

**Fusionsalgorithmus zur autonomen Positionsschätzung
im Gebäude, basierend auf MEMS-Inertialsensoren
im Smartphone**

Dissertation

zur Vorlage beim Promotionsausschuss

zur Erlangung des Grades Dr.-Ing.

von

Thomas Willemsen

Geomatik | HafenCity Universität Hamburg

Tag der mündlichen Prüfung: 22.06.2016

Erstgutachter: Prof. Dr.-Ing. Harald Sternberg (HCU Hamburg)

Zweitgutachter: Prof. Dr.-Ing. Jörg Blankenbach (RWTH Aachen)

Vorwort

Die vorliegende Arbeit ist während meiner Tätigkeit als Wissenschaftlicher Mitarbeiter an der HafenCity Universität in den Jahren 2011 bis 2016 entstanden. Fachlich dem Studiengang Geomatik zugeordnet, behandelt diese Arbeit eine Möglichkeit einer weitestgehend autonom funktionierenden Positionsschätzung auf dem Smartphone für die Fußgänger Indoor-Navigation. Sie wurde als Dissertation dem Promotionsausschuss vorgelegt und angenommen.

Für die Unterstützung während der Bearbeitungszeit danke ich vor allem und zuallererst meiner Frau Tanja, die in den letzten Jahren mein stärkster Rückhalt war.

Meinen Eltern danke ich für die jahrelange Unterstützung während meiner Ausbildungszeit, die dadurch die Voraussetzung für eine intensive Fokussierung auf die fachlichen Schwerpunkte schafften.

Vielen Dank auch meinen Gutachtern, Prof. Dr.-Ing. Harald Sternberg und Prof. Dr.-Ing. Jörg Blankenbach, für die konstruktiven Gespräche und die Begutachtung dieser Arbeit.

Ein besonderer Dank gilt weiterhin meinem geschätzten Kollegen Dr.-Ing. Friedrich Keller. Die anregenden Diskussionen während der Bearbeitungszeit, halfen viele Herausforderungen zu bewältigen.

Abschließend bedanke ich mich bei allen meinen Freunden und Familienmitgliedern für die Rücksicht und das entgegengebrachte Verständnis, aufgrund meiner regelmäßigen Abwesenheiten in den letzten Jahren.

Hamburg, den 1. August 2016

Thomas Willemsen

Kurzfassung

Eine infrastrukturbasierte Indoor-Navigation in GNSS abgeschatteten Bereichen bringt üblicherweise einen hohen Implementierungs-, Wartungs- und Kostenaufwand mit sich. Die Entwicklung einer Alternative, die praktikabel, kostengünstig und leicht verfügbar ist, ist daher wünschenswert. Um diesen Anforderungen gerecht zu werden, wird in der vorliegenden Arbeit am Beispiel der Testgeräte Samsung Galaxy Nexus sowie Google Nexus 4 die Tauglichkeit moderner Smartphones für eine weitgehend autonome Positionsschätzung innerhalb von Gebäuden untersucht. Dazu wird eine rein auf MEMS-INS-basierte Positionsschätzung umgesetzt, die eine Übertragbarkeit auf andere Gebäude erlaubt und Inzellösungen für nur ein Gebäude vermeidet. Analysiert werden dazu drei Algorithmen auf Basis von Kalman Filter, Partikelfilter und Zustandsabfragen auf einem Routing-Graph im Hinblick auf die Verarbeitung stochastisch unterschiedlicher Messdaten, aus denen sich ein Fusionsalgorithmus für eine auf Inertialsensoren basierte Positionsschätzung entwickeln lässt. Sensoren, die dazu von Interesse sein können, sind Beschleunigungsmesser, Magnetfeldsensor, Gyroskop und Barometer. Sie werden hinsichtlich der Messeigenschaften untersucht und, sofern erforderlich, für Navigationszwecke kalibriert. Zudem wird eine Abgrenzung der verwendeten Zusatzinformationen aus der gegebenen Infrastruktur, wie beispielsweise Kartenmaterial, Routing-Graph, WLAN Netzwerk etc., vorgenommen. Bei der praktischen Umsetzung in einem Testgebäude wird eine Implementierung als Navigationsapplikation mit Routing und Kartengrundlage diskutiert. Für das Routing werden zwei Algorithmen untersucht sowie Ansätze für die automatische Erstellung von Karten- und Routing-Daten vorgestellt. Insgesamt lässt sich festhalten, dass sich die untersuchten Testgeräte ohne gesonderte Kalibrierung einsetzen lassen. Kalman Filter, Partikelfilter und topologischer Ansatz arbeiten im Prinzip so zuverlässig, dass sie unter kontrollierten Testbedingungen gute Ergebnisse erzielen. Kalman Filter und Partikelfilter generieren durch Positionskorrekturen mittels Routing-Graphen in bestimmten Bereichen, etwa in Eingangshallen, Unsicherheiten. Der topologische Ansatz funktioniert lediglich auf einem Routing-Graph und verfälscht die Positionsschätzung, sobald der Nutzer vom dem definierten Weg abweicht. Der hier favorisierte Filter auf Basis von Partikelfilter und topologischem Ansatz minimiert

solche Schwächen und erlaubt die Positionsschätzung im gesamten Testgebäude. Anhand von 46 Datensätzen kann demonstriert werden, dass der Algorithmus in 70 Prozent der Fälle die Zielgenauigkeit der Positionsschätzung von <5 m einhält – je nach Handhabung des Smartphones während der Navigation. Somit ist in Abhängigkeit des Nutzerverhaltens eine Positionsschätzung im Gebäude generell ohne weitere infrastrukturbasierte Korrekturen möglich.

Abstract

An infrastructure-based indoor navigation in GNSS-shaded areas is usually accompanied with a relatively high implementation effort and considerable costs. Thus, an alternative method which is practicable, cost-effective and easily available would be of interest. In the present work a largely autonomous position estimate within buildings is developed. A Samsung Galaxy Nexus and a Google Nexus 4 are used as they represent modern smartphones. A purely MEMS inertial sensors based position estimate is realized which permits a fast transferability to other buildings and helps to avoid using indoor navigation as an isolated solution for an individual building. Three algorithms are therefore analyzed based on Kalman filter, particle filter and state detection on a routing-graph with regard to the processing of stochastically different measurement data. From these a final algorithm for a position estimate based on inertial sensors is developed. The sensors of interest are the accelerometer, the magnetic field sensor the gyroscope and the barometer. They are examined regarding their measuring characteristics and are calibrated for navigation purpose if necessary. Furthermore, a demarcation of the additional used information from the given infrastructure, as for example maps, routing-graph, WLAN network etc., is carried out. In the process of practical realization in a test building an implementation as navigation application based on routing-graph and map is discussed. Two algorithms are examined for the routing and different approaches for an automatic map and routing-graph data generation are presented. In summary it can be stated that the examined test devices can be used without calibration. Kalman filter, particle filter and the topological approach achieve good results under controlled test conditions in general. The Kalman and particle filter generate uncertainties by position corrections by routing-graph in certain areas as foyers for example. The topological approach works only on an existing routing-graph and falsifies the position estimate if the user is not walking on the defined paths. A fusion of particle filter and topological approach minimizes such weaknesses and permits the position estimate in the whole test building. The analysis of 46 data sets shows that the algorithm generates positions of the aimed accuracy in 70 percent of the cases. Depending on the handling of the navigation device it is therefore possible to generate a po-

sition estimate in buildings without additional infrastructure-based corrections in general.

Inhaltsverzeichnis

Vorwort	i
Kurzfassung	iii
Abstract	v
Inhaltsverzeichnis	vii
1 Einführung	1
1.1 Stand der Forschung	1
1.2 Konzept und Ziel	4
2 Grundlagen	9
2.1 Micro Electro Mechanical System (MEMS)	9
2.1.1 Technologie und typische Anwendungsgebiete	9
2.1.2 Beschleunigungsmesser	11
2.1.3 Gyroskop	13
2.1.4 Magnetfeldsensor	15
2.1.5 Barometer	16
2.2 Stochastische Filter	17
2.2.1 Kalman Filter	18
2.2.2 Partikelfilter	23
2.2.3 Vergleich von Kalman Filter und Partikelfilter	28
2.3 Gebäudepläne und Routing-Netz	29
2.4 Graphentheorie und Routingalgorithmen	33
2.5 Versuchshardware	36
2.5.1 Smartphones mit integrierten MEMS	37
2.5.2 Android als Entwicklungsumgebung	47
2.5.3 Arbeiten auf Sensorebene am Beispiel von Arduino	48
3 Untersuchungen und Kalibrierung von MEMS im Smartphone	54
3.1 Stockwerkserkennung mit Barometer BMP-180	54
3.2 Kalibrierung eines Gyroskops	60

3.3	Kalibrierung eines Beschleunigungsmessers	70
3.4	Kalibrierung eines Magnetfeldsensors	78
3.5	Einfluss der Nicht-Orthogonalität	85
3.6	Qualität des Zeitstempels im Nexus 4	90
4	Komponenten einer Smartphone gestützten hybriden Positionsschätzung	95
4.1	Trajektorie aus Smartphone-Sensoren	95
4.1.1	Pedestrian Dead Reckoning	96
4.1.2	Berechnung einer Trajektorie (2D + 1D)	105
4.2	Infrastrukturabhängige Positionsschätzung	111
4.2.1	Unterscheidung und Überblick der bekanntesten Verfahren	111
4.2.2	WLAN Fingerprinting am Beispiel des HCU-Gebäudes . . .	122
4.3	Möglichkeiten der Stützung einer INS-basierten Positionsschätzung	129
4.3.1	Weitere Eingrenzung von Stützungsmöglichkeiten	130
4.3.2	Favorisierte Komponenten	137
5	Gestützte MEMS-basierte Positionsschätzung	140
5.1	Kalman Filter	140
5.1.1	Aufbau des Filters	140
5.1.2	Ergebnisdiskussion zweier repräsentativer Routen	147
5.2	Partikelfilter	156
5.2.1	Aufbau des Filters	156
5.2.2	Ergebnisdiskussion zweier repräsentativer Routen	161
5.3	Topologischer Ansatz auf einem Routing-Graphen	164
5.3.1	Zustandsschätzung aus MEMS-Sensordaten	166
5.3.2	Ergebnisdiskussion zweier repräsentativer Routen	170
5.4	Vergleich und Diskussion der vorgestellten Ansätze	176
5.5	Finaler Fusionsalgorithmus - Kantenbasiertes Partikelfilter	185
5.5.1	Fusion von Partikelfilter und topologischem Ansatz	185
5.5.2	Ergebnisdiskussion zweier repräsentativer Routen	190
5.5.3	Untersuchungen zur Zuverlässigkeit	200

