten Datensatzes vorgenommen, um diesen an eine Standardnormalverteilung mit Mittelwert null und Varianz eins anzunähern. Das Vorgehen ist nötig, da viele Klassifikatoren im Bereich des maschinellen Lernens üblicherweise eine solche Verteilung der Werte erwarten, um korrekt arbeiten zu können [163]. Im vorliegenden Fall wird die Standardisierung mittels einer Verschiebung hin zum Mittelwert null und einer anschließenden Skalierung der Werte auf die Varianz eins erreicht.

Bevor die eigentliche Klassifizierung durchgeführt werden kann, muss ein geeigneter Klassifikator gewählt werden. Hierzu existieren diverse Lernalgorithmen, die sich in der Literatur bewährt haben [109]. Die Entscheidung für einen bestimmten Algorithmus ist nicht trivial und kann von verschiedenen Faktoren wie bspw. der Datenverteilung oder der vorhandenen Feature-Anzahl abhängen. Im vorliegenden Fall steht nur eine geringe Anzahl von maximal 77 Features zur Verfügung, die nicht zwangsläufig statistisch unabhängig sind. Außerdem kann die Menge an Samples für eine Vielzahl von Nutzern beliebig ansteigen. Aus diesen Überlegungen heraus scheinen die folgenden gängigen Klassifikatoren aus dem Bereich des maschinellen Lernens hier am geeignetsten zu sein [163]:

- **Entscheidungsbaum (DT<sup>2</sup>):** Stellt eine parameterfreie Methode des überwachten Lernens zum Lösen von Entscheidungs- bzw. Klassifizierungsproblemen dar. Er ist einfach zu interpretieren und kommt auch mit nicht normalisierten Daten gut zurecht. Allerdings neigen Entscheidungsbäume schnell zu Overfitting, weshalb eine geringe Feature-Anzahl sowie eine begrenzte Tiefe und eine Mindestanzahl an Samples pro Blatt von Vorteil sind, um dieses Problem zu vermeiden. Entscheidungsbäume gelten zudem häufig als instabil, da bereits eine kleine Veränderung der Daten einen komplett anderen Baum erzeugen kann. Durch die Verwendung mehrerer Bauminstanzen wie bspw. bei Random-Forest lässt sich hier eine Verbesserung erzielen.
- **Random-Forest:** Zur Reduzierung von Overfitting vereint der Random-Forest eine beliebige Anzahl von unkorrelierten Entscheidungsbäumen, welche auf Basis verschiedener Teilmengen des vorhandenen Trainingsdatensatzes trainiert werden. Die Entscheidungen werden schließlich gemittelt und es wird diejenige Klasse zurückgeliefert, die am häufigsten gewählt wurde. Die Teilmengen werden im vorliegenden Fall durch Ziehen mit Zurücklegen

---

<sup>2</sup>Abkürzung für engl.: *Decision Tree*

aus dem Datensatz erzeugt. Der Vorteil von Random-Forest liegt neben der Kontrolle von Overfitting bei der Parallelisierbarkeit des Testverfahrens, wodurch sich auch größere Datensätze effizient und schnell klassifizieren lassen. Im vorliegenden Fall wird der Random-Forest mit 100 Entscheidungsbäumen getestet, was für den verwendeten Datensatz als ausreichend erscheint.

- **Support-Vector-Machine (SVM)**: Kann zur Regression und Klassifizierung verwendet werden. Dabei wird anhand von bekannten Trainingsdaten versucht, eine passende Hyperebene in den Vektorraum der vorhandenen Feature-Vektoren einzufügen, um diese gemäß den Klassen maximal zu separieren. Der Vorteil von SVMs ist demnach, dass sie selbst mit hochdimensionalen Daten effizient umgehen können. Allerdings kann sich das Ergebnis verschlechtern, wenn die Feature-Anzahl die Menge an gegebenen Samples übersteigt. Dies ist aber im vorliegenden Szenario nicht der Fall.
- **Mehrlagiges Perzeptron (MLP)**: Zählt zur Kategorie der künstlichen neuronalen Netze. Demnach verwendet ein MLP die vorhandenen Features als primäre Eingabewerte und lernt auf Basis von Trainingsdaten über mehrere nicht-lineare Ebenen eine Funktion, welche wiederum zur Klassifizierung oder zur Regression genutzt werden kann. Somit besitzt es die Fähigkeit, nicht-lineare Modelle in Echtzeit zu erlernen. Allerdings können die gewählten Anfangsgewichte zu Unterschieden in der Klassifizierungsgenauigkeit führen. Zudem ist eine Skalierung der ausgewählten Features unbedingt erforderlich, um brauchbare Ergebnisse zu erhalten. MLPs können sehr komplex und gerade für große Datenmengen extrem zeitaufwendig werden.
- **Voting**: Kombiniert eine beliebige Anzahl an verschiedenen Klassifikatoren und vereint die jeweiligen Ergebnisse durch eine Mehrheitsentscheidung. Dabei kann entweder diejenige Klasse zurückgeliefert werden, die am häufigsten gewählt wurde (harte Entscheidung), oder welche die durchschnittlich höchste Wahrscheinlichkeit besitzt (weiche Entscheidung). Im vorliegenden Fall wird eine harte Entscheidung auf Basis der genannten Klassifikatoren umgesetzt. Der Vorteil von Voting besteht darin, die Schwächen der einzelnen trainierten Modelle bei ähnlicher Funktionsfähigkeit auszugleichen.

Zur Beurteilung des vorgestellten Gesamtkonzepts sollen die eben angeführten Lernalgorithmen für das vorliegende Klassifizierungsproblem unter Verwendung der drei Feature-Gruppen evaluiert werden. Eine der gängigsten Metriken stellt hierbei die Vorhersagegenauigkeit des trainierten Klassifikators dar, die durch das Verhältnis von korrekt zugewiesenen Klassen zur Gesamtzahl der Klassifizierungsanfragen definiert ist. Zur Bestimmung der Genauigkeit muss also die korrekte Klasse für jede Anfrage bereits bekannt sein. Deshalb ist es gängige Praxis, dass die Menge der Trainingsdaten, deren Klassenzuordnung feststeht, auf eine bestimmte Weise geteilt wird, so dass der eine Teil zum Trainieren und der andere zum Testen des Klassifikators verwendet wird. Hierzu

bieten sich nach Kotsiantis et al. [109] drei prinzipielle Verfahren an: Beim ersten werden zwei Drittel der Menge zum Trainieren und das verbleibende Drittel zum Testen verwendet. Eine andere beliebte Möglichkeit ist das *Kreuzvalidierungsverfahren* und ein Spezialfall davon die *Leave-One-Out-Kreuzvalidierung*. Bei beiden Verfahren werden die Trainingsdaten in  $n$  gleichgroße und unabhängige Teilmengen zerlegt, wobei die Leave-One-Out-Kreuzvalidierung jedes einzelne Datenelement als eine Teilmenge auffasst. Anschließend wird für jede dieser Teilmengen der Klassifikator getestet, welcher zuvor auf den  $n-1$  übrigen Mengen trainiert wurde. Das Gesamtergebnis ergibt sich dann aus dem arithmetischen Mittel der Fehlerraten über alle getesteten Teilmengen. Dadurch soll Overfitting auch bei kleineren Datensätzen minimiert werden. In einigen Arbeiten wie bspw. [219] wird zur Evaluation des Klassifikators eine zehnfache stratifizierte Kreuzvalidierung verwendet. Diese achtet bei der Aufteilung der Gesamtmenge darauf, dass die Daten in den zehn Teilmengen eine möglichst gleiche Verteilung aufweisen. Dadurch lässt sich die Varianz über den Ergebnissen der einzelnen Teilmengen gering halten. Ein solches Verfahren kommt deshalb in der nachfolgenden Evaluation zum Einsatz.

### 5.2.3 Evaluation und Klassifizierungsergebnisse

Als Datensatz zur Informationsgewinnung und der anschließenden Feature-Berechnung dienen nochmals die simulierten WLAN-Beobachtungen der vier Nutzergruppen, die bereits zur Evaluation der Aktivitätserkennung verwendet wurden (vgl. Abschnitt 5.1.3). Ziel ist es, auf Basis der errechneten Feature-Gruppen und des gewählten Klassifikators jedem der 480 erzeugten Nutzer die korrekte Rolle (Mitarbeiter, Student, Reinigungskraft, Zufall) zuweisen zu können. Dabei soll evaluiert werden, welche Kombination aus Feature-Gruppen und Klassifikator die höchste Vorhersagegenauigkeit erreicht, was mittels der zehnfachen stratifizierten Kreuzvalidierung untersucht wird.

Zunächst werden hierzu alle errechneten Features ohne vorangegangene Selektion betrachtet, um ein erstes Verständnis für den Datensatz und die einzelnen Feature-Gruppen zu erhalten. Diese werden skaliert und anschließend zum Trainieren und Testen der fünf vorgestellten Klassifikatoren verwendet, welche innerhalb der *scikit-learn-API* [31] in Python implementiert sind. Tabelle 5.1 fasst die Ergebnisse dieses Versuchs zusammen. Die höchsten erreichten Werte für die Vorhersagegenauigkeit eines Klassifikators sind darin jeweils grün markiert. Es zeigt sich, dass eine hohe Genauigkeit ( $\geq 0,94$ ) erzielt wird, wenn die Basis-Features (FG 1) bei der Klassifizierung berücksichtigt werden. In Kombination mit den Aktivitätsmerkmalen konnten diese Ergebnisse im Falle des Random-Forest und für Voting leicht verbessert werden. Demgegenüber steht die Beobachtung, dass Aktivitäts- (FG 2) und Orts-Features (FG 3) einzeln und in Kombination eine relativ geringe Vorhersagegenauigkeit aufweisen ( $\leq 0,9$ ). Nur der Random-Forest-Klassifikator schneidet dabei mit bis zu 0,93 noch vergleichsweise gut ab. Insgesamt zeigt sich somit, dass die Men-

Features		Klassifikatoren				
FG	#	DT	R. Forest	SVM	MLP	Voting
1	49	0,95	0,98	0,97	0,97	0,97
2	18	0,87	0,93	0,85	0,88	0,90
3	10	0,66	0,73	0,73	0,72	0,74
1+2	67	0,95	0,99	0,97	0,97	0,98
1+3	59	0,95	0,99	0,97	0,97	0,97
2+3	28	0,85	0,92	0,88	0,89	0,90
1+2+3	77	0,94	0,99	0,97	0,97	0,98

Tabelle 5.1: Vorhersagegenauigkeiten für jede Kombination aus Feature-Gruppen und Klassifikatoren ohne vorherige Selektion. Die grün markierten Werte stellen die besten Ergebnisse pro Spalte dar.

ge der Basis-Features die wichtigsten und eindeutigsten Merkmale beinhaltet, um Nutzer adäquat ihren entsprechenden Rollen zuzuordnen. Eine allgemeine signifikante Verbesserung durch die zusätzliche Verwendung von Aktivitäts- bzw. Orts-Features kann hier nicht festgestellt werden. Dieses Erkenntnis wird im späteren Verlauf der Arbeit auf Basis realer Daten nochmals diskutiert.

Es sei darauf hingewiesen, dass die hier verwendeten simulierten Daten auf lediglich vier vordefinierten Nutzerrollen basieren. Deshalb ist eine Überanpassung der Modelle sehr wahrscheinlich, insbesondere wenn alle berechneten Features wie im vorliegenden Versuch ohne Vorselektion genutzt werden. Der vermutete Overfitting-Effekt zeigt sich vor allem in der letzten Zeile der Tabelle 5.1, in der die Vorhersagegenauigkeit aller komplexeren Klassifikatoren extrem hoch ausfällt ( $\geq 0,97$ ). Um diesen Effekt zu reduzieren und die Generalisierbarkeit der Modelle zu erhöhen, wird der eben durchgeführte Versuch mit selektierten Feature-Werten wiederholt.

Hierzu wird die Gesamtmenge der errechneten Features anhand der drei beschriebenen Verfahren jeweils auf möglichst eindeutige Merkmale reduziert, die dann zur Klassifizierung verwendet werden. Die daraus resultierenden Ergebnisse legen nahe, dass die Selektion über einen Varianz-Schwellwert von 1,0 keine wesentliche Veränderung zum vorherigen Versuch bewirkt, da insgesamt nur 17 Features aus den drei Gruppen entfernt wurden. Im Fall von  $k$ -Best-Selektion und PCA wurde  $k = 10$  gewählt, so dass alle Feature-Gruppen nach der Vorauswahl die gleiche Anzahl von zehn wichtigen Merkmalen besitzen. Nach diesen Selektionsverfahren fallen die Vorhersagegenauigkeiten für fast alle Kombinationen wie erwartet etwas geringer aus als beim vorherigen Versuch. Tabelle 5.2 fasst exemplarisch die erreichten Ergebnisse zusammen, wenn zuvor eine Feature-Selektion über PCA mit  $k = 10$  durchgeführt wurde. Die grün markierten Werte stellen wieder die besten Ergebnisse pro Spalte dar.

Generell zeigen sich die selben Tendenzen wie zuvor. So werden alle höheren Werte ( $\geq 0,94$ ) nur unter Einbeziehung der Basis-Features erreicht, während die Vorhersagegenauigkeit bei Aktivitäts- und Orts-Features oder deren Kom-

Features		Klassifikatoren				
FG	#	DT	R. Forest	SVM	MLP	Vote
1	10	0,88	0,94	0,94	0,94	0,94
2	10	0,87	0,90	0,85	0,88	0,90
3	10	0,68	0,73	0,75	0,73	0,75
1+2	20	0,92	0,96	0,96	0,97	0,97
1+3	20	0,85	0,94	0,94	0,94	0,95
2+3	20	0,84	0,90	0,84	0,88	0,88
1+2+3	30	0,87	0,96	0,96	0,96	0,97

Tabelle 5.2: Erreichte Vorhersagegenauigkeit für jede Kombination nach der Feature-Selektion über PCA mit  $k = 10$ .

bination durchgehend schlechter ausfällt. Dementsprechend lässt sich festhalten, dass Zeit- und Signalinformationen aus den reinen Rohdaten der WLAN-Beobachtungen bereits die wichtigsten Merkmale zur Klassifizierung von Nutzerrollen beinhalten. In Kombination mit den Aktivitäts-Features konnten bei diesem Versuch alle Klassifikatoren ihre Vorhersagegenauigkeit nochmals um zwei bis vier Prozentpunkte steigern. Die besten Ergebnisse von mindestens 0,92 werden somit durch die Kombination der Feature-Gruppen eins und zwei erreicht. Dabei arbeiten die komplexeren Klassifikatoren erwartungsgemäß etwas genauer, wobei Werte von 0,96 bzw. 0,97 erzielt werden, was ein passables Klassifizierungsergebnis darstellt. Somit lässt sich festhalten, dass eine adäquate Abschätzung des Bewegungszustands einer Person auch zur Klassifizierung von Nutzerrollen hilfreich sein kann. Es sei angemerkt, dass diese Ergebnisse auf simulierten Daten von lediglich vier Nutzergruppen basieren. Deshalb ist es wahrscheinlich, dass in realen Szenarien ein komplexeres Klassifizierungsproblem vorliegt und die erreichten Genauigkeiten geringer ausfallen. Dieser Punkt soll in der anschließenden Diskussion nochmals aufgegriffen werden.

Die Orts-Features hingegen führen wie beim vorherigen Versuch zur geringsten Vorhersagegenauigkeit unter allen getesteten Klassifikatoren. Dies lässt sich dadurch erklären, dass aufgrund der Positionsfehler sowie der einfachen Struktur des Gebäudes und der durchgeführten Simulationen sehr ähnliche Trajektorien in allen Nutzergruppen entstehen, welche sich nicht eindeutig zuordnen lassen. Auch diese Annahme soll nun mittels realer WLAN-Beobachtungen und ohne Vorwissen über die anwesenden Nutzer diskutiert werden.

#### 5.2.4 Diskussion anhand realer Daten

Zur Diskussion der eben vorgestellten Ergebnisse, wird das Konzept auf die realen Daten angewendet, welche bereits in Abschnitt 3.2.4 zur Modellvalidierung verwendet wurden. Das bedeutet, es liegen nun die gleichen drei Feature-Gruppen auf Basis tatsächlicher WLAN-Beobachtungen vor, die analog zur Simulation im entsprechenden Gebäude an den fünf Messstationen aufgezeichnet

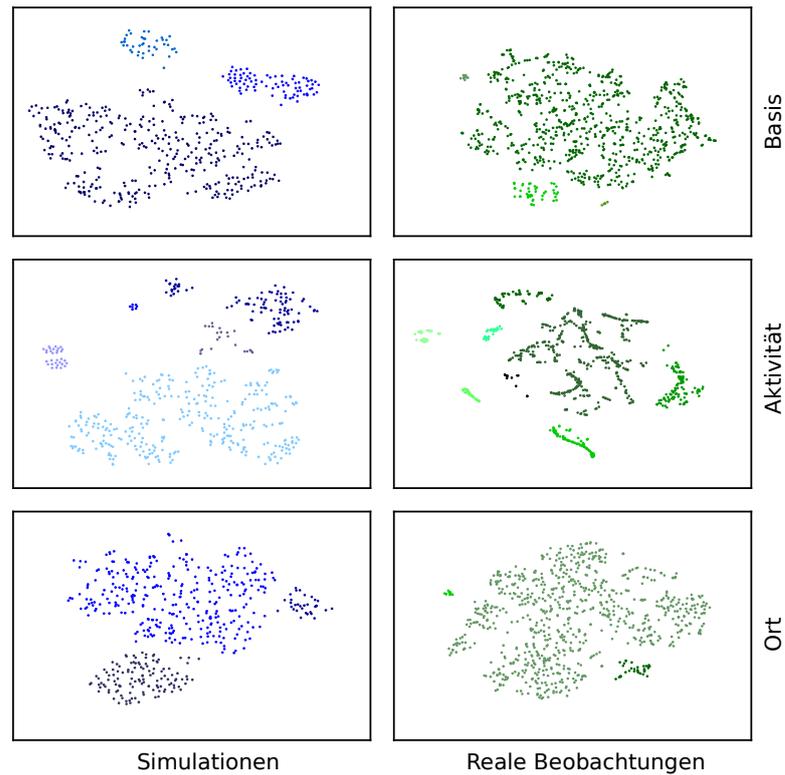


Abbildung 5.6: Vergleich von simulierten und realen Daten anhand ihrer Cluster-Tendenzen auf Basis der drei Feature-Gruppen.

wurden (vgl. Abbildung 3.7). Allerdings fehlt für diesen Datensatz die Grundwahrheit und die entsprechenden Nutzerrollen sind unbekannt. Daher können die oben genannten Klassifikatoren mit diesen Daten weder trainiert noch kann deren Vorhersagegenauigkeit validiert werden. Um trotzdem eine Diskussion der Ergebnissen aus den Simulationen durchführen zu können, werden im Folgenden gängige Verfahren aus dem unüberwachten Lernen (engl. *unsupervised learning*) wie *Manifold-Learning* bzw. *Clustering* verwendet.

Als erster Schritt werden beide Datensätze getrennt nach den einzelnen Feature-Gruppen visualisiert. Dazu müssen die multidimensionalen Feature-Vektoren zunächst auf zwei Dimension reduziert werden. Hierfür wird *t-distributed stochastic neighbor embedding* (t-SNE) verwendet, welches ebenfalls innerhalb der *scikit-learn-API* [31] zur Verfügung gestellt wird. In Abbildung 5.6 wird das entsprechende Ergebnis graphisch dargestellt, wobei die einzelnen Cluster-Tendenzen zur besseren Veranschaulichung in unterschiedlichen Farben visualisiert sind. Wegen der fehlenden Grundwahrheit können die Nutzerrollen den einzelnen Datenpunkten nicht valide zugeordnet werden. Allerdings lassen sich anhand der jeweiligen Datenzusammensetzung einige interessante Schlussfolgerungen ableiten. Es wird davon ausgegangen, dass die beobachteten Cluster-Tendenzen eine Aussage über vorhandene Nutzerrollen zulassen,

die sich in den Daten nicht direkt ablesen lassen.

Im Fall der Basis-Features (vgl. erste Reihe) zeigen die simulierten Daten leicht stärkere Zusammenhänge als die realen Daten. Dies lässt sich auch objektiv anhand einer höheren Hopkins-Statistik belegen – im Fall der Simulationen von  $H_{\text{sim}} = 0,66$  im Vergleich zu den realen Daten von  $H_{\text{real}} = 0,64$ . Dadurch wird die Vermutung bestätigt, dass die vier simulierten Nutzergruppen die Realität vereinfacht abbilden, wenngleich der festgestellte Unterschied gering ausfällt. So ist davon auszugehen, dass die vorherigen Klassifizierungsergebnisse plausibel sind, aber in der Realität etwas schlechter ausfallen.

Für die Aktivitäts-Features (vgl. zweite Reihe) werden in beiden Datensätzen die stärksten Cluster-Tendenzen von  $H_{\text{sim}} = 0,73$  und  $H_{\text{real}} = 0,81$  beobachtet. Insbesondere im Fall der realen Daten ist ein derart hoher Wert sowie die visuelle Darstellung der Daten interessant. Deshalb sollte diese Beobachtung in einer zukünftigen Studie mit einer bekannten Grundwahrheit näher untersucht werden. Zusammen mit den bisher gewonnenen Erkenntnissen lässt sich daher schlussfolgern, dass die extrahierten Informationen über Ruhe- und Bewegungsphasen einer Person geeignet erscheinen, um das unterschiedliche Nutzerverhalten genauer zu charakterisieren.

Im Fall der Orts-Features (vgl. letzte Reihe) werden hingegen die schwächsten Cluster-Tendenzen für den simulierten Datensatz von  $H_{\text{sim}} = 0,60$  festgestellt. Dies erklärt die geringe Vorhersagegenauigkeit in den vorherigen Versuchen, da die Feature-Vektoren nicht eindeutig separiert und somit zuverlässig in Klassen eingeteilt werden können. Zudem wird die Annahme bestätigt, dass die simulierten Trajektorien alle sehr ähnlich sind, wohingegen die realen Daten eine höhere Cluster-Tendenz von  $H_{\text{real}} = 0,67$  aufweisen. Der Grund könnte sein, dass tatsächliche Personenbewegungen komplexer und daher differenzierter ablaufen als es die Simulationen vorgeben. Somit liegt die Vermutung nahe, dass in realen Szenarien die Orts-Features auch zu einer verbesserten Genauigkeit bei der Klassifizierung von Nutzerrollen beitragen können. Dies muss allerdings erst durch weitere Untersuchungen bestätigt werden.

Zusammenfassend lässt sich festhalten, dass die beiden Datensätze vergleichbare Tendenzen bzgl. der Cluster-Bildung aufweisen, was die Realitätstreue der simulierten Daten unterstreicht. Anhand der Diskussion konnte dargelegt werden, dass die oben gewonnen Erkenntnisse entsprechend ihre Gültigkeit behalten und dass das hier vorgestellte Konzept auch in realen Szenarien anwendbar ist. Dabei zeigten vor allem die Basis- und Aktivitäts-Features eine bessere Eignung zur Klassifizierung von Nutzerrollen. Für eine genaue Bewertung der Validität müssen die Ergebnisse noch mit einer Grundwahrheit überprüft werden. Dies kann bspw. die Basis zukünftiger Arbeiten sein, in denen mit Methoden des überwachten Lernens neue Erkenntnisse über vorhandene Nutzergruppen gewonnen werden können.