6 Implementierung für eine Navigationsanwendung	213
6.1 Weitere Komponenten	213
6.1.1 Routenberechnung	213
6.1.2 Raumattribute	216
6.1.3 Erstellung von Visualisierungs- und Routinggrundlage . . .	217
6.2 Voraussetzungen und Zusammenspiel der Komponenten	224
6.3 Anwendungen (Apps)	227
7 Zusammenfassung und Ausblick	233
Literatur	239
Abbildungsverzeichnis	x
Tabellenverzeichnis	xxii
Stichwortverzeichnis	xxiv

1 Einführung

1.1 Stand der Forschung

Die Nutzung von Smartphones zur Navigation ist heutzutage eine Selbstverständlichkeit. Sie basiert auf dem Empfang und der Verarbeitung von Signalen eines Globalen Navigationssatellitensystems (GNSS). In Gebäuden, in denen GNSS-Signale üblicherweise nicht empfangen werden können, versagt die GNSS basierte Positionsbestimmung allerdings. Dabei können auch unter Innenraumbedingungen standortbezogene Dienste von großem Nutzen sein. Anwendungen sind denkbar auf einem Messegelände, innerhalb von Einkaufszentren, auf Großbahnhöfen oder in modernen Bürokomplexen. Daher gewinnt das Forschungsfeld der Indoor-Navigation an Bedeutung und es wird mit Nachdruck an Technologien und Verfahren gearbeitet, die eine Positionsschätzung ohne GNSS-Signale ermöglichen.

Ein Überblick über mögliche Verfahren zur Positionsschätzung findet sich in Mautz [2012]. Dort wird eine Klassifizierung der Verfahren zur Positionsschätzung nach dem Kriterium der jeweils erreichbaren Genauigkeiten der Position vorgeschlagen. Während Systeme mit einer Genauigkeit kleiner als 10 cm für die Ortung im Bereich des Facility Managements erforderlich sind, sind Systeme mit einer Genauigkeit von 1 bis 5 m für die Personennavigation hinreichend. Eine andere Klassifizierung lässt sich im Hinblick auf die eingesetzten Technologien zur Positionsschätzung vornehmen. Hierbei werden infrastrukturgestützte Systeme, hybride Systeme und bildbasierte Methoden unterschieden.

Infrastrukturgestützte Systeme arbeiten drahtlos und bestimmen unter anderem Distanzen zu Referenzpunkten. Die hierbei eingesetzten Verfahren zur Distanzmessung sind entweder ultraschallbasiert [Mok u. a., 2012] oder nutzen ein breites Spektrum elektromagnetischer Wellen [Blankenbach u. a., 2011]. Bei den von Blankenbach u. a. [2011] vorgestellten Verfahren werden künstliche Magnetfelder durch Magnetspulen erzeugt. Die Distanzen werden aus den Signalen mittels Signalverarbeitung und Kreuzkorrelation abgeleitet. Mit der Kenntnis der Position der Spulen sowie einem Smartphone zum Empfang lässt sich eine Position mittels Trilateration schätzen. Im Nahbereich (4 m) können so Genauigkeiten von 0,25 m erzielt werden. Im von Mok u. a. [2012] dargelegtem Verfahren konnten

bei Feldtests in unterschiedlichen Räumlichkeiten ($< 40 \times 40 \text{ m}$) mit Ultra Wide Band-Technologie (UWB) 3D-Positionsgenauigkeiten (95 %) von 1 bis 5 m erreicht werden.

Andere infrastrukturbasierte Systeme, die kurzweiliges Laserlicht einsetzen, erreichen je nach eingesetztem Verfahren Genauigkeiten bis in den sub-Millimeter Bereich. Mit steigender Wellenlänge vom Infrarotbereich bis hin zu Positionsbestimmungsverfahren, die Funktechniken verwenden, sinkt die erreichbare Genauigkeit. Hier sind vor allem Techniken der Mobilfunkortung (zellbasierte Positionsbestimmung) und die Positionsbestimmung durch Drahtlosnetzwerke wie Wireless-LAN (WLAN) und Bluetooth zu nennen [Mautz, 2012]. Für alle diese Technologien gibt es dezidierte Lösungen, die aber nur mit großem Aufwand zu realisieren sind. Sie stellen hohe Anforderungen an die Implementierung und lassen sich oft nur für einzelne Räume, Gebäude oder Fabrikhallen umsetzen.

Eine ausführliche Dokumentation der bestehenden Methoden zur Positionsschätzung in GNSS-abgeschatteten Bereichen findet sich im Handbuch von Zekavat u. Buehrer [2012]. Am Beispiel der Anwendungsmöglichkeiten von WLAN Signalen machen die Autoren deutlich, wie vielfältig und differenziert inzwischen die Methoden zur Positionsschätzung sind. Sie stellen Zeitmessungen (time of arrival), relative Zeitmessungen (time difference of arrival), zellbasierende Positionsschätzungen (cell of origin), ein Dämpfungsmodell der Signalstärke zur Streckenbestimmung sowie Fingerprinting-Ansätze zur Verwendung von WLAN Infrastrukturen vor.

Willert u. a. [2011] und Händler [2012] beschreiben ein Verfahren zur Positionsschätzung im Gebäude mit Hilfe von Bildern. Zu diesem Zweck werden codierte Marken auf Türen angebracht sowie die Geometrie der Türen verwendet. Während einer Navigation werden mit einem mitgeführten Smartphone Bilder erzeugt und die Bilder, die Türen mit codierten Marken enthalten, zur Positionsschätzung herangezogen. Zur Positionsschätzung wird im Bild nach einem codierten Target gesucht, das die im Bild sichtbare Tür identifizieren lässt. Aus Kamerabild und Informationen über die Geometrie der Tür kann anschließend durch einen Rückwärtsschnitt die Position des Nutzers abgeleitet werden. Andere bildbasierte Anwendungen verwenden die Erkennung von markanten Objekten (Feature Er-

kennung) zur relativen Orientierung zwischen zwei Bildpaaren oder nutzen die Kamera zur Schritterkennung [Marouane u. Ebert, 2015]. Aufgrund der hohen Anforderungen an die Prozessorleistung, an die Lichtverhältnisse und die einheitliche Objektsichtbarkeit eignen sich bildbasierte Systeme gegenwärtig nur bedingt für smartphonebasierte Anwendungen zur Personennavigation.

Verfahren, die den hybriden Systemen zuzuordnen sind, basieren auf dem Einsatz von Inertialsensoren (INS) zur Bewegungserfassung. Die Sensoren liefern Informationen über Linearbeschleunigungen und Drehraten, mit denen sich die relative Position und Orientierung auf Basis von Strapdown Algorithmen oder Pedestrian Dead Reckoning berechnen lassen. Die INS sind dazu an dem Objekt bzw. dem Gerät des Nutzers angebracht, dessen aktuelle Position bestimmt werden soll. Sofern für die Positionsschätzung keine weiteren infrastrukturbasierten Messelemente Verwendung finden, stellt die INS-basierte Positionsschätzung ein autonomes Verfahren dar. Da allerdings die aus INS-Daten ermittelte Position zwangsläufig nur kurze Zeit von ausreichender Genauigkeit ist (je nach Qualität der eingesetzten Sensoren wenige Sekunden bis hin zu einigen Minuten), muss eine Korrektur durch zusätzliche externe Informationen vorgenommen werden. INS-basierte Positionsschätzverfahren werden daher üblicherweise in Kombination mit infrastrukturgestützten Systemen angewandt und daher als hybride Systeme bezeichnet.

Die Idee, unterschiedliche Sensoren zur Positionsbestimmung zu kombinieren, haben bereits Poor u. a. [2009], Fu u. Retscher [2009], Liu u. a. [2010], Klingbeil u. a. [2010b] und Romanovas u. a. [2013] untersucht und Systeme vorgestellt, die für spezielle Infrastruktur-Empfänger-Kombinationen brauchbare Ergebnisse erzielen. Neben der zusätzlich nötigen Hardware stellt die Größe dieser Systeme ([Poor u. a., 2009], [Liu u. a., 2010]) einen entscheidenden Nachteil dar, so dass sie nur unter bestimmten Bedingungen sinnvoll zu nutzen sind. Zudem ist die Anbringung zusätzlicher Sensoren im Gebäude (beispielsweise RFID-Sensoren) oder am Körper des Nutzers (beispielsweise ein Beschleunigungssensor am Fuß) erforderlich ([Fu u. Retscher, 2009], [Klingbeil u. a., 2010b], [Romanovas u. a., 2013]). Im Vergleich zu anderen Methoden setzen INS-basierte Verfahren für die Fußgänger-navigation eine starre Haltung am Körper voraus, um die Körperbewegungen und die Orientierung anhand der Daten richtig zuzuordnen zu können. Aus diesem Grund

müssen die Sensoren möglichst an Gürtel oder Fuß der Nutzer platziert werden, um den Einfluss weiterer Bewegungen des Körpers weitgehend auszuschließen.

Für die Fusionierung der in ihrer Stochastik und ihren Einheiten stark unterschiedlichen Sensordaten setzen Klingbeil u. a. [2010b] und Romanovas u. a. [2013] Kalman Filter (KF) und Partikelfilter (PF) ein, um eine Trajektorie zu erzeugen. Die Fusion von stochastisch stark unterschiedlichen Informationen kann besonders für den hybriden Ansatz eine komplexe Aufgabe darstellen, da sich die verwendeten Sensordaten hier besonders stark voneinander unterscheiden. Für den Einsatz von Kalman Filter und Partikelfilter müssen die Eingangsdaten samt der Unsicherheiten im Vorfeld möglichst bekannt sein. Dies ermöglicht eine optimale Abstimmung im Filter. Detailliertere Informationen zur Kalibrierung der Sensoren und auch zum Vergleich der Schätzverfahren finden sich bei Klingbeil [2006], Klingbeil u. a. [2010a] sowie Klingbeil u. Romanovas [2012].