## 5.3 Zusammenfassung

In diesem Kapitel wurde untersucht, inwieweit Aktivitäts- und Identitätsinformationen aus aufgezeichneten WLAN-Beobachtungen gewonnen werden können und wie verlässlich solche Informationen sind. Hierbei lag der Fokus auf der Erkennung von einfachen Aktivitäten wie *Stehen* bzw. *Gehen*, welche über dem Zeitverlauf die Ruhe- und Bewegungsphasen eines Nutzers beschreiben. Eine derartig genaue Unterscheidung ist auf Basis von Positionsdaten nur schwer möglich, weshalb in diesem Kapitel ein anderes Verfahren vorgeschlagen und evaluiert wurde. Dabei werden die Unterschiede der Empfangssignalstärken im Zeitverlauf betrachtet und dem entsprechenden Bewegungszustand zugeordnet. Es hat sich herausgestellt, dass ein Viterbi-basierter Ansatz unter Berücksichtigung der Euklidischen Distanz und mehreren Beobachtungen das beste Ergebnis unter allen Versuchen erzielt. Dabei wurde eine Verbesserung von 21% gegenüber der Referenzimplementierung erreicht. Durch die hohe Genauigkeit dieses Verfahrens konnte zudem für die meisten Nutzergruppen die durchschnittliche Abweichung bei der Pfadermittlung reduziert werden. Hierzu wurde das Bewegungsmodell des State-Partikel-Filters leicht modifiziert und mit der extrahierten Aktivitätsinformation angereichert. Im Fall einer ungenaueren Aktivitätserkennung kann sich dieser Effekt jedoch auch ins Negative umkehren, wie die durchgeführte Evaluation gezeigt hat.

Im zweiten Teil des Kapitels wurden alle bisherigen Informationen über Zeit, Signalstärken, Ort und Aktivität genutzt, um die Identität eines Nutzers zu ermitteln. Dabei lag der Fokus auf der Klassifizierung von Nutzerrollen, also der Einordnung der relevanten Informationen in bestimmte Klassen, die eine Rolle beschreiben. Hierzu wurden insgesamt 77 Features in drei Informationsgruppen errechnet und für den Klassifizierungsschritt auf Basis geeigneter Algorithmen des maschinellen Lernens verwendet. Es zeigte sich, dass die Basis-Features wie Zeit und Empfangssignalstärke, die bereits aus den Rohdaten extrahiert werden können, die essentielle Information zur Klassifizierung von Nutzerrollen beinhalten. Durch die Kombination mit den Aktivitäts-Features konnte die Klassifizierungsgenauigkeit zusätzlich um ca. 3% gesteigert werden. Eine anschließende Diskussion anhand tatsächlicher WLAN-Aufzeichnungen ergab zudem, dass die Orts-Information in realen Szenarien u.U. zu einer genaueren Klassifizierung beitragen kann. Zudem liegt die Vermutung nahe, dass in der Realität die Klassifizierungsgenauigkeit etwas schlechter ausfallen könnte, da das reale Nutzerverhalten komplexer ist als in der Simulation. Insgesamt komplettieren die hier vorgestellten Methoden das Konzept zur Extraktion primärer Kontextinformationen aus aufgezeichneten WLAN-Daten.



# 6 Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde ein Gesamtkonzept entwickelt, um die vier primären Typen von Kontextinformationen (Zeit, Ort, Aktivität und Identität) nach Abowd et al. [2] aus aufgezeichneten WLAN-Daten zu extrahieren. Solche Informationen können unter anderem für eine genaue Analyse von Nutzerströmen verwendet werden, ohne dass die aktive Beteiligung oder gar das Einverständnis der Nutzer erforderlich ist. Dies stoßt vor allem bei Gebäude- und Infrastrukturbetreibern auf großes Interesse, da sie anderweitig nur schwer an die begehrten Informationen ihrer Nutzer herankommen. Gerade die vielen Startup-Unternehmen, die WLAN-Tracking zur Kundenanalyse als Business-Lösung für den Einzelhandel anbieten, belegen die aktuell hohe Nachfrage an der vorgestellten Technik und den angebotenen Lösungen. Allerdings fehlte bisher eine genauere wissenschaftliche Untersuchung bzgl. der Güte solcher Verfahren. Im Rahmen der vorliegenden Arbeit wurde ein Beitrag geleistet, um diese Wissenslücken zu schließen. Zudem wurden neuartige Methoden zur Extraktion von Kontextinformationen entwickelt und evaluiert. Die wichtigsten Ergebnisse hieraus sollen im Folgenden zusammengefasst werden.

## 6.1 Ergebnisse der Arbeit

Zunächst wurde im Rahmen der Arbeit untersucht, wie und welche Informationen sich über aufgezeichnete Bluetooth- bzw. WLAN-Signale erfassen lassen. Hierzu wurde ein Literaturüberblick gegeben und eigene Untersuchungen am Münchner Flughafen durchgeführt. Dabei zeigte sich, dass sehr viele Daten in kurzer Zeit gesammelt werden, wobei das Verhältnis erfasster Bluetooth/WLAN Geräte zwischen 2,8% und 4% lag. Über die aufgezeichneten MAC-Adressen wurde zudem die Verteilung der Geräte nach Herstellerkennung analysiert. Hierbei dominierten Apple-Geräte, da diese eine aktivere Netzwerksuche betreiben. Im Anschluss daran wurden vier Ansätze zur Bestimmung von Nutzerströmen auf Basis einer einfachen Nahbereichserkennung vorgestellt. Zur Evaluation wurden Bluetooth- und WLAN-Signale vor und nach der Sicherheitskontrolle des Flughafens aufgezeichnet und die errechneten Ströme mit den zur Verfügung gestellten Bordkarten-Scans als Grundwahrheit verglichen. Es wurde deutlich, dass WLAN die besseren Ergebnisse liefert und dass ein bloßes Zählen der MAC-Adressen nur zu einem moderaten Zusammenhang führt, obwohl dies in vielen anderen Arbeiten propagiert wird.

Die besten Resultate wurden unter Berücksichtigung der Domäne und der Anwendung des hybriden Ansatzes erreicht, der im Durchschnitt eine starke Korrelation von 0,75 zur Grundwahrheit aufzeigte. Auf Basis der gewonnenen Erkenntnisse konnte im Anschluss eine Simulationsumgebung für WLAN-Beobachtungen entwickelt werden, die ein neuartiges Modell zur Aussendung von Probe Requests beinhaltet. Zur Bewertung der Realitätstreue wurde eine ergebnisbezogene Modellvalidierung durchgeführt. Dabei zeigten die realen WLAN-Beobachtungen eine höhere Anzahl an Ausreißern, aber gleichzeitig eine im Durchschnitt geringere Beobachtungsfrequenz als die Simulationen. Dieser festgestellte Verhaltensunterschied ist aufgrund des schwer vorhersagbaren Probe-Verhaltens als akzeptabel einzustufen.

Nachdem die Datenerfassung intensiv untersucht wurde, beschäftigten sich die restlichen Kapitel umfassend mit der Extraktion von primären Kontextinformationen aus WLAN-Daten. Hierbei stand zunächst die Ortsinformation im Fokus, welche bei der Nutzerstromanalyse die wichtigste Kontextinformation darstellt. Um diese zu erhalten, wurde eine kontinuierliche Positionierung in einem Gebäude auf Basis der aufgezeichneten Beobachtungsvektoren angestrebt. Zu diesem Zweck mussten zuvor Messknoten auf geeignete Weise verteilt werden. Das Ziel dabei war es, Hotspots zu identifizieren, so dass mit möglichst wenigen Knoten möglichst viele Trajektorien erfasst werden konnten. Im Rahmen der Arbeit wurden hierzu drei geometrische Ansätze vorgestellt, welche auf Basis simulierter Nutzerpfade evaluiert wurden. Dabei zeigte sich, dass nur das dritte Verfahren über die Berechnung repräsentativer Trajektorien und einem Dichte-basierten Clustering ein etwas besseres Resultat erzielen konnte als die Brute-Force-Methode. Auf Basis des Ergebnisses wurden anschließend Messknoten in einem Universitätsgebäude verteilt und zur Aufzeichnung der Beobachtungsvektoren für die Pfadermittlung verwendet. Bei der deterministischen Positionierung über einen gewichteten kNN-Ansatz zeigten sich sehr hohe Distanzfehler sowie eine geringe Präzision und eine niedrige Robustheit. Die probabilistischen Methoden erzielten hingegen bessere Ergebnisse, weshalb nur solche Verfahren für die Standortverfolgung in Betracht kommen. Im Rahmen der Arbeit wurde zudem der State-Partikel-Filter als eine neuartige hybride Methode entwickelt, der das diskrete Zustandsmodell des Viterbi-Ansatzes mit dem Bewegungsmodell des SIR-Partikel-Filters für Freiflächen kombiniert. Während der Evaluation zeigte sich deutlich, dass ein eingeschränkter Lösungsraum besser geeignet ist, als eine Berücksichtigung aller möglichen Positionen. So lieferte der State-Partikel-Filter in Bezug auf Genauigkeit, Präzision und Robustheit die besten Resultate. Zudem wurden deutlich weniger Partikel als beim SIR-Ansatz benötigt, was zu einer verbesserten Skalierbarkeit führt. Dies ist gerade bei der Nutzerstromerfassung von Vorteil, da hier viele Personen verfolgt werden müssen.

Nach der Ortsinformation standen im letzten Kapitel die Aktivitäts- und Identitätsinformation im Fokus der Untersuchungen. Hierzu wurde ein Verfahren vorgestellt, welches Ruhe- und Bewegungsphasen eines Nutzers auf Grund-

lage der auftretenden Signalvarianzen im Zeitverlauf bestimmen kann. Zudem wurde zwischen einer deterministischen und einer probabilistischen Entscheidungsfindung unterschieden. Die beiden Varianten wurden auf Basis simulierter Nutzergruppen mittels verschiedener Distanzfunktionen evaluiert. Es zeigte sich, dass ein Schwellwert-basiertes Verfahren nicht die nötige Flexibilität besitzt, um verlässliche Resultate zu liefern. Der eigene Viterbi-basierte Ansatz erreichte hingegen adäquate Ergebnisse für die meisten Nutzergruppen. Im Vergleich zu einem implementierten Referenzsystem erzielte das hier entwickelte Verfahren eine durchschnittliche Verbesserung von 21% im F-Maß. Zudem konnte der Lokalisierungsfehler bei der probabilistischen Pfadermittlung mithilfe des abgeschätzten Bewegungszustands verringert werden. Auf Basis der somit zur Verfügung stehenden Informationen wie Zeit, Ort und Aktivität wurde abschließend eine Klassifizierung von Nutzerrollen durchgeführt, die der Identitätserkennung dient. Hierzu wurden für die drei Informationsgruppen zunächst statistische Eigenschaften berechnet. Auf Basis geeigneter Lernalgorithmen wurde dann untersucht, welche Kombination aus Feature-Gruppen und Klassifikator die höchste Vorhersagegenauigkeit liefert. Es stellte sich heraus, dass die essentiellen Informationen bereits in den Basis-Features enthalten sind. Über eine Kombination mit den Aktivitäts-Features konnte die Genauigkeit nochmals um ca. 3% gesteigert werden und lag im besten Fall bei 97%. Da diese Ergebnisse auf simulierten Daten erzielt wurden, die ggf. eine Vereinfachung des Problems darstellen, wurde eine Diskussion auf Basis realer Daten durchgeführt. Hierbei wurden ähnliche Cluster-Tendenzen festgestellt, was die Validität der Untersuchungen unterstreicht.

## 6.2 Ausblick

In der vorliegenden Arbeit wurde gezeigt, dass eine Extraktion primärer Kontextinformationen aus WLAN-Daten ohne aktiver Beteiligung der Nutzer mit geringem technischen Aufwand und ohne Modifikation der gängigen Hard- oder Software möglich ist. Dadurch entstehen zugleich große Risiken für die Privatsphäre von Nutzern mobiler Geräte. In der Literatur existieren Arbeiten, die auf diese Risiken aufmerksam machen [42, 48, 115], jedoch fehlen bislang fundierte Untersuchungen und flächendeckende Schutzmaßnahmen. Die großen Anbieter mobiler Plattformen wie Apple oder Google bieten seit jüngster Zeit einen Mechanismus zur Randomisierung der MAC-Adresse an, um Verfahren, wie sie in der vorliegenden Arbeit entwickelt wurden, zumindest zu erschweren. Es zeigt sich aber, dass ein derartiger Schutz noch nicht ausreichend ist und umgangen werden kann [136, 161, 203]. In zukünftigen Arbeiten ist es demnach wichtig, diese Aspekte bei der Extraktion von Informationen zu berücksichtigen. Dabei stellt sich bspw. die Frage, ob WLAN weiterhin als Tracking-Technologie funktionieren kann und inwieweit die hier entwickelten Ansätze ihre Gültigkeit behalten oder auf andere Techniken adaptiert werden können, sollten in Zukunft weitere Schutzmechanismen etabliert werden.

In den durchgeführten Versuchen und Evaluationen wurde meistens ein einfaches Gebäude herangezogen, das nicht die Komplexität großer öffentlicher Einrichtungen wie bspw. Flughäfen oder Bahnhöfe aufweist. Zudem wurden bisher keine Stockwerkwechsel betrachtet. Der Grund hierfür lag vor allem in der Machbarkeit, reale WLAN-Daten in einem bekannten und abgeschlossenen Szenario aufzuzeichnen. Zukünftige Arbeiten sollten daher komplexere Umgebungen wählen, um die hier entwickelten Verfahren genauer validieren zu können und ggf. zu erweitern. Allerdings wird dabei der Zugang zu einer verlässlichen Grundwahrheit erneut eine große Herausforderung darstellen. Mithilfe einer flächendeckenden Videoüberwachung und genauen visuellen Verfahren zur Standortverfolgung ließe sich dieses Problem lösen. Damit wären auch weitere Untersuchungen bzgl. Klassifizierung realer Nutzerrollen mittels Verfahren des überwachten Lernens möglich, die in der vorliegenden Arbeit nur diskutiert werden konnten. Hier ergeben sich für die Nutzerstromanalyse viele interessante Aspekte. Darunter zählt neben der Bildung und Zuweisung von Cluster-Bezeichnungen auch das Trainieren und Testen von Lernalgorithmen auf das komplexe menschliche Bewegungsverhalten. In diesem Bereich wird generell ein großes Potential für zukünftige Arbeiten gesehen.

Ein weiterer Aspekt, der in der vorliegenden Arbeit noch nicht behandelt wurde, ist die Erkennung komplexerer Aktivitäten aus WLAN-Beobachtungen. Hier ließe sich bspw. untersuchen, ob auf Basis der Signalvarianzen auch detailliertere Bewegungen wie *Laufen* oder *Fahrradfahren* zuverlässig bestimmt werden können, indem die Beobachtungen feingranularer unterteilt werden. Zudem ist es interessant, den Einfluss des Nutzungsverhaltens mobiler Geräte auf die Erkennungsrate solcher Aktivitäten zu evaluieren, da bspw. ein Smartphone eher selten genutzt wird, während der Nutzer Fahrrad fährt. Hierzu eignen sich Versuche, in denen das Testgerät in der Hosentasche, im Rucksack oder in der Hand mitgeführt wird. In zukünftigen Studien können solche Untersuchungen weiter ausgebaut und evaluiert werden.

# Literaturverzeichnis

- [1] ABEDI, N., A. BHASKAR und E. CHUNG: *Bluetooth and Wi-Fi MAC Address Based Crowd Data Collection and Monitoring: Benefits, Challenges and Enhancement*. In: *Australasian Transport Research Forum (ATRF), 36th, 2013, Brisbane, Queensland, Australia*, 2013.
- [2] ABOWD, G. D., A. K. DEY, P. J. BROWN, N. DAVIES, M. SMITH und P. STEGGLES: *Towards a better understanding of context and context-awareness*. In: *International Symposium on Handheld and Ubiquitous Computing*, S. 304–307. Springer, 1999.
- [3] ADAMS, N., R. GOLD, B. N. SCHILIT, M. M. TSO und R. WANT: *An infrared network for mobile computers*. In: *Proceedings USENIX Symposium on Mobile & Location-independent Computing*, Bd. 10, 1993.
- [4] AHMED, T., T. B. PEDERSEN und H. LU: *Capturing hotspots for constrained indoor movement*. In: *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, S. 472–475. ACM, 2013.
- [5] AL-ALI, A., F. A. ALOUL, N. R. AJI, A. A. AL-ZAROUNI und N. H. FAKHRO: *Mobile RFID tracking system*. In: *Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on*, S. 1–4. IEEE, 2008.
- [6] ALT, H. und M. GODAU: *Computing the Fréchet distance between two polygonal curves*. *International Journal of Computational Geometry & Applications*, 5(01n02):75–91, 1995.
- [7] ANDROIDDEVELOPERS: *Bluetooth*. <https://developer.android.com/guide/topics/connectivity/bluetooth.html>, 2017. letzter Abruf: 02.11.2017.
- [8] ANGUELOV, D., K.-C. LEE, S. B. GOKTURK und B. SUMENGEN: *Contextual identity recognition in personal photo albums*. In: *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, S. 1–7. IEEE, 2007.
- [9] ANKERST, M., M. M. BREUNIG, H.-P. KRIEGEL und J. SANDER: *OPTICS: ordering points to identify the clustering structure*. In: *ACM Sigmod record*, Bd. 28, S. 49–60. ACM, 1999.

- [10] ARONOV, B., A. DRIEMEL, M. V. KREVELD, M. LÖFFLER und F. STAALS: *Segmentation of trajectories on nonmonotone criteria*. ACM Transactions on Algorithms (TALG), 12(2):26, 2016.
- [11] ARORA, S., C. LUND, R. MOTWANI, M. SUDAN und M. SZEGEDY: *Proof verification and the hardness of approximation problems*. Journal of the ACM (JACM), 45(3):501–555, 1998.
- [12] AVCI, A., S. BOSCH, M. MARIN-PERIANU, R. MARIN-PERIANU und P. HAVINGA: *Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey*. In: *Architecture of computing systems (ARCS), 2010 23rd international conference on*, S. 1–10. VDE, 2010.
- [13] AXEL, K. et al.: *Location-based services: fundamentals and operation*. John Wiley & Sons, 2005.
- [14] AZIZYAN, M., I. CONSTANDACHE und R. ROY CHOUDHURY: *Surround-Sense: mobile phone localization via ambience fingerprinting*. In: *Proceedings of the 15th annual international conference on Mobile computing and networking*, S. 261–272. ACM, 2009.
- [15] BA, S. O. und J.-M. ODOBEZ: *A Rao-Blackwellized mixed state particle filter for head pose tracking*. In: *ACM ICMI Workshop on Multimodal Multiparty Meeting Processing (MMMP)*, Nr. EPFL-CONF-83238, 2005.
- [16] BAHL, P. und V. N. PADMANABHAN: *RADAR: An in-building RF-based user location and tracking system*. In: *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Bd. 2, S. 775–784. IEEE, 2000.
- [17] BAHL, P., V. N. PADMANABHAN und A. BALACHANDRAN: *Enhancements to the RADAR user location and tracking system*. Microsoft Research, 2(MSR-TR-2000-12):775–784, 2000.
- [18] BAI, F. und A. HELMY: *A survey of mobility models*. Wireless Adhoc Networks. University of Southern California, USA, 206:147, 2004.
- [19] BANERJEE, A. und R. N. DAVE: *Validating clusters using the Hopkins statistic*. In: *Fuzzy systems, 2004. Proceedings. 2004 IEEE international conference on*, Bd. 1, S. 149–153. IEEE, 2004.
- [20] BARBERA, M. V., A. EPASTO, A. MEI, V. C. PERTA und J. STEFA: *Signals from the crowd: uncovering social relationships through smartphone probes*. In: *Proceedings of the 2013 conference on Internet measurement conference*, S. 265–276. ACM, 2013.

- 
- [21] BARTOLETTI, S., A. CONTI, A. GIORGETTI und M. Z. WIN: *Sensor radar networks for indoor tracking*. *Wireless Communications Letters*, IEEE, 3(2):157–160, 2014.
- [22] BAUM, L. E. und T. PETRIE: *Statistical inference for probabilistic functions of finite state Markov chains*. *The annals of mathematical statistics*, 37(6):1554–1563, 1966.
- [23] BLESER, G. und D. STRICKER: *Advanced tracking through efficient image processing and visual–inertial sensor fusion*. *Computers & Graphics*, 33(1):59–72, 2009.
- [24] BLUETOOTH, S.: *Specification of the Bluetooth System, version 1.2*. Nov, 2003.
- [25] BOGGS, J. A.: *Geolocation of an audio source in a multipath environment using time-of-arrival*. Techn. Ber., DTIC Document, 2004.
- [26] BONNÉ, B., A. BARZAN, P. QUAX und W. LAMOTTE: *WiFiPi: Involuntary tracking of visitors at mass events*. In: *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*, S. 1–6. IEEE, 2013.
- [27] BOSE, A. und C. H. FOH: *A practical path loss model for indoor WiFi positioning enhancement*. In: *Information, Communications & Signal Processing, 2007 6th International Conference on*, S. 1–5. IEEE, 2007.
- [28] BROOKS, A., A. MAKARENKO, T. KAUPP, S. WILLIAMS und H. DURRANT-WHYTE: *Implementation of an indoor active sensor network*. In: *Experimental Robotics IX*, S. 397–406. Springer, 2006.
- [29] BUCHHOLZ, P.: *Validierung von Modellen*. [http://ls4-www.cs.tu-dortmund.de/download/buchholz/MA0/MA0\\_8.pdf](http://ls4-www.cs.tu-dortmund.de/download/buchholz/MA0/MA0_8.pdf), 2016. letzter Ab-ruf: 02.11.2017.
- [30] BUCHIN, M., A. DRIEMEL, M. VAN KREVELD und V. SACRISTÁN: *An algorithmic framework for segmenting trajectories based on spatio-temporal criteria*. In: *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, S. 202–211. ACM, 2010.
- [31] BUITINCK, L., G. LOUPPE, M. BLONDEL, F. PEDREGOSA, A. MUELLER, O. GRISEL, V. NICULAE, P. PRETTENHOFER, A. GRAMFORT, J. GROBLER, R. LAYTON, J. VANDERPLAS, A. JOLY, B. HOLT und G. VAROQUAUX: *API design for machine learning software: experiences from the scikit-learn project*. In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, S. 108–122, 2013.