Lukianto u. a. [2010], Lukianto u. Sternberg [2011a] und Lukianto u. Sternberg [2011b] stellen vorangegangene Arbeiten der Forschungsgruppe dar. Das Ziel war die Eigenentwicklung einer Low-Cost INS zur Fußgängernavigation und daher bilden sie die Basis für dieses Forschungsvorhaben. Sternberg u. Schwalm [2007], Sternberg u. a. [2009] und Sternberg u. a. [2010] thematisieren Problemstellungen, die zuvor zu der Formulierung des Forschungsvorhabens geführt haben. Dabei wurde der Einsatz einer Low-Cost INS (3.000 €) sowie der Aufbau einer eigenen Sensorplattform untersucht. Mittlerweile besitzen moderne Smartphones eine große Anzahl von MEMS-Sensoren (Micro Electro Mechanical System), die sich für die Indoor-Navigation eignen und daher ist der Aufbau einer eigenen Sensorplattform für die Fußgängernavigation nicht mehr notwendig.

1.2 Konzept und Ziel

Die große Zahl der in Smartphones integrierten Sensoren eröffnet viele Nutzungsmöglichkeiten, beispielsweise eine satellitengestützte Navigation mit eingebauten GNSS-Empfänger. Eine Verwendung für die Fußgängernavigation in GNSS-abgeschatteten Bereichen ist anzustreben, damit eine Navigation nach dem Prinzip *von Tür zu Tür* auch übergreifend im Gebäude sowie im Außenbereich funk-

tionieren kann. Infrastrukturbasierte Positionierungsverfahren für die Fußgänger-navigation funktionieren meist auf Basis von Funktechniken, für die es in den Smartphones entsprechende Empfänger gibt. Außerdem können die integrierten MEMS-Sensoren - wie Beschleunigungsmesser, Gyroskop, Magnetfeldsensor und Barometer - zur Positionsschätzung genutzt werden. Alle weiteren notwendigen Komponenten für eine Anwendung zur Indoor-Navigation - ein Display zur Visualisierung, ein leistungsfähiger Prozessor, der Zugang zum Internet und eine integrierte Kamera - gehören zur Standard-Ausstattung moderner Smartphones.

Da die MEMS-Sensoren im Smartphones integriert sind und somit zu jederzeit deren Sensordaten verfügbar sind, sind MEMS INS für eine autonome Positionsschätzung ideal. Aufgrund der mit der Zeit sinkenden Genauigkeit muss die Positionsbestimmung allerdings durch externe Informationen gestützt werden.

In Abb. 1.1 sind die möglichen Bausteine zur Umsetzung einer Indoor-Navigation auf Basis einer Positionsschätzung mit INS beschrieben. Das Konzept gliedert sich in fünf Teilbereiche [Willemsen u. a., 2014b]. Die Auswahl der Navigationshardware wird nach der Anwendung und der anzusprechenden Nutzergruppe entschieden. Beispielsweise eignet sich ein Tablet für eine Applikation als Museumsführer, da sich auf seinem relativ großen Display Informationen übersichtlich darstellen lassen. Das Smartphone als Gerät, das für den Nutzer stets verfügbar ist, bietet sich für spontan zu nutzende mobile Anwendungen an. Eigene Hardware-Lösungen könnten für den industriellen Einsatz erforderlich sein, da an die verwendete Hardware oft spezielle und hohe Anforderungen gestellt werden.

Um die Sensordaten für die Schätzverfahren optimal nutzen zu können, müssen Messunsicherheiten der Smartphone-Sensoren bekannt sein. Zudem müssen die Nutzungsmöglichkeiten der Sensoren für die Positionsschätzung in Abhängigkeit von der Qualität der Sensordaten untersucht werden. Beispielsweise ist von Interesse, ob das in Smartphones integrierte Barometer ausreicht, um den Übergang zwischen zwei Stockwerken in einem Gebäude zu erkennen.

Der algorithmisch aufwendigste Part bei der Entwicklung einer Anwendung zur Indoor-Navigation auf Basis von INS-Daten ist die Positionsschätzung. Das gilt besonders bei der Berücksichtigung vieler unterschiedlicher Informationen zu Korrekturen für die Fusionierung auf Basis von Kalman Filter und Partikelfilter.

1.2 Konzept und Ziel

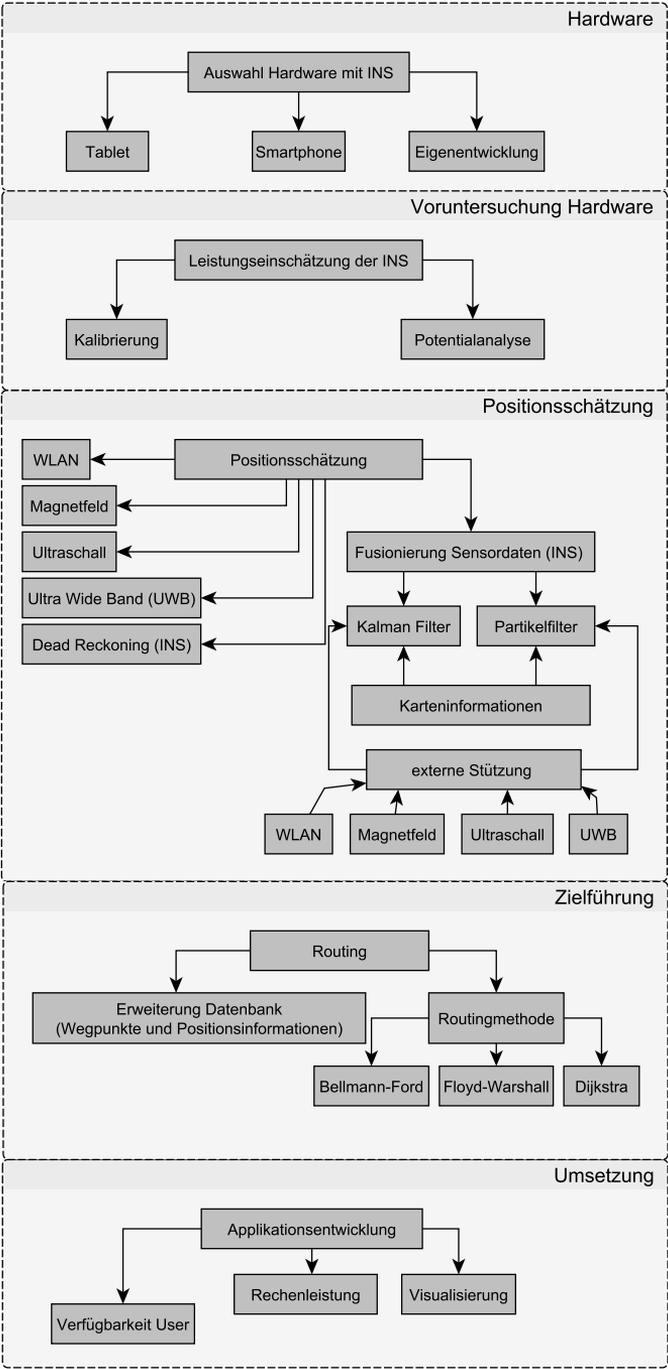


Abb. 1.1: Konzept zur Umsetzung eines Indoor-Navigationssystems [Willemsen u. a., 2014b]

Ein Navigationssystem benötigt neben der Bestimmung des Standortes auch ein Verfahren zur Zielführung. Dafür bieten sich verschiedene Routing-Algorithmen an, die je nach Kostendefinition alle Wegmöglichkeiten auf einem Routing-Graph berechnen lassen. Zu nennen sind hier der bidirektionale Dijkstra-Algorithmus und der auf ihm aufbauende A*-Algorithmus.

Da Eingriffe in die Infrastruktur der Gebäude notwendig sind und viele Methoden zur Positionsschätzung existieren, gibt es im Gegensatz zur GNSS-Navigation für die Indoor-Navigation gegenwärtig nur Insellösungen, die aufwendig sowie kostspielig zu implementieren sind. Unterschiedliche Anforderungen an die Nutzung und verschiedene Gebäudetypen machen zudem Eingriffe in die Infrastruktur notwendig. Eine Buchhandlung etwa weist andere bauliche Strukturen und ein völlig anderes Nutzungsspektrum auf als eine Messehalle oder ein Museum. Wirklich durchsetzen wird sich die Indoor-Navigation erst, wenn sie keine oder nur sehr wenige Eingriffe in die Infrastruktur der Gebäude voraussetzt.

Um eine breite Akzeptanz zu erlangen, sollte das Verfahren zur Positionsschätzung prinzipiell in allen Innenräumen oder Gebäuden nutzbar sein. Einzige Voraussetzung ist die Verfügbarkeit eines Grundbestands an Navigationsinfrastruktur wie ein Routing-Graph und Gebäudepläne, die beispielsweise aus Flucht- und Rettungsplänen abgeleitet werden könnten [Peter u. a., 2012].

Im Rahmen des vorliegenden Projekts wird für die Zielführung einer Person innerhalb von Gebäuden eine Genauigkeit von 1 bis 5 m in der Lage sowie eine eindeutige Raumidentifizierung angestrebt, dabei werden die 5 m in der Lage als maximale Abweichungen zur Referenz verstanden (MPE = Maximum Permissible Error). Die Höhenbestimmung soll eine Unterscheidung der Stockwerke und Ebenen erlauben.

Die Implementierung einer weitgehend autonomen Positionsschätzung unter Innenraumbedingungen auf einem Smartphone sowie die Nutzung lediglich typischer vorhandener Infrastrukturmerkmale öffentlicher Gebäude wird in diesem Forschungsvorhaben eine Grundvoraussetzung sein und stellt gegenüber bisherigen Lösungsansätzen eine deutliche Verbesserung bezüglich des Implementierungsaufwands als auch der Kosten dar.

Im ersten Teil dieser Arbeit wird die Testsensorik im Hinblick auf die Qualität der Messinformationen untersucht und dafür Ansätze zur Kalibrierung sowie die Art der Verwendung diskutiert. Als Testgeräte kommen ein Samsung Galaxy Nexus sowie ein Google Nexus 4 zum Einsatz, die jeweils über die Sensoren verfügen, die für die Positionsschätzung in GNSS-abgeschatteten Bereichen von Interesse sind. Darauf aufbauend wird ein erster Ansatz zur Fusionierung der Smartphone-Sensoren mittels Dead Reckoning behandelt. Anschließend werden ausgewählte Möglichkeiten zur Stützung einer Navigation in Abhängigkeit der Minimierung von Infrastrukturmaßnahmen vorgestellt.

Die Fusion aller favorisierten Möglichkeiten wird am Beispiel eines Kalman Filters, eines Partikelfilters und eines topologischen Ansatzes diskutiert. Anschließend wird der hier favorisierte Ansatz vorgestellt, der auf Basis eines Partikelfilters und des topologischen Ansatzes entwickelt wird. Mit den Ergebnissen einer Stichprobe aus 22 Nutzern und unterschiedlichen Smartphone-Modellen zur Untersuchung der allgemeinen Anwendbarkeit des favorisierten Fusionsfilters schließt das Kapitel ab. Das anschließende Kapitel widmet sich der Implementierung einer Navigationsapplikation sowie der halbautomatischen Generierung von Grundriss und Routing-Graph.