- [32] BULUSU, N., J. HEIDEMANN und D. ESTRIN: *Adaptive beacon placement*. In: *Distributed Computing Systems, 2001. 21st International Conference On.*, S. 489–498. IEEE, 2001.
- [33] CHAKRABORTY, G., K. NAIK, D. CHAKRABORTY, N. SHIRATORI und D. WEI: *Analysis of the Bluetooth device discovery protocol*. *Wireless Networks*, 16(2):421–436, 2010.
- [34] CHAWATHE, S. S.: *Beacon placement for indoor localization using bluetooth*. In: *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, S. 980–985. IEEE, 2008.
- [35] CHEN, Q., B. TAN, K. WOODBRIDGE und K. CHETTY: *Indoor target tracking using high doppler resolution passive Wi-Fi radar*. In: *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, S. 5565–5569. IEEE, 2015.
- [36] CHEN, Z., H. T. SHEN und X. ZHOU: *Discovering popular routes from trajectories*. In: *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*, S. 900–911. IEEE, 2011.
- [37] CHIP: *Runtastic PRO GPS Laufen, Joggen und Fitness Tracker*. <http://beste-apps.chip.de/ios/app/runtastic-pro-iphone-app,366626332/>. letzter Abruf: 02.11.2017.
- [38] CHIP: *WakeApp - der wissenschaftliche Wecker und Schlafrekorder*. <http://beste-apps.chip.de/ios/app/wakeapp-schlafphasenueberwachen-und-angenehm-aufwachen,392358256/>. letzter Abruf: 02.11.2017.
- [39] CHOI, M.-S. und B. JANG: *An Accurate Fingerprinting based Indoor Positioning Algorithm*. *International Journal of Applied Engineering Research*, 12(1):86–90, 2017.
- [40] CIURANA, M., F. BARCELO-ARROYO und F. IZQUIERDO: *A ranging system with IEEE 802.11 data frames*. In: *Radio and Wireless Symposium, 2007 IEEE*, S. 133–136. IEEE, 2007.
- [41] CIURANA, M., D. LÓPEZ und F. BARCELÓ-ARROYO: *SofTOA: Software ranging for TOA-based positioning of WLAN terminals*. In: *International Symposium on Location-and Context-Awareness*, S. 207–221. Springer, 2009.
- [42] CUNCHE, M.: *I know your MAC Address: Targeted tracking of individual using Wi-Fi*. *Journal of Computer Virology and Hacking Techniques*, 10(4):219–227, 2014.

- [43] CUNCHE, M., M.-A. KAAFAR und R. BORELI: *Linking wireless devices using information contained in Wi-Fi probe requests*. Pervasive and Mobile Computing, 11:56–69, 2014.
- [44] DAI-LABOR: *Kontextsensitive und Adaptive Dienste*. [http://www.dai-labor.de/ngs/forschung/kontextadaptive\\_dienste/](http://www.dai-labor.de/ngs/forschung/kontextadaptive_dienste/). letzter Abruf: 02.11.2017.
- [45] DANALET, A., M. BIERLAIRE und B. FAROOQ: *Estimating Pedestrian Destinations using Traces from WiFi Infrastructures*. In: *Pedestrian and Evacuation Dynamics 2012*, S. 1341–1352. Springer, 2014.
- [46] DANCEY, C. P. und J. REIDY: *Statistics without maths for psychology*. Pearson Education, 2007.
- [47] DANELLJAN, M., A. ROBINSON, F. S. KHAN und M. FELSBERG: *Beyond correlation filters: Learning continuous convolution operators for visual tracking*. In: *European Conference on Computer Vision*, S. 472–488. Springer, 2016.
- [48] DEMIR, L.: *Wi-Fi tracking: what about privacy*. Doktorarbeit, M2 SCCI Security, Cryptology and Coding of Information-UFR IMAG, 2013.
- [49] DEUTSCHESTARTUPS: *Marktübersicht - Smart Home kommt im Alltag an*. <https://www.deutsche-startups.de/2015/04/13/smart-home-kommt-im-alltag/>, 2017. letzter Abruf: 02.11.2017.
- [50] DHILLON, S. S., K. CHAKRABARTY und S. IYENGAR: *Sensor placement for grid coverage under imprecise detections*. In: *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, Bd. 2, S. 1581–1587. IEEE, 2002.
- [51] DODGE, S., R. WEIBEL und A.-K. LAUTENSCHÜTZ: *Towards a taxonomy of movement patterns*. Information visualization, 7(3-4):240–252, 2008.
- [52] DOUGLAS, D. H. und T. K. PEUCKER: *Algorithms for the reduction of the number of points required to represent a digitized line or its caricature*. Cartographica: The International Journal for Geographic Information and Geovisualization, 10(2):112–122, 1973.
- [53] EIDENBENZ, S., C. STAMM und P. WIDMAYER: *Inapproximability results for guarding polygons and terrains*. Algorithmica, 31(1):79–113, 2001.
- [54] EITER, T. und H. MANNILA: *Computing discrete Fréchet distance*. Techn. Ber., Citeseer, 1994.

- [55] ELMENREICH, W.: *Sensor fusion in time-triggered systems*. 2002.
- [56] ESTER, M., H.-P. KRIEGEL, J. SANDER, X. XU et al.: *A density-based algorithm for discovering clusters in large spatial databases with noise..* In: *Kdd*, Bd. 96, S. 226–231, 1996.
- [57] FANG, S.-H., T.-N. LIN und K.-C. LEE: *A novel algorithm for multipath fingerprinting in indoor WLAN environments*. *IEEE transactions on wireless communications*, 7(9), 2008.
- [58] FARKAS, K., Á. HUSZÁK und G. GÓDOR: *Optimization of Wi-Fi access point placement for indoor localization*. *Journal IIT (Informatics & IT Today)*, 1(1):28–33, 2013.
- [59] FEARON, J. D.: *What is identity (as we now use the word)*. Unpublished manuscript, Stanford University, Stanford, Calif, 1999.
- [60] FINK, G. A.: *Markov models for pattern recognition: from theory to applications*. Springer Science & Business Media, 2014.
- [61] FORNEY, G. D.: *The viterbi algorithm*. *Proceedings of the IEEE*, 61(3):268–278, 1973.
- [62] FREUDIGER, J.: *How talkative is your mobile device?: an experimental study of Wi-Fi probe requests*. In: *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, S. 8. ACM, 2015.
- [63] GADE, R., A. JØRGENSEN, M. M. JENSEN, T. ALLDIECK, M. ABOU-ZLEIKHA, M. G. CHRISTENSEN, T. B. MOESLUND, M. K. POULSEN, R. G. LARSEN, J. FRANCH et al.: *Automatic analysis of activities in sports arenas using thermal cameras*. In: *2016 12th International Conference on Signal-image Technology & Internet-based Systems (sitis)*. IEEE Computer Society Press, 2016.
- [64] GAFFNEY, S. und P. SMYTH: *Trajectory clustering with mixtures of regression models*. In: *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, S. 63–72. ACM, 1999.
- [65] GENTNER, C., E. MUÑOZ, M. KHIDER, E. STAUDINGER, S. SAND und A. DAMMANN: *Particle filter based positioning with 3GPP-LTE in indoor environments*. In: *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, S. 301–308. IEEE, 2012.
- [66] GIANNOTTI, F., M. NANNI, F. PINELLI und D. PEDRESCHI: *Trajectory pattern mining*. In: *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, S. 330–339. ACM, 2007.

- 
- [67] GIUSTINIANO, D. und S. MANGOLD: *CAESAR: carrier sense-based ranging in off-the-shelf 802.11 wireless LAN*. In: *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, S. 10. ACM, 2011.
- [68] GOLDEN, S. A. und S. S. BATEMAN: *Sensor measurements for Wi-Fi location with emphasis on time-of-arrival ranging*. *IEEE Transactions on Mobile Computing*, 6(10), 2007.
- [69] GONZÁLEZ-BAÑOS, H.: *A randomized art-gallery algorithm for sensor placement*. In: *Proceedings of the seventeenth annual symposium on Computational geometry*, S. 232–240. ACM, 2001.
- [70] GOODALL, N. J.: *Fundamental Characteristics of Wi-Fi and Wireless Local Area Network Re-identification for Transportation*. IET Intelligent Transport Systems, 2016.
- [71] GOUTTE, C. und E. GAUSSIER: *A probabilistic interpretation of precision, recall and F-score, with implication for evaluation*. In: *European Conference on Information Retrieval*, S. 345–359. Springer, 2005.
- [72] GRÜNWARD, P. D., I. J. MYUNG und M. A. PITT: *Advances in minimum description length: Theory and applications*. MIT press, 2005.
- [73] GU, Y., A. LO und I. NIEMEGERES: *A survey of indoor positioning systems for wireless personal networks*. *IEEE Communications surveys & tutorials*, 11(1):13–32, 2009.
- [74] GUDMUNDSSON, J., M. VAN KREVELD und F. STAALS: *Algorithms for hotspot computation on trajectory data*. In: *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, S. 134–143. ACM, 2013.
- [75] GUDMUNDSSON, J., P. LAUBE und T. WOLLE: *Movement patterns in spatio-temporal data*. In: *Encyclopedia of GIS*, S. 726–732. Springer, 2008.
- [76] GUNAWAN, M., B. LI, T. GALLAGHER, A. G. DEMPSTER und G. RETSCHER: *A new method to generate and maintain a WiFi fingerprinting database automatically by using RFID*. In: *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, S. 1–6. IEEE, 2012.
- [77] GUSTAFSSON, F. und F. GUNNARSSON: *Positioning using time-difference of arrival measurements*. In: *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, Bd. 6, S. VI–553. IEEE, 2003.

- [78] GUSTAFSSON, F., F. GUNNARSSON, N. BERGMAN, U. FORSSELL, J. JANSSON, R. KARLSSON und P.-J. NORDLUND: *Particle filters for positioning, navigation, and tracking*. IEEE Transactions on signal processing, 50(2):425–437, 2002.
- [79] HAENELT, K.: *Ähnlichkeitsmaße für vektoren*. [http://kontext.fraunhofer.de/haenelt/kurs/fohlen/Haenelt\\_VektorAehnlichkeit.pdf](http://kontext.fraunhofer.de/haenelt/kurs/fohlen/Haenelt_VektorAehnlichkeit.pdf), 2007. letzter Abruf: 02.11.2017.
- [80] HAN, J., J. PEI und M. KAMBER: *Data mining: concepts and techniques*. Elsevier, 2011.
- [81] HAR-PELED, S. et al.: *New similarity measures between polylines with applications to morphing and polygon sweeping*. Discrete & Computational Geometry, 28(4):535–569, 2002.
- [82] HARLE, R.: *A Survey of Indoor Inertial Positioning Systems for Pedestrians*. IEEE Communications Surveys and Tutorials, 15(3):1281–1293, 2013.
- [83] HAY, S. und R. HARLE: *Bluetooth tracking without discoverability*. In: *International Symposium on Location-and Context-Awareness*, S. 120–137. Springer, 2009.
- [84] HERSHBERGER, J. E. und J. SNOEYINK: *Speeding up the Douglas-Peucker line-simplification algorithm*. University of British Columbia, Department of Computer Science, 1992.
- [85] HIGHTOWER, J. und G. BORRIELLO: *Location systems for ubiquitous computing*. Computer, 34(8):57–66, 2001.
- [86] HONG, H., C. LUO und M. C. CHAN: *SocialProbe: Understanding Social Interaction Through Passive WiFi Monitoring*. In: *Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, S. 94–103. ACM, 2016.
- [87] HONKAVIRTA, V., T. PERALA, S. ALI-LOYTTY und R. PICHE: *A comparative survey of WLAN location fingerprinting methods*. In: *Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on*, S. 243–251. IEEE, 2009.
- [88] HOSSAIN, A. M. und W.-S. SOH: *A survey of calibration-free indoor positioning systems*. Computer Communications, 66:1–13, 2015.
- [89] HOTEIT, S., S. SECCI, S. SOBOLEVSKY, C. RATTI und G. PUJOLLE: *Estimating human trajectories and hotspots through mobile phone data*. Computer Networks, 64:296–307, 2014.