2 Grundlagen

2.1 Micro Electro Mechanical System (MEMS)

Grundlage der Positionsschätzung bilden die in Smartphones verbauten Low-Cost-Sensoren. Die für eine möglichst autonome Positionsschätzung geeigneten MEMS-Sensoren in den Geräten werden in diesem Abschnitt vorgestellt.

2.1.1 Technologie und typische Anwendungsgebiete

Micro Electro Mechanical Systems (MEMS) sind Sensoren, die stark miniaturisiert die Funktionen üblicherweise hochwertigerer und größerer Sensoren übernehmen. Eine Weiterentwicklung sind die Nano Electro Mechanical Systems (NEMS), die einen noch kleineren Sensortyp darstellen. Bei MEMS und NEMS werden mechanische, elektrische, optische und chemische Komponenten in Miniaturgröße verarbeitet. Von Wild-Pfeiffer u. Schäfer [2011] werden diese Sensoren mit folgenden Eigenschaften charakterisiert:

- Fertigung feinmechanisch nicht möglich,
- kompakte Bauweise,
- geringes Gewicht,
- hohe Ausfallsicherheit,
- geringer Leistungsbedarf,
- kostengünstige Produktion,
- parallele Fertigungstechnik vieler Mikrokomponenten auf einem Substrat (batch-fabricated).

Aufgrund dieser Eigenschaften und der großen Nachfrage werden die Sensoren im Handel zu vergleichsweise niedrigen Stückpreisen angeboten. Die Größe eines Sensors liegt in der Regel zwischen 0,02 mm bis 1,0 mm. Die einzelnen Bauteile der MEMS-Sensoren besitzen eine Größe von 0,001 mm und 0,1 mm. Als Fertigungsmaterialien kommen Silikon, Polymere, Metalle sowie Keramiken zum Einsatz. Wild-Pfeiffer u. Schäfer [2011] charakterisieren die Mikrosystemtechnik

als Querschnittstechnologie, da sie in verschiedensten Branchen von Bedeutung ist. Dazu zählen die IT-Branche, die Automobiltechnik und die Konsumelektronik als zentrale Anwendungsbereiche. In der IT-Branche finden sich MEMS-Sensoren in Notebooks und Tintenstrahldruckern, bei der Automobilindustrie sind Airbag-Systeme und ESP-Steuergeräte damit ausgestattet und in der Konsumelektronik dienen sie der Erweiterung des Nutzungsspektrums; für Komfort und Spieldynamik sorgen sie in Smartphones und in Spielkonsolen. Ein wichtiger Anwendungsbereich ist auch die Medizintechnik (Implantate, Blutdrucküberwachungssysteme), sie werden in der Telekommunikation und bei der industriellen Fertigung zur Systemüberwachung der Abläufe und Maschinen eingesetzt [Wild-Pfeiffer u. Schäfer, 2011].

Nutzungen finden sich auch im Vermessungswesen, beispielsweise zur Überwachung von Bauwerken oder bei kinematischen Messtechniken. Ein Beispiel für den Einsatz der MEMS-Sensorik zur Navigation ist das Low-Cost Inertial-Messsystem MTiG der Firma Xsens Technologies B.V. (Abb. 2.1). Dieses System besitzt einen Beschleunigungsmesser, ein Gyroskop und einen Magnetfeldsensor. Bei Bedarf kann zusätzlich ein GNSS-Empfänger angeschlossen werden. Bevor die Arbeit an dieser Untersuchung zu in Smartphones integrierten MEMS aufgenommen wurde, ist das Messsystem für die ersten Untersuchungen von MEMS-Sensoren für die Fußgängernavigation verwendet worden, wie beispielsweise in Lukianto u. a. [2010].

Weitere Anwendungen von MEMS-Sensoren sind bei Sternberg u. a. [2015] dokumentiert. Vorgestellt werden hier unter anderem die Steuerung eines kinematischen Reflektors für eine Mobile-Mapping-Plattform und ein Sensor zur Orientierungsbestimmung für die Steuerung von Audiosequenzen im Rahmen eines künstlerischen Projekts.

MEMS-Sensoren, die auf dem Markt erhältlich sind, erlauben eine freie Entwicklung von Modulen für unterschiedlichste Zwecke. Im Zuge der Vorarbeiten zu dieser Arbeit ist eine Sensorplattform entstanden, die in Abb. 2.2 zu sehen ist. Deutlich erkennbar auf der grünen Platine sind die einzelnen Inertialsensoren auf den kleinen roten Platinen. Es handelt sich bei diesen um einen dreiachsigen Beschleunigungsmesser, ein dreiachsiges Gyroskop, einen dreiachsigen Magnetfeld-



Abb. 2.1: Low Cost Inertial Messsystem MTiG von Xsens [Xsens, 2013].



Abb. 2.2: Platine mit MEMS-Sensoren auf roten Platinen [Lukianto u. Sternberg, 2011b]

sensor sowie einen Luftdrucksensor [Lukianto u. Sternberg, 2011b]. Smartphones bieten sich zur Entwicklung einer INS-gestützten Navigation wegen der rapiden technologischen Fortschritte in diesem Bereich, der immensen Verbreitung der Geräte und der relativ simplen Möglichkeit zur Visualisierung an. Sinnvoll ist eine Eigenentwicklung dennoch, da die gemessenen Rohdaten gegenüber einer Anwendung im Smartphones unverfälscht sind und nicht vom Betriebssystem modifiziert werden. Im Folgenden werden Beispiele für Funktionsprinzipien der jeweils verwendeten Sensoren – Beschleunigungsmesser, Gyroskop, Magnetfeldsensor sowie Barometer – vorgestellt.

2.1.2 Beschleunigungsmesser

Ein dreiachsiger Beschleunigungsmesser, wie er in Smartphones genutzt wird, registriert Beschleunigungen des Sensors entlang der jeweiligen Achsrichtungen. Bei Stillstand verteilt sich die Erdbeschleunigung mit $g = 9,813 \text{ m/s}^2$ auf alle drei Achsen. Diese Eigenschaft bildet die Basis der Sensorkalibrierung (Kapitelabschnitt *3.3 Kalibrierung eines Beschleunigungsmessers*).

Beim Feder-Masse-Prinzip wird die Funktion anhand einer Probemasse, die an zwei Federn aufgehängt ist, hergestellt. Registriert der Beschleunigungsmesser einen Impuls, bewegt sich die Probemasse aufgrund ihrer Trägheit innerhalb des

2.1 Micro Electro Mechanical System (MEMS)

Sensoren zunächst nicht mit. In Abb. 2.3 ist das Prinzip entlang einer Achse dargestellt. Die Auslenkung der Probemasse ist dabei proportional zur Beschleunigung [Wild-Pfeiffer u. Schäfer, 2011]. Nach dem Zweiten Newtonschen Gesetz gilt für die Trägheitskraft $F = m * a$, wobei m für die Masse und a für den Wert der Beschleunigung steht. Für die Federkraft F gilt entsprechend: $F = k * \Delta l$ mit k für die Federkonstante und Δl für die Längenänderung der Masse. Nach der Gleichsetzung und Umstellung lässt sich nun die Beschleunigung berechnen mit (1):

$$a = \frac{k}{m} * \Delta l \quad (1)$$

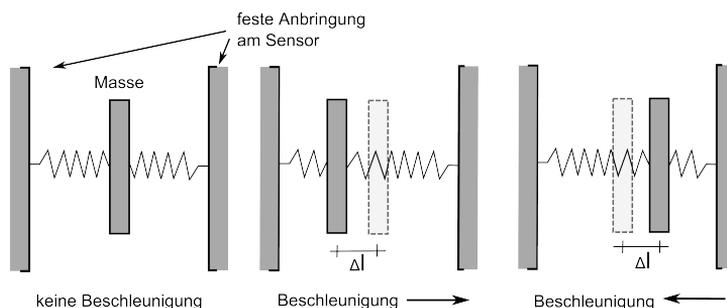


Abb. 2.3: Feder-Masse-Prinzip eines Beschleunigungsmessers entlang einer Achse (abgeleitet nach: <http://www.elektronik-kompodium.de/sites/bau/1503041.htm>)

Bei dreiachsigen Sensoren werden drei orthogonal zueinander ausgerichtete Probemassen mit Federn benötigt. Die Funktionsprinzipien der Bauarten unterscheiden sich in der Verarbeitung des mechanischen Vorgangs zu einem elektrischem Signal [Aggarwal u. a., 2010]. Die ersten Micro-Beschleunigungssensoren waren die piezoresistiven Beschleunigungssensoren. Sie sind stark temperaturabhängig ($0,2 \text{ \%}/^\circ\text{C}$) und erreichen nur eine insgesamt geringe Auflösung. Kapazitive Beschleunigungssensoren enthalten eine bewegliche Masse, die durch Bewegung zwischen zwei leitenden Elementen ihre Kapazität verändert. Diese Sensoren ermöglichen hohe Auflösungen und sind nur geringfügig temperaturabhängig. Piezoelektrische Beschleunigungssensoren bestimmen die Bewegung der Probemasse direkt zu einem elektrischen Signal und benötigen keine zusätzliche Umsetzung.

Tunneling Beschleunigungssensoren schließlich zeichnen sich durch ihre sehr hohe Auflösung und einen sehr weiten Messbereich aus.

Die bekannte Fehlercharakteristik von teuren Inertialsensoren ist auch hier gültig [Wendel, 2007]. Ein vereinfachtes Fehlermodell zeigt die folgende Formel (2). Um die tatsächliche Beschleunigung $\vec{\tilde{a}}$ messen zu können, sind Orthogonalität, Maßstäbe, Offsets sowie das sensorinhärente Rauschen zu korrigieren. Hierbei werden Glieder höherer Ordnung vernachlässigt, die sich dadurch im sensorinhärentem Rauschen mit abbilden.

$$\vec{\tilde{a}} = M_{Acc} * \vec{a} + \vec{b}_a + \vec{n}_a \quad (2)$$

In M_{Acc} befinden sich auf der Hauptdiagonalen die Skalenfaktoren zur Maßstabskorrektur und auf den Nebendiagonalen die Parameter zur Korrektur der Nicht-Orthogonalität. Der Vektor für die Offsetkorrektur \vec{b}_a ist vorzeichenrichtig an die Rohdaten anzubringen. Die Orthogonalität, der Maßstab und der Offset lassen sich mit Hilfe eines Referenzwertes für die Erdbeschleunigung kalibrieren. Der Vektor \vec{n}_a beschreibt das sensorinhärente Rauschen [Wendel, 2007]. Bei Inertialsensoren verhält sich das Rauschen nicht um Null normalverteilt und verändert sich über den Messzeitraum stetig, sodass es sich nur anteilig korrigieren lässt. Durch die Integration von Beschleunigungen zu Geschwindigkeiten produziert das sensorinhärente Rauschen daher eine Drift. Bei MEMS-Sensoren ist dieser Effekt aufgrund des vergleichsweise großen Rauschens sehr ausgeprägt.

2.1.3 Gyroskop

Gyroskope werden als Drehraten-Sensoren bezeichnet. Sie registrieren die Rotationsgeschwindigkeit um die jeweiligen Achsen des Sensorkoordinatensystems. Ihre Funktionsweise basiert auf mechanischen, optischen oder vibrationsbasierten Methoden. Bei der MEMS-Technologie wird aus Kostengründen auf die Vibrationsmethode zurückgegriffen. Allerdings ist sie nicht so genau wie die mechanische oder die optische Methode [Wild-Pfeiffer u. Schäfer, 2011].

In Abb. 2.4 ist das Funktionsprinzip eines Vibrationskreisels um die z-Achse darge-

2.1 Micro Electro Mechanical System (MEMS)

stellt. Basierend auf dem Coriolis-Effekt wird eine Masse in x-Richtung in Schwingung versetzt. Erfolgt anschließend eine Drehung des Sensors um die z-Achse, so entsteht eine Beschleunigung entlang der y-Achse mit gleicher Frequenz. Die Amplitude dieser Schwingung ist das Maß für die Drehgeschwindigkeit um die z-Achse.

Dieser Zusammenhang wird mit $\vec{f} = 2 * m * \vec{v} * \Omega_z$ beschrieben, wobei \vec{f} für die Frequenz in y-Richtung steht, \vec{v} die Geschwindigkeit der Prüfmasse m und Ω_z die Drehgeschwindigkeit um die z-Achse repräsentiert. Die Umsetzung im MEMS-Sensor kann über prismatische Biegebalken, Stimmgabel-Strukturen, rotationssymmetrische Resonanzkörper oder nach dem Feder-Masse-Prinzip erfolgen [Wild-Pfeiffer u. Schäfer, 2011].

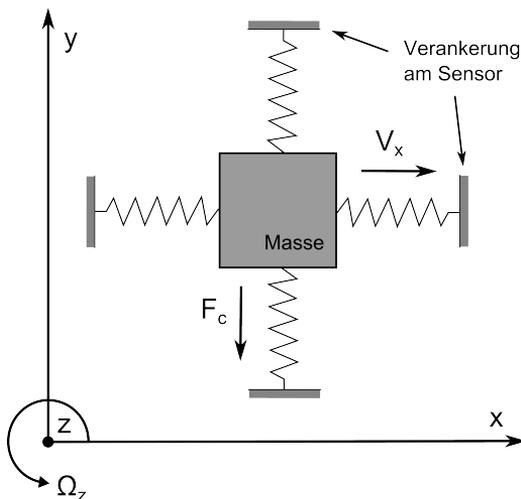


Abb. 2.4: Feder-Masse-Prinzip um die Z-Achse für ein Gyroskop (abgeleitet nach: http://www.hsg-imit.de/fileadmin/gfx/pdfs/AiF-Praesentationen_2007/ok_2007_03_22_AiF-Tag_RefGyro-final_Jan_Dehnert.pdf)

Das Fehlermodell des Drehratensensors kann vom oben beschriebenen Beschleunigungsmesser übernommen werden und ist in Formel (3) für den Drehratensensor angepasst. Es umfasst die Nicht-Orthogonalität auf den Nebendiagonalen von M_{Gyr} , die Skalenfaktoren auf der Hauptdiagonalen von M_{Gyr} , die Offsets \vec{b}_r und das inhärente Rauschen \vec{n}_r .

$$\vec{r} = M_{Gyr} * \vec{r} + \vec{b}_r + \vec{n}_r \quad (3)$$

2.1.4 Magnetfeldsensor

Die Mehrzahl der in Smartphones verbauten Magnetfeldsensoren basiert auf dem Hall-Effekt zur Bestimmung des äußeren Magnetfelds [Cai u. a., 2016]. Der Hall-Effekt beschreibt die Bewegung von Elektronen in einem stromdurchflossenen elektrischen Leiter. Dieser Leiter besteht bei einem MEMS-Magnetfeldsensor aus einem dünnen Plättchen mit einer Ausdehnung von ca. $0,2 * 0,2 * 0,01 \text{ mm}^3$. Auf diesem Leiter wird durch die Spannung V ein künstliches Magnetfeld erzeugt, das aus dem Gleichgewicht gerät, sobald der Einfluss eines äußeren Magnetfelds wirksam wird. Dies zeigt sich in einem Überhang von Elektronen bzw. in einem elektrischen Feld, das senkrecht zum Magnetfeld und zur Stromrichtung I steht (Abb. 2.5). Die Änderung der Spannung ist ein Maß für die Stärke des Magnetfeldes.

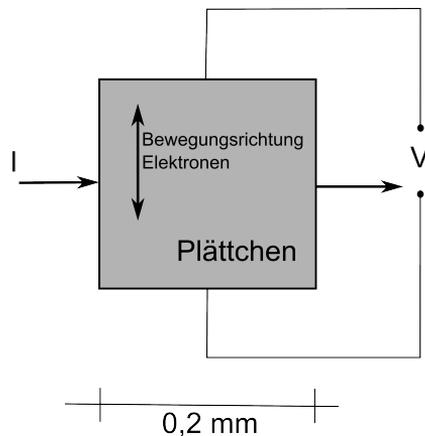


Abb. 2.5: Funktionsweise einer Achse des Magnetfeldsensors auf Basis des Hall-Effekts (abgeleitet nach: [Cai u. a., 2016]).

Nachteile der auf dem Hall-Effekt basierenden Magnetfeldsensoren sind die geringe Sensibilität und die hohe Temperaturempfindlichkeit. Sie werden aber durch Vorteile wie geringe Größe, niedriger Energieverbrauch und geringe Fertigungskosten kompensiert. Die Messunsicherheiten sind vergleichsweise gering angesichts der

2.1 Micro Electro Mechanical System (MEMS)

üblichen Störgrößen bei Innenraumanwendungen. Bei diesen Störgrößen handelt es sich um lokale, künstliche Magnetfelder, hervorgerufen durch metallische Gegenstände oder elektronische Geräte, die sich mit dem Erdmagnetfeld überlagern. Die Fehlercharakteristik eines dreiachsigen Sensors entspricht dem des Beschleunigungsmessers sowie des Gyroskops und kann beschrieben werden durch die Formel (4):

$$\tilde{\vec{n}} = M_{Mag} * \vec{m} + \vec{b}_m + \vec{n}_m \quad (4)$$

Die Nullpunktfehler sind in \vec{b}_m , die Nicht-Orthogonalitäten auf den Nebendiagonalen von M_{Mag} , die Maßstäbe in M_{Mag} auf der Hauptdiagonalen und das sensorinhärente Rauschen in \vec{n}_m zu finden.

2.1.5 Barometer

Barometer messen den Luftdruck und können, da der Luftdruck mit der Höhe abnimmt, auch zur Höhenmessung eingesetzt werden.

Bei einem MEMS-Barometer wird eine Membran verwendet, die auf der Außenseite dem Luftdruck der Umgebung ausgesetzt ist und auf der inneren Seite einen Hohlraum einschließt, der einen Referenzluftdruck enthält. Die Verformung der Membran durch Veränderungen des äußeren Luftdrucks und die damit einhergehende Veränderung des Materialwiderstands liefert die Berechnungsgrundlage für den vom Sensor erfassten Luftdruck [Wang, 2012]. Eine Realisierung mit Dehnungstreifen, die die Veränderung der Membran registrieren, zeigt Abb. 2.6.

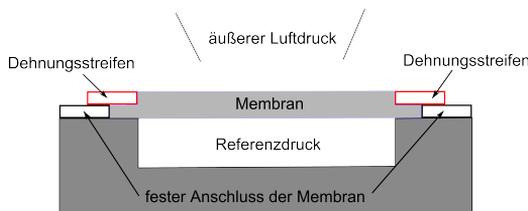


Abb. 2.6: Funktionsprinzip für einen MEMS-Barometer als Querschnitt durch einen Sensor (abgeleitet nach: <http://sus.ziti.uni-heidelberg.de/Lehre/Seminar03/07.pdf>).

Eine Höhenmessung unterliegt, unabhängig von der Qualität der Systeme, stets dem Einfluss der Wetterdrift. Für genaue Messungen wird daher ein Referenzgerät eingesetzt, dessen Standort während des gesamten Messvorgangs nicht verändert wird. Bei Innenraum-anwendungen können weitere Effekte hinzukommen, die sich – wie etwa der Wechsel von geschlossenen zu offenen Bereichen in Treppenhäusern oder Räumen mit offenen Fenstern – negativ auf die Höhenbestimmung durch Luftdruckmessungen auswirken. Auch kann die kompakte Bauweise der Smartphones und die damit gegebene Erwärmung des Geräts bei längerer Nutzung die Luftdruckmessung in den stark temperaturabhängigen MEMS-Barometern verfälschen. Die Fehlercharakteristik eines Barometers kann beschrieben werden durch die Formel (5). Wobei die Luftdruckmessung p durch die Luftdruckänderungen aufgrund der Wetterdrift p_w , von Temperaturveränderungen p_t sowie durch die Sensordrift d_p und dem inhärenten Rauschen n_p verfälscht wird.

$$\tilde{p} = p + p_w + p_t + d_p + n_p \quad (5)$$

2.2 Stochastische Filter

Stochastische Filter dienen dazu, Zustände linearer bzw. nichtlinearer Systeme zu schätzen. Dazu binden sie vorhandene Informationen zur Stützung ein [Wendel, 2011]. Es kann vorkommen, dass nicht zu jeder Zeit Stützinformationen vorliegen, diese nur teilweise verfügbar oder von schlechter Qualität sind. Dies Wissen um die Unsicherheiten der Zustände und der eingehenden Stützinformationen verwenden die stochastischen Filter, um optimale Zustände für jeden Zeitpunkt zu bestimmen. Letztendlich ist das Design eines Filters entscheidend für die Robustheit der Berechnung und der erreichbaren Qualität der Ergebnisse aus den verfügbaren Daten. Das bekannteste stochastische Filter ist das Kalman Filter. Im Folgenden werden das Kalman Filter und das Partikelfilter vorgestellt und anschließend ihre prinzipiellen Unterschiede diskutiert.