- 
- [90] HUSTED, N. und S. MYERS: *Mobile location tracking in metro areas: malnets and others*. In: *Proceedings of the 17th ACM conference on Computer and communications security*, S. 85–96. ACM, 2010.
- [91] IEEE COMPUTER SOCIETY, 3 Park Avenue, NY 10016-5997, USA: *IEEE Std 802.15.1-2005: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specification for Wireless Personal Area Networks (WPANs)*, June 2005.
- [92] IEEE COMPUTER SOCIETY, 3 Park Avenue, NY 10016-5997, USA: *IEEE Std 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, June 2007.
- [93] ISHIZUKA, M. und M. AIDA: *Performance study of node placement in sensor networks*. In: *Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference on*, S. 598–603. IEEE, 2004.
- [94] IZAKIAN, Z., M. S. MESGARI und A. ABRAHAM: *Automated clustering of trajectory data using a particle swarm optimization*. *Computers, Environment and Urban Systems*, 55:55–65, 2016.
- [95] JAIN, A. K., M. N. MURTY und P. J. FLYNN: *Data clustering: a review*. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.
- [96] JARDOSH, A., E. M. BELDING-ROYER, K. C. ALMEROTH und S. SURI: *Towards realistic mobility models for mobile ad hoc networks*. In: *Proceedings of the 9th annual international conference on Mobile computing and networking*, S. 217–229. ACM, 2003.
- [97] JEUNG, H., M. L. YIU und C. S. JENSEN: *Trajectory pattern mining*. In: *Computing with spatial trajectories*, S. 143–177. Springer, 2011.
- [98] JIANG, Y., X. PAN, K. LI, Q. LV, R. P. DICK, M. HANNIGAN und L. SHANG: *Ariel: Automatic wi-fi based room fingerprinting for indoor localization*. In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, S. 441–450. ACM, 2012.
- [99] KALAYEH, M. M., S. MUSSMANN, A. PETRAKOVA, N. D. V. LOBO und M. SHAH: *Understanding Trajectory Behavior: A Motion Pattern Approach*. arXiv preprint arXiv:1501.00614, 2015.
- [100] KALNIS, P., N. MAMOULIS und S. BAKIRAS: *On discovering moving clusters in spatio-temporal data*. In: *International Symposium on Spatial and Temporal Databases*, S. 364–381. Springer, 2005.
- [101] KALOGIANNI, E., R. SILERYTE, M. LAM, K. ZHOU, M. VAN DER HAM, S. VAN DER SPEK und E. VERBREE: *Passive wifi monitoring of the rhythm of the campus*. Agile, 2015.

- [102] KANTARWORLD PANEL: *Android growth: highest in EU5 in over two years*. <https://www.kantarworldpanel.com/global/News/Android-Share-Growth-is-Highest-in-EU5-in-Over-Two-Years>, 2016. letzter Abruf: 02.11.2017.
- [103] KESSEL, M.: *Bereitstellung von Umgebungsinformationen und Positionsdaten für ortsbezogene Dienste in Gebäuden*. Doktorarbeit, LMU, 2013.
- [104] KESSEL, M. und M. WERNER: *SMARTPOS: Accurate and precise indoor positioning on mobile phones*. In: *Proceedings of the First International Conference on Mobile Services, Resources, and Users, MOBILITY*, S. 158–163, 2011.
- [105] KHAN, M., N. SALMAN, A. ALI, A. KHAN und A. KEMP: *A comparative study of target tracking with Kalman filter, extended Kalman filter and particle filter using received signal strength measurements*. In: *Emerging Technologies (ICET), 2015 International Conference on*. IEEE, 2015.
- [106] KING, T. und M. B. KJÆRGAARD: *Composcan: adaptive scanning for efficient concurrent communications and positioning with 802.11*. In: *Proceedings of the 6th international conference on Mobile systems, applications, and services*, S. 67–80. ACM, 2008.
- [107] KIVIMÄKI, T., T. VUORELA, P. PELTOLA und J. VANHALA: *A review on device-free passive indoor positioning methods*. *International Journal of Smart Home*, 8(1):71–94, 2014.
- [108] KNOOP, S., S. VACEK und R. DILLMANN: *Sensor fusion for 3D human body tracking with an articulated 3D body model*. In: *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, S. 1686–1691. IEEE, 2006.
- [109] KOTSIANTIS, S. B., I. ZAHARAKIS und P. PINTELAS: *Supervised machine learning: A review of classification techniques*, 2007.
- [110] KOWEERAWONG, C., K. WIPUSITWARAKUN und K. KAEMARUNGSI: *Indoor localization improvement via adaptive RSS fingerprinting database*. In: *Information Networking (ICOIN), 2013 International Conference on*, S. 412–416. IEEE, 2013.
- [111] KOYUNCU, H. und S. H. YANG: *A survey of indoor positioning and object locating systems*. *IJCSNS International Journal of Computer Science and Network Security*, 10(5):121–128, 2010.
- [112] KRAUSE, A., R. RAJAGOPAL, A. GUPTA und C. GUESTRIN: *Simultaneous optimization of sensor placements and balanced schedules*. *IEEE Transactions on Automatic Control*, 56(10):2390–2405, 2011.

- [113] KRAUSE, A., A. SINGH und C. GUESTRIN: *Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies*. Journal of Machine Learning Research, 9(Feb):235–284, 2008.
- [114] KROLL, A.: *Computational Intelligence: Eine Einführung in Probleme, Methoden und technische Anwendungen*. Walter de Gruyter, 2013.
- [115] KROPEIT, T.: *Don't Trust Open Hotspots: Wi-Fi Hacker Detection and Privacy Protection via Smartphone*. Doktorarbeit, BS Thesis, Mar. 2015.[Online]. Available: [https://www.emsec.rub.de/media/attachments/files/2015/03/BA\\_Kropeit.pdf](https://www.emsec.rub.de/media/attachments/files/2015/03/BA_Kropeit.pdf), 2015.
- [116] KRUMM, J. und E. HORVITZ: *LOCADIO: Inferring Motion and Location from Wi-Fi Signal Strengths*. In: *mobile ubiquitous*, S. 4–13, 2004.
- [117] KWAPISZ, J. R., G. M. WEISS und S. A. MOORE: *Activity recognition using cell phone accelerometers*. ACM SigKDD Explorations Newsletter, 12(2):74–82, 2011.
- [118] LARSEN, J. E., P. SAPIEZYNSKI, A. STOPCZYNSKI, M. MØRUP und R. THEODORSEN: *Crowds, bluetooth, and rock'n'roll: understanding music festival participant behavior*. In: *Proceedings of the 1st ACM international workshop on Personal data meets distributed multimedia*, S. 11–18. ACM, 2013.
- [119] LAUBE, P., M. VAN KREVELD und S. IMFELD: *Finding REMO—detecting relative motion patterns in geospatial lifelines*. In: *Developments in spatial data handling*, S. 201–215. Springer, 2005.
- [120] LEE, D. L. und Q. CHEN: *A model-based wifi localization method*. In: *Proceedings of the 2nd international conference on Scalable information systems*, S. 40. ICST (Institute for Computer Sciences, Social Informatics and Telecommunications Engineering), 2007.
- [121] LEE, J.-G., J. HAN und K.-Y. WHANG: *Trajectory clustering: a partition-and-group framework*. In: *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, S. 593–604. ACM, 2007.
- [122] LEE, S., M. KIM, S. KANG, K. LEE und I. JUNG: *Smart scanning for mobile devices in WLANs*. In: *Communications (ICC), 2012 IEEE International Conference on*, S. 4960–4964. IEEE, 2012.
- [123] LEPPÄKOSKI, H., J. COLLIN und J. TAKALA: *Pedestrian navigation based on inertial sensors, indoor map, and WLAN signals*. Journal of Signal Processing Systems, 71(3):287–296, 2013.

- [124] LEUNE, T., C. KOCH und G. VON CÖLLN: *Anchor node placement in complex environments with physical raytracing and genetic algorithms*. In: *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on*, S. 1–6. IEEE, 2016.
- [125] LI, B., Y. WANG, H. K. LEE, A. DEMPSTER und C. RIZOS: *Method for yielding a database of location fingerprints in WLAN*. IEE Proceedings-Communications, 152(5):580–586, 2005.
- [126] LI, X., K. PAHLAVAN, M. LATVA-AHO und M. YLIANTTILA: *Comparison of indoor geolocation methods in DSSS and OFDM wireless LAN systems*. In: *Vehicular Technology Conference, 2000. IEEE-VTS Fall VTC 2000. 52nd*, Bd. 6, S. 3015–3020. IEEE, 2000.
- [127] LIM, R.: *Tracking smartphones using low-power sensor nodes*. In: *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, S. 52. ACM, 2013.
- [128] LINDQVIST, J., T. AURA, G. DANEZIS, T. KOPONEN, A. MYLLYNIEMI, J. MÄKI und M. ROE: *Privacy-preserving 802.11 access-point discovery*. In: *Proceedings of the second ACM conference on Wireless network security*, S. 123–130. ACM, 2009.
- [129] LIU, C., D. FANG, Z. YANG, H. JIANG, X. CHEN, W. WANG, T. XING und L. CAI: *RSS distribution-based passive localization and its application in sensor networks*. IEEE Transactions on Wireless Communications, 15(4):2883–2895, 2016.
- [130] LIU, H., H. DARABI, P. BANERJEE und J. LIU: *Survey of wireless indoor positioning techniques and systems*. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 37(6):1067–1080, 2007.
- [131] LOCAFOX: *Shopping-Apps: So schaffen Händler einen Mehrwert*. <https://www.locafox.de/blog/shopping-apps-so-schaffen-haendler-mit-apps-einen-mehrwert/>, 2015. letzter Abruf: 02.11.2017.
- [132] MAIER, M. und F. DORFMEISTER: *Fine-grained activity recognition of pedestrians travelling by subway*. In: *International Conference on Mobile Computing, Applications, and Services*, S. 122–139. Springer, 2013.
- [133] MAIER, M., L. SCHAUER und F. DORFMEISTER: *ProbeTags: Privacy-preserving proximity detection using Wi-Fi management frames*. In: *Wireless and Mobile Computing, Networking and Communications (Wi-Mob), 2015 IEEE 11th International Conference on*, S. 756–763. IEEE, 2015.

- 
- [134] MARANA, A., S. A. VELASTIN, L. D. F. COSTA und R. LOTUFO: *Automatic estimation of crowd density using texture*. Safety Science, 28(3):165–175, 1998.
- [135] MARCUS, P.: *Erfassung und Behandlung von Positionsfehlern in standortbasierter Autorisierung*. Doktorarbeit, LMU, 2015.
- [136] MARTIN, J., T. MAYBERRY, C. DONAHUE, L. FOPPE, L. BROWN, C. RIGGINS, E. C. RYE und D. BROWN: *A Study of MAC Address Randomization in Mobile Devices and When it Fails*. arXiv preprint arXiv:1703.02874, 2017.
- [137] MAUTZ, R.: *Indoor positioning technologies*, 2012.
- [138] MAUTZ, R. und S. TILCH: *Survey of optical indoor positioning systems*. In: *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, S. 1–7. IEEE, 2011.
- [139] MILLONIG, A., N. BRÄNDLE, M. RAY, D. BAUER und S. VAN DER SPEK: *Pedestrian Behaviour Monitoring: Methods and Experiences..* In: *BMI Book*, S. 11–42, 2009.
- [140] MOBILEWORLDLIVE: *Number of devices to hit 4.3 per person by 2020*. <https://www.mobileworldlive.com/featured-content/home-banner/connected-devices-to-hit-4-3-per-person-by-2020-report/>, 2014. letzter Abruf: 02.11.2017.
- [141] MOBUS, R. und U. KOLBE: *Multi-target multi-object tracking, sensor fusion of radar and infrared*. In: *Intelligent Vehicles Symposium, 2004 IEEE*, S. 732–737. IEEE, 2004.
- [142] MORENO, B., V. C. TIMES, C. RENSO und V. BOGORNY: *Looking Inside the Stops of Trajectories of Moving Objects..* In: *Geoinfo*, S. 9–20, 2010.
- [143] MUN, M., D. ESTRIN, J. BURKE und M. HANSEN: *Parsimonious mobility classification using GSM and WiFi traces*. In: *Proceedings of the Fifth Workshop on Embedded Networked Sensors (HotEmNets)*, 2008.
- [144] MUNDER, S. und D. M. GAVRILA: *An experimental study on pedestrian classification*. IEEE transactions on pattern analysis and machine intelligence, 28(11):1863–1868, 2006.
- [145] MUSA, A. und J. ERIKSSON: *Tracking unmodified smartphones using wi-fi monitors*. In: *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, S. 281–294. ACM, 2012.