2.2.1 Kalman Filter

Kalman Filter (KF) sind effiziente Werkzeuge zur Schätzung stochastischer Prozesse [Aggarwal u. a., 2010]. Sie zeichnen sich besonders im Hinblick auf die Handhabbarkeit unterschiedlichster Informationen aus. Diese können sich in Auflösung, Dimension, Fehlercharakteristikum und Zeitpunkt der Verfügbarkeit unterscheiden. Voraussetzung dabei ist, dass von normalverteilten Unsicherheiten ausgegangen wird.

Allgemeiner Ansatz

Der Ansatz nach Kalman verfolgt die Schätzung eines Zustandes in Abhängigkeit von vorangegangenen Schätzungen und den dabei erreichten Genauigkeiten. Die Schätzung erfolgt auf Basis eines Systemmodells, das den zukünftigen Zustand möglichst genau beschreibt. Sie wird in definierten Zeitschritten fortgeführt, muss dabei aber wegen des nicht perfekt angepassten Systemmodells oder wegen möglicher äußerer Einflüsse regelmäßig eine Stützung erfahren [Kalman, 1960]. Die Art und Weise dieser Korrektur wird durch das Messwertmodell definiert, in das die möglichen Messungen oder andere Informationen zur Stützung einfließen. Das Messwertmodell und das Systemmodells werden in der Estimation bzw. Korrektur zueinander transformiert und in Beziehung gesetzt.

Der Fall beschreibt einen linearen Zustand x^- , der aus der Transitionsmatrix A , welche hier auch als Bewegungsmodell bezeichnet wird, und dem vorherigen Zustand x berechnet wird. Der Zustandsvektor x und die Transitionsmatrix A werden als Systemmodell bezeichnet. Um die Unschärfe des Bewegungsmodells zu berücksichtigen, wird ein Systemrauschen w eingefügt (6). Mit dem Systemrauschen w , das normalverteilt, weiß verrauschte Werte enthält, kann die Steifigkeit des Systemmodells gegenüber dem Einfluss der gemessenen Größen gesteuert werden [Wendel, 2011]. Die im allgemeinen Fall üblicherweise eingebundene Störeingangsmatrix B und die weiteren Stör- bzw. Stellgrößen u werden hier vernachlässigt, da eine Berücksichtigung im Systemmodell, der in dieser Arbeit eingesetzten Filter nicht notwendig ist.

$$x^- = A * x + w \tag{6}$$

Eine zukünftige Schätzung ist bei diesem Ansatz immer von der vorangegangenen Schätzung abhängig. Die Unsicherheiten vergrößern sich entsprechend mit jedem Berechnungsschritt. Die Kovarianzen jeder einzelnen Schätzung werden mit

$$Q_x^- = A * Q_x * A' + Q_w \quad (7)$$

berechnet. Die Kovarianzmatrix Q_x der vorherigen Schätzung wird mit der Transitionsmatrix A , in Beziehung gesetzt. Auch hier muss das Systemrauschen Berücksichtigung finden. Dazu wird die Kovarianzmatrix des Systemrauschens Q_w addiert. Die Matrix Q_w enthält daher auf der Hauptdiagonalen die quadrierten Rauschterme aus dem Vektor w . Diese Schätzung oder auch Propagation wird mit der Zuordnung von x^- nach x abgeschlossen, sofern keine Korrektur in der Estimation durch reale Messwerte erfolgt.

Für eine Estimation des KF müssen die vorhandenen Stützelemente bzw. Eingangsgrößen an den Schätzparametern angepasst werden. Hierfür wird die Messmatrix H aufgebaut. Sie bildet den Übergang zwischen Messwertvektor y und Zustandsvektor x . Der Zusammenhang zwischen Messwertvektor und Zustandsvektor zeigt (8):

$$y = H * x + v. \quad (8)$$

Die Unsicherheiten der gemessenen Eingangsgrößen werden als Messrauschen v bezeichnet und durch Addition berücksichtigt. Das Messrauschen wird wie das Systemrauschen w als weißes Rauschen angenommen. Die Intensität der Korrektur wird abhängig von den bekannten Messunsicherheiten mit Hilfe der Kalman-Verstärkungsmatrix K berechnet. Die auch als Gainmatrix bezeichnete Kalman-Verstärkungsmatrix enthält Gewichtungsfaktoren und steuert den Grad der Korrektur aus der Differenz der gemessenen Größen zur aktuellen Schätzung (9). Die bekannten Größen für die zuvor geschätzte Kovarianzmatrix und die Messmatrix gehen in die Formel ein. Die Kovarianzmatrix des Messrauschens R_v beinhaltet die quadrierten Unsicherheiten des Messrauschens v . Das Systemrauschen und

das Messrauschen bilden die Stellschrauben zwischen Propagation und Estimation [Wendel, 2011].

$$K = Q_x^- * H' * (H * Q_x^- * H' + R_v)^{-1} \quad (9)$$

Nach der Berechnung der Verstärkungsmatrix K kann die Estimation mit Hilfe der Messwerte erfolgen:

$$x = x^- + K * (y - H * x^-) \quad (10)$$

$$Q_x = (I - K * H) * Q_x^- \quad (11)$$

Das x wird aus der Differenz der aktuellen Schätzung x^- und Multiplikation der Gain-Faktoren mit den mit H transformierten Messgrößen bestimmt. Anschließend werden die Kovarianzen mit Hilfe der Gain-Matrix angepasst. Nach erfolgter Estimation kann mit dem neuen Zustandsvektor x sowie der Kovarianz-Matrix Q_x eine neue Schätzung x^- und Q_x^- berechnet werden (6, 7). Die Schätzung des Systemzustandes wird verbessert und die Kovarianz-Matrix des Schätzers angepasst.

Das Prinzip des KF gleicht einer sequentiellen Ausgleichung, sofern kein Systemrauschen w vorliegt. Es ist modular aufgebaut und benötigt daher keine feste Abfragestruktur von Messwerten zur Stützung. Es bietet sich aber an, die Schätzung des Systemzustandes in festen Zeitintervallen durchzuführen und auf die Verfügbarkeit von Messwerten für den Estimationsschritt zu warten.

Wird das KF als Parameterschätzer eingesetzt – wie in dieser Arbeit zur Kalibrierung des Beschleunigungsmessers und des Magnetfeldsensors –, so streben die Varianzen der Schätzer gegen null, wenn die Anzahl der Messungen gegen unendlich ansteigt. Je nach Fragestellung bieten sich Variationen der hier beschriebenen Struktur des KF an. Modifizierte KF sind der Erweiterte Kalman Filter, der Sigma Point Kalman Filter und der Iterative Erweiterte Kalman Filter (Kalman Filter 2. Ordnung) [Wendel, 2011]. In der vorliegenden Arbeit bleiben andere Varianten

des KF außer Betracht, da Sie keinen Mehrwert zur Positionsschätzung liefern oder aber vom Ansatz her lediglich eine Mischung von KF und dem weiter unten vorgestellten Partikelfilter darstellen. Für einen Überblick werden zunächst die bekanntesten Variationen des KF kurz beschrieben.

Modifikationen des Kalman Filters

Erweitertes Kalman Filter

Das Erweiterte Kalman Filter (EKF) wird auch als indirekter Kalman Filter oder total-state-space-Kalman-Filter bezeichnet. Es dient dazu, nichtlineare Prozesse zu schätzen. Dazu wird der geschätzte Systemzustand x durch ein nichtlineares Systemmodell repräsentiert (12). Die Transitionsmatrix A und die Messmatrix H enthalten mittels Taylor-Reihen-Entwicklung linearisierte Parameter. Einziger Unterschied im Berechnungsablauf des EKF zum KF ist die Berechnung der Jacobi-Matrizen für jeden Zeitschritt, so dass ein entsprechend großer Rechenaufwand erforderlich ist. Die Änderungen zum allgemeinen Ansatz zeigen (12), (13) und (14):

$$y = h(x) + v. \tag{12}$$

$$a = \frac{\partial h(x)}{\partial x} \tag{13}$$

$$x = x^- + K * (y - h(x^-)) \tag{14}$$

Das Filter verarbeitet statt der vorliegenden Messwerte nur die Differenz zwischen den Messwerten y und den geschätzten erwarteten Messwerten $h(x^-)$ [Wendel, 2011]. Das EKF hat keine erwartungstreue Schätzung. Daher erzeugen Nichtlinearitäten von System- und Messmodell Unsicherheiten und können den Filter zum Divergieren bringen.

Sigma-Point Kalman Filter

Das Sigma-Point Kalman Filter ähnelt vom Aufbau her einem Partikelfilter. Eine definierte Anzahl an Sigma-Punkten repräsentiert durch die Verteilung den

Mittelwert und die Kovarianz des normalverteilten Zufallsvektors. Jeder Sigma-Punkt repräsentiert dabei einen Zustandsvektor. In der Propagation wird nun jeder Sigma-Punkt einzeln behandelt, so dass Mittelwert und Varianz bestimmt werden können. Die Umsetzung kann linear oder mit nichtlinearen Modellen (EKF) erfolgen. Je nach Wahl der Skalierungsparameter und der Gewichtungsfaktoren sind weitere Differenzierungen nötig. Der Unscented Kalman Filter und der Central Difference Kalman Filter sind die bekanntesten Varianten des Sigma-Point Kalman Filters.

Ein Sigma-Point Kalman Filter ist in der Lage, bei nichtlinearen Zusammenhängen genauere Zustände zu erfassen, als dies allein ein EKF könnte. Dafür ist im Normalfall ein höherer Rechenaufwand erforderlich. Nach Wendel [2011] ist der Einsatz eines Sigma Point-Kalman Filters gegenüber einem EKF sinnvoll, wenn ein zu beschreibender Zustand hochgradig nichtlinear ist. Bei linearen Systemen liefern Kalman Filter und Sigma-Point Kalman Filter identische Ergebnisse.