- [146] MUSICKI, D. und W. KOCH: *Geolocation using TDOA and FDOA measurements*. In: *Information Fusion, 2008 11th International Conference on*, S. 1–8. IEEE, 2008.
- [147] MUTHUKRISHNAN, K., B. J. VAN DER ZWAAG und P. HAVINGA: *Inferring motion and location using WLAN RSSI*. In: *Mobile Entity Localization and Tracking in GPS-less Environments*, S. 163–182. Springer, 2009.
- [148] NAM, H. und B. HAN: *Learning multi-domain convolutional neural networks for visual tracking*. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, S. 4293–4302, 2016.
- [149] NAVARRO-SERMENT, L. E., J. M. DOLAN und P. K. KHOSLA: *Optimal sensor placement for cooperative distributed vision*. In: *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, Bd. 1, S. 939–944. IEEE, 2004.
- [150] NI, L. M., Y. LIU, Y. C. LAU und A. P. PATIL: *LANDMARC: indoor location sensing using active RFID*. *Wireless networks*, 10(6):701–710, 2004.
- [151] NIELS, R., L. VUURPIJL et al.: *Using Dynamic Time Warping for intuitive handwriting recognition*. In: *Proc. IGS*, S. 217–221. Citeseer, 2005.
- [152] NISHIDE, R. und H. TAKADA: *Exploring efficient methods to extract pedestrian flows on a mobile adhoc network*. In: *UBICOMM 2012, The 6th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, S. 29–34, 2012.
- [153] NURMINEN, H., A. RISTIMAKI, S. ALI-LOYTTY und R. PICHÉ: *Particle filter and smoother for indoor localization*. In: *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*, S. 1–10. IEEE, 2013.
- [154] NURMINEN, H., J. TALVITIE, S. ALI-LOYTTY, P. MULLER, E. LOHAN, R. PICHÉ und M. RENFORS: *Statistical path loss parameter estimation and positioning using RSS measurements in indoor wireless networks*. In: *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, S. 1–9. IEEE, 2012.
- [155] OREIFEJ, O., R. MEHRAN und M. SHAH: *Human identity recognition in aerial images*. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, S. 709–716. IEEE, 2010.
- [156] ORR, R. J. und G. D. ABOWD: *The smart floor: A mechanism for natural user identification and tracking*. In: *CHI'00 extended abstracts on Human factors in computing systems*, S. 275–276. ACM, 2000.

- [157] OUCHI, K. und M. DOI: *Indoor-outdoor activity recognition by a smartphone*. In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, S. 600–601. ACM, 2012.
- [158] PADILLA-LÓPEZ, J. R., A. A. CHAARAOUI und F. FLÓREZ-REVUELTA: *Visual privacy protection methods: A survey*. *Expert Systems with Applications*, 42(9):4177–4195, 2015.
- [159] PAHLAVAN, K., X. LI, M. YLIANTTILA, R. CHANA und M. LATVA-AHO: *An overview of wireless indoor geolocation techniques and systems*. In: *Mobile and Wireless Communications Networks*, S. 1–13. Springer, 2000.
- [160] PALMA, A. T., V. BOGORNY, B. KUIJPERS und L. O. ALVARES: *A clustering-based approach for discovering interesting places in trajectories*. In: *Proceedings of the 2008 ACM symposium on Applied computing*, S. 863–868. ACM, 2008.
- [161] PANG, J., B. GREENSTEIN, R. GUMMADI, S. SESHAN und D. WETHERALL: *802.11 user fingerprinting*. In: *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, S. 99–110. ACM, 2007.
- [162] PARLAK, S., I. MARSIC, A. SARCEVIC, W. U. BAJWA, L. J. WATERHOUSE und R. S. BURD: *Passive RFID for Object and Use Detection During Trauma Resuscitation*. *IEEE Transactions on Mobile Computing*, 15(4):924–937, 2016.
- [163] PEDREGOSA, F., G. VAROQUAUX, A. GRAMFORT, V. MICHEL, B. THIRION, O. GRISEL, M. BLONDEL, P. PRETTENHOFER, R. WEISS, V. DUBOURG, J. VANDERPLAS, A. PASSOS, D. COURNAPEAU, M. BRUCHER, M. PERROT und E. DUCHESNAY: *Scikit-learn: Machine Learning in Python*. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [164] PRENTOW, T. S., A. J. RUIZ-RUIZ, H. BLUNCK, A. STISEN und M. B. KJÆRGAARD: *Spatio-temporal facility utilization analysis from exhaustive wifi monitoring*. *Pervasive and Mobile Computing*, 16:305–316, 2015.
- [165] QIN, W., J. ZHANG, B. LI und L. SUN: *Discovering human presence activities with smartphones using nonintrusive wi-fi sniffer sensors: the big data prospective*. *International Journal of Distributed Sensor Networks*, 9(12):927940, 2013.
- [166] RABINER, L. R.: *A tutorial on hidden Markov models and selected applications in speech recognition*. *Proceedings of the IEEE*, 77(2):257–286, 1989.

- [167] RAHMALAN, H., M. S. NIXON und J. N. CARTER: *On crowd density estimation for surveillance*. In: *Crime and Security, 2006. The Institution of Engineering and Technology Conference on*, S. 540–545. IET, 2006.
- [168] RAJAGOPAL, N., S. CHAYAPATHY, B. SINOPOLI und A. ROWE: *Beacon placement for range-based indoor localization*. In: *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on*, S. 1–8. IEEE, 2016.
- [169] RAO, A. S., J. GUBBI, S. MARUSIC, P. STANLEY und M. PALANISWAMI: *Crowd density estimation based on optical flow and hierarchical clustering*. In: *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on*, S. 494–499. IEEE, 2013.
- [170] RATANAMAHATANA, C. A., J. LIN, D. GUNOPULOS, E. KEOGH, M. VLACHOS und G. DAS: *Mining time series data*. In: *Data mining and knowledge discovery handbook*, S. 1049–1077. Springer, 2009.
- [171] RUIZ-RUIZ, A. J., H. BLUNCK, T. S. PRENTOW, A. STISEN und M. B. KJÆRGAARD: *Analysis methods for extracting knowledge from large-scale WiFi monitoring to inform building facility planning*. In: *Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on*, S. 130–138. IEEE, 2014.
- [172] SANCHEZ, I., Z. M. M. AYE, B. I. RUBINSTEIN und K. RAMAMOHANARAO: *Fast trajectory clustering using Hashing methods*. In: *Neural Networks (IJCNN), 2016 International Joint Conference on*, S. 3689–3696. IEEE, 2016.
- [173] SASSI, A., C. BOREAN, R. GIANNANTONIO, M. MAMEI, D. MANA und F. ZAMBONELLI: *Crowd steering in public spaces: Approaches and strategies*. In: *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PI-COM), 2015 IEEE International Conference on*, S. 2098–2105. IEEE, 2015.
- [174] SAVITZKY, A. und M. J. GOLAY: *Smoothing and differentiation of data by simplified least squares procedures..* Analytical chemistry, 36(8):1627–1639, 1964.
- [175] SCHAUER, L.: *Discovering hotspots: A placement strategy for Wi-Fi based trajectory monitoring within buildings*. In: *SAI Intelligent Systems Conference (IntelliSys), 2015*, S. 371–380. IEEE, 2015.

- 
- [176] SCHAUER, L.: *Analyzing the Digital Society by Tracking Mobile Customer Devices*. In: *Digital Marketplaces Unleashed*, S. 467–478. Springer, 2018.
- [177] SCHAUER, L., F. DORFMEISTER und M. MAIER: *Potentials and limitations of wifi-positioning using time-of-flight*. In: *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*, S. 1–9. IEEE, 2013.
- [178] SCHAUER, L., F. DORFMEISTER und F. WIRTH: *Analyzing passive Wi-Fi fingerprinting for privacy-preserving indoor-positioning*. In: *Localization and GNSS (ICL-GNSS), 2016 International Conference on*, S. 1–6. IEEE, 2016.
- [179] SCHAUER, L., A. HIRSCHLEB und M. ZADDACH: *Der Flughafen als Smart City*. *Informatik-Spektrum*, 39(2):115–121, 2016.
- [180] SCHAUER, L. und C. LINNHOFF-POPIEN: *Extracting Context Information from Wi-Fi Captures*. In: *Proceedings of the 10th International Conference on Pervasive Technologies Related to Assistive Environments*, S. 123–130. ACM, 2017.
- [181] SCHAUER, L., P. MARCUS und C. LINNHOFF-POPIEN: *Towards feasible Wi-Fi based indoor tracking systems using probabilistic methods*. In: *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on*, S. 1–8. IEEE, 2016.
- [182] SCHAUER, L. und M. WERNER: *Analyzing pedestrian flows based on wi-fi and bluetooth captures*. In: *EAI Endorsed Transactions on Ubiquitous Environments*, Bd. 1 d. Reihe 4, S. 1–10. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2015.
- [183] SCHAUER, L. und M. WERNER: *Clustering of Inertial Indoor Positioning Data*. In: *1st KuVS Expert Talk on Localization*, S. 21–23. Aachener Informatik-Berichte, 2015.
- [184] SCHAUER, L. und M. WERNER: *Kunden-Monitoring im stationären Handel*. In: *Marktplätze im Umbruch*, S. 73–82. Springer, 2015.
- [185] SCHAUER, L., M. WERNER und P. MARCUS: *Estimating crowd densities and pedestrian flows using wi-fi and bluetooth*. In: *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, S. 171–177. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014.