Iterativer Erweiterter Kalman Filter

Das Iterativ Erweiterte Kalman Filter dient dazu, die Leistungsfähigkeit des Kalman Filters vergleichsweise unkompliziert zu verbessern. Durch die Optimierung der Messmatrix H soll die Zustandsschätzung vereinfacht werden. Dazu wird der Messwert, wie beim EKF üblich, verarbeitet. Statt einer parallelen Bestimmung der Kovarianzmatrix wird auf Basis der neuen Zustandsschätzung das Messmodell erneut linearisiert. Dieser Vorgang kann einige Male wiederholt werden, bis sich die Zustandsschätzung nicht mehr ändert. Anschließend erfolgt die Bestimmung der Kovarianzmatrix [Wendel, 2011].

Eine weitere Möglichkeit besteht in der Variation des EKF mit einer zusätzlichen zweiten Ableitung in der Linearisierung in der Modellbildung. Dieser Filter wird auch als Kalman Filter zweiter Ordnung bezeichnet. Ist bereits vor Modellbildung bekannt, dass ein Filter zweiter Ordnung benötigt wird, schlägt Wendel [2011] jedoch vor, gleich den Sigma-Point Kalman Filter anzuwenden.

2.2.2 Partikelfilter

Partikelfilter (PF) werden den Monte Carlo Methoden zugeordnet und dienen der Verarbeitung hochgradig nichtlinearer Zusammenhänge. Dabei kann der PF anders als das KF jede beliebige Verteilung repräsentieren. Die Darstellung der Unsicherheiten erfolgt durch eine Anzahl von möglichen nächsten Schätzungen (Partikeln) [Aggarwal u. a., 2010]. Im folgenden Abschnitt wird der allgemeine Aufbau des Partikelfilters am Beispiel des Bootstrap-Partikelfilters beschrieben. Es gibt eine große Anzahl an Variationen des Partikelfilters. Die Algorithmen zur Positionsschätzung werden in dieser Arbeit allerdings auf Basis des Bootstrap-Partikelfilters umgesetzt.

Allgemeiner Ansatz (Bootstrap Partikelfilter)

Beim Bootstrap-Partikelfilter wird die Schätzung des Zustands mit Hilfe einer zuvor definierten Anzahl von möglichen neuen Zuständen und den Partikeln sowie in Abhängigkeit zur Wahrscheinlichkeitsdichtefunktion (WDF) vorgenommen. Die WDF steuert dabei die Gewichtung der zu schätzenden Partikel in Abhängigkeit von den eingehenden Korrekturdaten [Aggarwal u. a., 2010].

Grundlage des Bootstrap-Filters bildet ein nichtlineares Systemmodell $f(x_{k-1})$, mit dem sich die Zustände x schätzen lassen (15). Die Messelemente zur Korrektur werden mit y beschrieben und der Zusammenhang zu den geschätzten Zuständen im Messmodell mit $h(x)$ gekennzeichnet. Das Systemrauschen w und das Messrauschen v werden als weißes Rauschen angenommen.

$$x = f(x_{k-1}) + w_k \quad (15)$$

$$y = h(x_k) + v_k \quad (16)$$

Die Wahrscheinlichkeitsdichte $p(x_{0:k}|y_{1:k})$, die die Wahrscheinlichkeit der aktuellen Schätzung beschreibt, wird durch die zufällige Verteilung von N Partikeln mit einem Zufallszahlengenerator angenähert. Dadurch entstandene Approximationsfehler verschwinden, wenn $N \rightarrow \infty$ strebt. Im Initialisierungsschritt des Partikel-

filters wird der gesamte mögliche Wertebereich durch die Partikel abgedeckt. Die Wahrscheinlichkeitsdichte wird daher beschrieben durch (17).

$$p(x_{0:k}|y_{1:k}) = \sum_{i=1}^N w^i * \delta(x_{k-1} - x_{k-1}^i) \quad (17)$$

Wobei w^i die jeweiligen Gewichte der Partikel repräsentiert, für diese gilt $\sum_{i=1}^N w^i = 1$. Die Diracsche Delta-Funktion wird durch δ dargestellt und definiert den Wertebereich zwischen 0 und 1 für $\delta(x - x_0)$. Die Bestimmung der Wahrscheinlichkeitsdichte bildet den zweiten Schritt der Initialisierung des Partikelfilters. Da zum Initialisierungsschritt des Filters oft keine weiteren Informationen verfügbar sind oder schon in die Streuung der Partikel eingeflossen sind, können hier die Partikel mit $w^i = 1/N$ gewichtet werden.

Nach der Initialisierung des Partikelfilters folgt die Zustandsschätzung, die Propagation. In der Propagation werden mit Hilfe der Wahrscheinlichkeitsdichte p_{w_k} in Abhängigkeit zum Systemmodell $f(x_{k-1}^i)$ neue Partikel erzeugt (18), um eine bessere Approximation an die WDF zu erhalten [Wendel, 2011]. Die Partikelgewichte w_i bleiben in der Propagation unverändert.

$$x_k^i \propto p_{w_k}(x_k - f(x_{k-1}^i)) \quad (18)$$

Der Estimationsschritt beinhaltet die Stützung durch Messwerte und erfolgt analog zum Propagieren mit (19):

$$p(y_k|x_k) = p_{v_k}(y_k - h_{x_k}). \quad (19)$$

Daraus folgt, dass im Estimationsschritt lediglich die Gewichte w^i der Partikel neu berechnet werden (20) und die Partikel selbst unverändert bleiben. Dabei ist die Normierungskonstante c mit (21) so zu wählen, dass die Summe der Gewichte 1 ergibt.

$$w^{i,+} = c * p_{v_k}(y_k - h_{x_k^i}) * w^i \quad (20)$$

$$c = \frac{1}{\sum_{i=1}^N p_{v_k}(y_k - h_{x_k^i}) * w^i} \quad (21)$$

Durch die Gewichtung der einzelnen Partikel in der Estimation kann der Zustand eintreten, dass nur wenige Partikel ein hohes Gewicht erhalten. Dies wird als Degeneration der Partikelwolke bezeichnet und kann zur Folge haben, dass das Filter nach wenigen Iterationen divergiert. Um dies zu verhindern, wird ein sogenanntes Resampling eingeführt. Dabei werden wenige hoch gewichtete Partikel in mehrere Partikel aufgeteilt, die sich in der Summe das jeweilige Gewicht teilen. Das Ziel ist es, jedem Partikel das gleiche Gewicht mit $1/N$ zuzuordnen. Gesteuert wird das Resampling durch einen Zufallsgenerator. Je nach Gewicht erhalten die Partikel bestimmte Intervallgrenzen. In einem Intervall von 0 bis 1 werden nun Zufallszahlen gezogen. Wenn die Intervallgrenze eines Partikels durch die Zufallszahl ausgewählt wird, wird das Partikel mit dem Gewicht $1/N$ reproduziert. Dies wird N -mal wiederholt. Auf diese Weise werden hochgewichtete Partikel häufiger reproduziert und weniger hoch gewichtete Partikel kaum oder gar nicht reproduziert [Wendel, 2011].

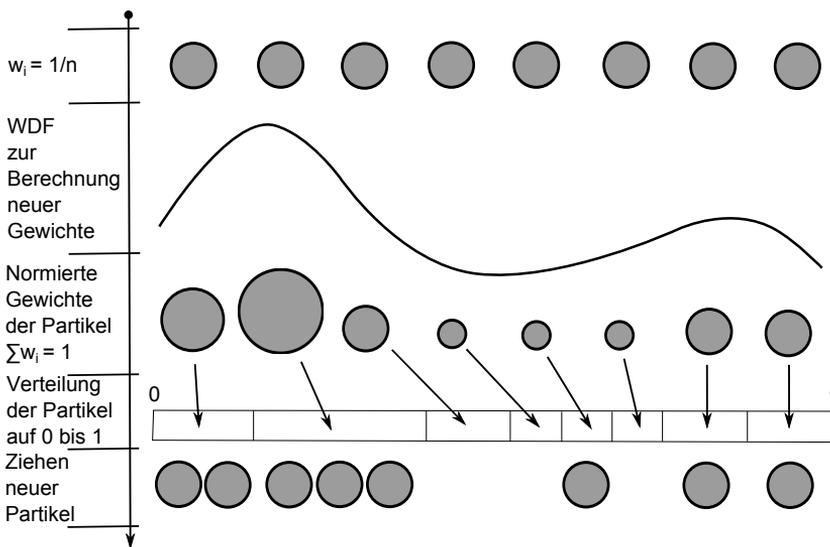


Abb. 2.7: Prinzip des Resamplings im Bootstrap Partikelfilter (abgeleitet nach: [Aggarwal u. a., 2010] und [Wendel, 2011])

In Abb. 2.7 wird der Ablauf des Resamplings gezeigt. Bei diesem Beispiel werden die Partikel in der obersten Zeile zu Beginn als gleich gewichtet betrachtet. Die Größe der Kreise repräsentiert die Höhe der Gewichtung. Die WDF erlaubt die Berechnung neuer Gewichte auf Basis von eingehender Stützinformationen, beispielsweise einer Streckenmessung, einer Abstandsmessung oder Karteninformationen. Die Partikel erhalten neue Gewichte, wie in Abb. 2.7 in der mittleren Zeile dargestellt und an der Größe der Kreise zu erkennen. Die neuen Partikel werden anschließend einem Resampling unterzogen und es entstehen gleich hoch gewichtete Partikel, die anschließend in der Propagation Verwendung finden. Wann ein Resampling durchgeführt werden muss, kann durch die Anzahl effektiver Partikel N_{eff} gesteuert werden (22).

$$N_{eff} = \frac{1}{\sum_{i=1}^N (w^i)^2} \quad (22)$$

Wenn alle Partikel das gleiche Gewicht aufweisen, dann gilt $N_{eff} = N$, im Gegensatz dazu gilt $N_{eff} = 1$, wenn ein Partikel das gesamte Gewicht auf sich vereint. Als Grenzwert kann eine frei definierbare Schwelle festgesetzt werden. Wendel [2011] favorisiert 70 % als Grenzwert für das Resampling.

Modifikationen des Partikelfilters

Die unterschiedlichen Modifikationen des Partikelfilters unterscheiden sich hauptsächlich in der Anpassung einzelner Komponenten, wie beispielsweise bei den Resampling-Methoden. Im Folgenden werden einige Variationen des Partikelfilters kurz beschrieben.