- [186] SCHILLING, F.: *Tracking Smartphones using 802.11 Management Frames*. Bachelorarbeit, LMU München, 2014.
- [187] SEIFELDIN, M., A. SAEED, A. E. KOSBA, A. EL-KEYI und M. YOUSSEF: *Nuzzer: A large-scale device-free passive localization system for wireless environments*. *IEEE Transactions on Mobile Computing*, 12(7):1321–1334, 2013.
- [188] SHEN, J., J. CAO, X. LIU, J. WEN und Y. CHEN: *Feature-Based Room-Level Localization of Unmodified Smartphones*. In: *Smart City 360*, S. 125–136. Springer, 2016.
- [189] SHERMER, T. C.: *Recent results in art galleries (geometry)*. *Proceedings of the IEEE*, 80(9):1384–1399, 1992.
- [190] SHETH, A., S. NEDEVSCHI, R. PATRA, S. SURANA, E. BREWER und L. SUBRAMANIAN: *Packet loss characterization in WiFi-based long distance networks*. In: *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, S. 312–320. IEEE, 2007.
- [191] SHOAI B, M., S. BOSCH, O. D. INCEL, H. SCHOLTEN und P. J. HAVINGA: *A survey of online activity recognition using mobile phones*. *Sensors*, 15(1):2059–2085, 2015.
- [192] SHOAI B, M., H. SCHOLTEN und P. J. HAVINGA: *Towards physical activity recognition using smartphone sensors*. In: *Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC)*, S. 80–87. IEEE, 2013.
- [193] STATISTA: *Anzahl der Nutzer von Smartphones in Deutschland bis 2016*. <https://de.statista.com/statistik/daten/studie/198959/umfrage/anzahl-der-smartphonennutzer-in-deutschland-seit-2010/>, 2017. letzter Abruf: 02.11.2017.
- [194] STATISTA: *Smartphone users worldwide 2014-2020*. <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>, 2017. letzter Abruf: 02.11.2017.
- [195] SUH, Y.-H., S. K. RHEE und K.-W. LEE: *Continuous location tracking of people by multiple depth cameras*. In: *Information and Communication Technology Convergence (ICTC), 2015 International Conference on*, S. 170–172. IEEE, 2015.
- [196] SUI, Y., Z. ZHANG, G. WANG, Y. TANG und L. ZHANG: *Real-Time Visual Tracking: Promoting the Robustness of Correlation Filter Learning*. In: *European Conference on Computer Vision*, S. 662–678. Springer, 2016.

- 
- [197] THIAGARAJAN, A., L. RAVINDRANATH, K. LACURTS, S. MADDEN, H. BALAKRISHNAN, S. TOLEDO und J. ERIKSSON: *VTrack: accurate, energy-aware road traffic delay estimation using mobile phones*. In: *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, S. 85–98. ACM, 2009.
- [198] THRUN, S., W. BURGARD und D. FOX: *Probabilistic Robotics*. The MIT Pres, 2006.
- [199] TIWARI, S. und S. KAUSHIK: *Mining popular places in a geo-spatial region based on GPS data using semantic information*. In: *International Workshop on Databases in Networked Information Systems*, S. 262–276. Springer, 2013.
- [200] TOLL, W. G. VAN, A. F. COOK IV, R. GERAERTS et al.: *Realistic crowd simulation with density-based path planning*. Advanced School for Computing and Imaging (ASCI)-ICT. OPEN, S. 3–9, 2012.
- [201] TROGH, J., D. PLETS, L. MARTENS und W. JOSEPH: *Advanced real-time indoor tracking based on the viterbi algorithm and semantic data*. International Journal of Distributed Sensor Networks, 2015.
- [202] TRUCCO, E. und K. PLAKAS: *Video tracking: a concise survey*. IEEE Journal of Oceanic Engineering, 31(2):520–529, 2006.
- [203] VANHOEF, M., C. MATTE, M. CUNCHE, L. S. CARDOSO und F. PIENSSENS: *Why MAC Address Randomization is not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms*. In: *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, S. 413–424. ACM, 2016.
- [204] VELAYOS, H. und G. KARLSSON: *Limitations of range estimation in wireless LAN*. In: *Proc. 1st Workshop on Positioning, Navigation and Communication (WPNC'04), Hannover, Germany, 2004*.
- [205] VERSICHELE, M., T. NEUTENS, M. DELAFONTAINE und N. VAN DE WEGHE: *The use of Bluetooth for analysing spatiotemporal dynamics of human movement at mass events: A case study of the Ghent Festivities*. Applied Geography, 32(2):208–220, 2012.
- [206] VITERBI, A.: *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm*. IEEE transactions on Information Theory, 13(2):260–269, 1967.
- [207] VLACHOS, M., D. GUNOPULOS und G. KOLLIOS: *Robust similarity measures for mobile object trajectories*. In: *Database and Expert Systems Applications, 2002. Proceedings. 13th International Workshop on*, S. 721–726. IEEE, 2002.

- [208] WANG, H., H. LENZ, A. SZABO, J. BAMBERGER und U. D. HANE-BECK: *WLAN-based pedestrian tracking using particle filters and low-cost MEMS sensors*. In: *Positioning, Navigation and Communication, 2007. WPNC'07. 4th Workshop on*, S. 1–7. IEEE, 2007.
- [209] WANG, Y., J. YANG, H. LIU, Y. CHEN, M. GRUTESER und R. P. MARTIN: *Measuring human queues using WiFi signals*. In: *Proceedings of the 19th annual international conference on Mobile computing & networking*, S. 235–238. ACM, 2013.
- [210] WANT, R., A. HOPPER, V. FALCAO und J. GIBBONS: *The active badge location system*. *ACM Transactions on Information Systems (TOIS)*, 10(1):91–102, 1992.
- [211] WARD, A., A. JONES und A. HOPPER: *A new location technique for the active office*. *IEEE Personal communications*, 4(5):42–47, 1997.
- [212] WEISER, M.: *The computer for the 21st century*. *Scientific american*, 265(3):94–104, 1991.
- [213] WEISER, M.: *Ubiquitous computing*. *Computer*, 26(10):71–72, 1993.
- [214] WEPPNER, J., B. BISCHKE und P. LUKOWICZ: *Monitoring crowd condition in public spaces by tracking mobile consumer devices with wifi interface*. In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, S. 1363–1371. ACM, 2016.
- [215] WEPPNER, J. und P. LUKOWICZ: *Bluetooth based collaborative crowd density estimation with mobile phones*. In: *Pervasive Computing and Communications (PerCom), 2013 IEEE International Conference on*, S. 193–200. IEEE, 2013.
- [216] WERNER, M.: *Efficiently using bitmap floorplans for indoor navigation on mobile phones*. In: *The Seventh International Conference on Wireless and Mobile Communications (ICWMC)*, S. 225–230, 2011.
- [217] WERNER, M.: *Ubiquitous Navigation: skalierbare ortsbezogene Dienste in Gebäuden..* Doktorarbeit, Universitätsbibliothek der Ludwig-Maximilians-Universität, 2012.
- [218] WERNER, M. und S. FELD: *Homotopy and alternative routes in indoor navigation scenarios*. In: *Indoor Positioning and Indoor Navigation (IPIN), 2014 International Conference on*, S. 230–238. IEEE, 2014.
- [219] WERNER, M., C. HAHN und L. SCHAUER: *DeepMoVIPS: Visual indoor positioning using transfer learning*. In: *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on*, S. 1–7. IEEE, 2016.

- [220] WERNER, M., L. SCHAUER und A. SCHARF: *Reliable trajectory classification using Wi-Fi signal strength in indoor scenarios*. In: *2014 IEEE/ION Position, Location and Navigation Symposium-PLANS 2014*, S. 663–670. IEEE, 2014.
- [221] WIDYAWAN, M. KLEPAL und S. BEAUREGARD: *A Novel Backtracking Particle Filter for Pattern Matching Indoor Localization*. In: *Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments, MELT '08*, S. 79–84, New York, NY, USA, 2008. ACM.
- [222] WIKIPEDIA: *OPTICS*. <https://de.wikipedia.org/wiki/OPTICS>, 2015. letzter Abruf: 02.11.2017.
- [223] WIKIPEDIA: *Google Now*. [https://en.wikipedia.org/wiki/Google\\_Now](https://en.wikipedia.org/wiki/Google_Now), 2016. letzter Abruf: 02.11.2017.
- [224] WIKIPEDIA: *Bluetooth*. <https://de.wikipedia.org/wiki/Bluetooth>, 2017. letzter Abruf: 02.11.2017.
- [225] WIKIPEDIA: *Bluetooth*. <https://en.wikipedia.org/wiki/Bluetooth>, 2017. letzter Abruf: 02.11.2017.
- [226] WIKIPEDIA: *Cortana (Software)*. [https://de.wikipedia.org/wiki/Cortana\\_\(Software\)](https://de.wikipedia.org/wiki/Cortana_(Software)), 2017. letzter Abruf: 02.11.2017.
- [227] WIKIPEDIA: *Siri (Software)*. [https://de.wikipedia.org/wiki/Siri\\_\(Software\)](https://de.wikipedia.org/wiki/Siri_(Software)), 2017. letzter Abruf: 02.11.2017.
- [228] WIKIPEDIA: *Viterbi-Algorithmus*. <https://de.wikipedia.org/wiki/Viterbi-Algorithmus>, 2017. letzter Abruf: 02.11.2017.
- [229] WIKIPEDIA: *Wireless Local Area Network*. [https://de.wikipedia.org/wiki/Wireless\\_Local\\_Area\\_Network](https://de.wikipedia.org/wiki/Wireless_Local_Area_Network), 2017. letzter Abruf: 02.11.2017.
- [230] WIND, D. K., P. SAPIEZYNSKI, M. A. FURMAN und S. LEHMANN: *Inferring stop-locations from wifi*. *PloS one*, 11(2):e0149105, 2016.
- [231] WIRTSCHAFTSWOCHE: *Digitaler Wandel - Hochintelligente Smartphones auf Rädern*. <http://www.wiwo.de/unternehmen/auto/digitaler-wandel-hochintelligente-smartphones-auf-raedern/9793702-3.html>, 2017. letzter Abruf: 02.11.2017.
- [232] XU, Z., K. SANDRASEGARAN, X. KONG, X. ZHU, J. ZHAO, B. HU und C.-C. LIN: *Pedestrian Monitoring System using Wi-Fi Technology And RSSI Based Localization..* *International Journal of Wireless & Mobile Networks*, 5(4), 2013.

- [233] YAMASAKI, R., A. OGINO, T. TAMAKI, T. UTA, N. MATSUZAWA und T. KATO: *TDOA location system for IEEE 802.11 b WLAN*. In: *Wireless Communications and Networking Conference, 2005 IEEE*, Bd. 4, S. 2338–2343. IEEE, 2005.
- [234] YANG, J. und Y. CHEN: *Indoor localization using improved rss-based lateration methods*. In: *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, S. 1–6. IEEE, 2009.
- [235] YIN, J. H., S. A. VELASTIN und A. C. DAVIES: *Image processing techniques for crowd density estimation using a reference image*. In: *Recent Developments in Computer Vision*, S. 489–498. Springer, 1996.
- [236] YOON, H. und C. SHAHABI: *Robust time-referenced segmentation of moving object trajectories*. In: *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, S. 1121–1126. IEEE, 2008.
- [237] YOUNIS, M. und K. AKKAYA: *Strategies and techniques for node placement in wireless sensor networks: A survey*. *Ad Hoc Networks*, 6(4):621–655, 2008.
- [238] YOUSSEF, M. und A. AGRAWALA: *The Horus WLAN location determination system*. In: *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, S. 205–218. ACM, 2005.
- [239] YÜRÜR, Ö., C. H. LIU, Z. SHENG, V. C. LEUNG, W. MORENO und K. K. LEUNG: *Context-awareness for mobile sensing: A survey and future directions*. *IEEE Communications Surveys & Tutorials*, 18(1):68–93, 2016.
- [240] ZHANG, Y., M. G. AMIN und S. KAUSHIK: *Localization and tracking of passive RFID tags based on direction estimation*. *International Journal of Antennas and Propagation*, 2007, 2007.
- [241] ZHANG, Z.: *Microsoft kinect sensor and its effect*. *IEEE multimedia*, 19(2):4–10, 2012.
- [242] ZHENG, Y. und M. HASEGAWA-JOHNSON: *Formant tracking by mixture state particle filter*. In: *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP'04). IEEE International Conference on*, Bd. 1, S. I–565. IEEE, 2004.
- [243] ZHENG, Y., Q. LI, Y. CHEN, X. XIE und W.-Y. MA: *Understanding mobility based on GPS data*. In: *Proceedings of the 10th international conference on Ubiquitous computing*, S. 312–321. ACM, 2008.
- [244] ZHENG, Y. und X. ZHOU: *Computing with spatial trajectories*. Springer Science & Business Media, 2011.