Regularized Partikelfilter (RPF)

Eine Weiterentwicklung des Partikelfilters ist das KLD-Sampling auf Basis der Kullback-Leibler-Distanz. Es ist ein Maß für die Vergleichbarkeit zweier Verteilungen. Das KLD-Sampling verfolgt den Ansatz, die Anzahl der Partikel der Genauigkeit der Schätzung anzupassen. Beispielsweise kann eine Initialisierung mit vielen Partikeln stattfinden. Ist anschließend eine gute Position gefunden, können die Partikel reduziert werden, um die Performance des Filters zu verbessern. Ein Partikelfilter mit dieser Sampling-Methode wird auch als Adaptiver Partikelfil-

ter bezeichnet [Müller, 2007]. Weiterführende Informationen dazu finden sich bei Arulampalam u. a. [2002] und Aggarwal u. a. [2010].

Auxiliary Partikelfilter (APF)

Ein Problem des Bootstrap Filters liegt in der Möglichkeit des Verfalls einer großen Anzahl von Partikeln. Dazu kommt es, wenn sich der geschätzte Zustand stark von den Beobachtungen unterscheidet. Daher werden nur wenige Partikel in einem Schritt mit einem starkem Gewicht belegt. Dies produziert beim Resampling eine Reproduktion nur der stark gewichteten Partikel. Beim APF wird dieses Problem mit dem Prinzip des Importance Sampling (IS) reduziert. Ziel ist es, eine größere Variation der Partikelgewichte zu erreichen [Marchetti u. a., 2009], damit mehr Partikel des Filters beim Resampling regeneriert werden. Das Importance Sampling erlaubt die Schätzung der Partikel unter Berücksichtigung des Einflusses der Beobachtungen [Aggarwal u. a., 2010].

Rao-Blackwellized Partikelfilter (RBPF)

Der Rao-Blackwellized Partikelfilter ist eine Variation des PF, der hauptsächlich für die Lösung von *Simultaneous Localization and Mapping* (SLAM) Anwendungen verwendet wird. Ziel ist die Schätzung alternativer hypothetischer Wege eines Roboters mit gleichzeitiger Kartierung der Umgebung [Schwarzlmüller, 2007]. Im RBPF besteht das Zustandsmodell aus einem linearen Kalman Filter. Die Anzahl notwendiger Partikel im Filter kann somit reduziert werden, da der Kalman Filter im Zustandsvektor eine definierte Anzahl an Zuständen repräsentieren kann [Aggarwal u. a., 2010]. Dies dient zudem einer guten Performance.

Backtracking Partikelfilter

Gerade bei der Positionsschätzung mit Kartendaten als Stützinformation können Wege leicht fehlinterpretiert werden. So führt beispielsweise eine Richtungsänderung während der Navigation bei nah aufeinander folgenden Abzweigungen aufgrund der Positionsunsicherheit zu einer fehlerhaften Trajektorie. Bei einer ausreichend großen Partikelwolke wird allerdings ein Teil der Partikelwolke auf dem real zurückgelegten Weg gestreut. Der aufgrund der Menge an Partikel vom Filter fehlerhaft favorisierte Weg kann zunächst weiterverfolgt werden. Dies führt zu einem Abbruch des Filters, da die Stützinformationen mit fortschreitender Zeit hohe Korrekturen liefern. Der Backtracking Partikelfilter ermöglicht nun das

Rückwärtsrechnen im Filter, indem vorherige Wolken gespeichert werden [Widya-
wan u. a., 2008]. Bei der Gebäudenavigation kann sich der Backtracking Filter an
Gabelungen aufteilen und verfolgt zunächst die wahrscheinlichsten Partikel. Im
Fortgang der Filterung wird diese Wolke unsicherer und divergiert aufgrund der
Korrekturen der Stützer. Nun kann das Filter aus den rückwirkend gespeicherten
Berechnungsschritten auswählen und einen neuen Weg fortführen.

2.2.3 Vergleich von Kalman Filter und Partikelfilter

Schwarzlmüller [2007] zufolge können die Unterschiede zwischen Kalman Filter
und Partikelfilter nach globalen Kriterien verglichen werden. Die Tab. 2.1 zeigt
eine Differenzierung nach Genauigkeit, Robustheit, Sensorvielfalt, Effizienz und
Implementierungsaufwand. Diese Kriterien variieren zwischen den Filtern je nach
Anwendung. In diesem Abschnitt soll nun ein grundlegender Unterschied erörtert
werden.

Tab. 2.1: Vergleich von Partikelfilter und Kalman Filter nach Basiskriterien (+ = gut, 0
= neutral, - = schlecht [Schwarzlmüller, 2007])

Kriterium	Kalman Filter	Partikelfilter
Darstellung	unimodal	diskret
Genauigkeit	+	+
Robustheit	0	+
Sensorvielfalt	0	+
Effizienz	-	0
Implementierung	0	+

Während das Kalman Filter auf einer unimodalen Verteilung (Normalverteilung)
beruht, kann das Partikelfilter jede Art von Verteilung darstellen. Diese muss sich
allerdings diskret in Form von Partikeln repräsentieren lassen. Die erreichbaren
Genauigkeiten variieren dabei je nach Anwendungsfall. Besonders bei solchen An-
wendungen, bei denen die Normalverteilung lediglich als Annäherung gelten kann,
leiden die Schätzungen des KF, da das Filter auf normalverteilte Zufallsgrößen
angewiesen ist. Im Prinzip können beide Filter als robuste Werkzeuge eingesetzt

werden. Besonders das auf die spezifische Aufgabe möglichst angenäherte Bewegungsmodell und die Berücksichtigung von Informationen lassen hier Spielraum zu.

Bei der Auswahl der Sensorik für einen bestimmten Anwendungsfall können Informationen wichtig werden, die nicht normalverteilt sind. In einem solchen Fall muss bei einem KF die Normalverteilung als Annäherung verwendet werden, was aber zu systematischen Fehlern führen kann. Beim Partikelfilter dagegen können diese Informationen individuell und in Abhängigkeit der Verteilung modelliert werden. Dies macht Partikelfilter gegenüber dem KF effizienter [Neumann, 2002]. Bei einem Einsatz von ausschließlich normalverteilten Zufallsgrößen zeigen die Ergebnisse von KF und PF jedoch keine Unterschiede.

Unter der Annahme, dass alle Einflussgrößen normalverteilt sind, wird in Tab. 2.1 der Implementierungsaufwand eines KF als etwas niedriger veranschlagt als beim PF. Insgesamt ist der PF in diesem Vergleich nach Schwarzmüller [2007] gegenüber dem KF vorzuziehen. Dies heißt aber nicht, dass der PF in jedem Fall das bessere Werkzeug wäre, da in Schwarzmüller [2007] und Neumann [2002] der Vergleich anwendungsorientiert für die Roboter Selbstlokalisierung stattfindet. Zudem finden sich in den Variationen des KF und des PF Lösungswege, die sich im Ansatz überschneiden. Beispiele für eine kombinierte Nutzung beider Filterarten sind das Sigma-Point Kalman Filter und das Rao-Blackwellized Partikelfilter, in denen jeweils Teilwerkzeuge beider Verfahren integriert sind.

2.3 Gebäudepläne und Routing-Netz

Gebäudepläne erfüllen bei einer Navigationsanwendung einen doppelten Zweck: Sie dienen der Visualisierung und sie sind für die Einbindung in den Prozess der Positionsschätzung sinnvoll. Je nach Kartenformat ist bei der Implementierung die Systemleistung des Smartphones zu beachten. Und je nach Anwendung kann das Kartenmaterial im Raster- oder im Vektorformat von Interesse sein. Ihr jeweiliges Format ist grundlegend für die Art der Verarbeitung der räumlichen Informationen.

Rasterkarten basieren im Gegensatz zu Vektorkarten auf Pixeln und die Bildauflösung ist daher nur bedingt skalierbar. Zwischen benachbarten Pixeln besteht kein Zusammenhang, von dem man beispielsweise Kanten (Wände) ableiten könnte. Um so etwas zu leisten, sind aufwendige Algorithmen notwendig. Zudem benötigt ihre Darstellung auf dem Display eines Smartphones vergleichsweise viel Rechenzeit. Eine interaktive Zuordnung unterschiedlicher Nutzungen – wie die Darstellung von Treppen und Wänden – ist nicht möglich. Das Format einer Rasterkarte entspricht dem Format der digitalen Fotografie (*.png, *.jpg, *.tiff etc.). Der Import in eine Applikation ist daher unproblematisch. Ihre Skalierbarkeit wird in der Regel durch Zuhilfenahme mehrerer, unterschiedlich hoch aufgelöster Rasterkarten umgesetzt. Jeder Zoomstufe ist eine Rasterkarte zugeordnet. Eine Klassifizierung von Inhalten ist mit der Farbgebung möglich.

Vektorkarten bieten die Möglichkeit einer relativ einfachen, Differenzierung von Objekten wie Wand, Raum, Tür und Treppe. Ihre Klassifizierung lässt sich über zusätzliche Attribute eines Objektes umsetzen. Typische Formate erlauben außerdem die Speicherung weiterer Eigenschaften zur Geometrie- und Farbdarstellung. Die Differenz zum Rasterformat liegt in der geometrischen Identifizierung von Objekten mit Attributen [Ehlers u. Schiewe, 2012]. Eine Linie wird hier nicht durch die Aneinanderreihung von Pixeln dargestellt, sondern ist durch ihren Anfangs- und Endpunkt definiert. Eigenschaften wie Linienstärke und Linienform lassen sich definieren. Eine Skalierung auch mit Anpassung der Objektdarstellung ist im Vergleich zum Rasterformat aufwendiger durchführbar, aber hochwertiger in der Auflösung. Anders als beim Rasterformat werden für jede Zoomstufe identische Daten verwendet und machen eine Generalisierung auf Basis der Klassifizierung möglich. Vektorkarten lassen sich in unterschiedliche Dateiformate (*.wkt, *.xml) überführen und mit geringem Programmieraufwand in eine Applikation einbinden. Zudem gibt es Programmbibliotheken und Module, die eine schnelle, unkomplizierte Einbettung ermöglichen.

Gebäudepläne als Visualisierungsgrundlage zur Navigation

Eine Karte eignet sich als Grundlage zur Visualisierung besonders gut, da die navigierende Person mit ihr die eigene Position in der Örtlichkeit abgleichen kann. Grobe Unsicherheiten der Positionsschätzung lassen sich so kompensieren. Die

berechnete Route wird in der Karte abgebildet. Dies gibt dem Nutzer ein Stück Selbstbestimmung gegenüber einer Navigation per Richtungspfeil, die zwar als Anweisung eingängig ist, aber als Orientierungshilfe innerhalb von Gebäuden an ihre Grenzen stößt. Zumal die Richtungsanzeige durch die Unsicherheiten der Positionsschätzung stark verfälscht sein kann.

Die Anzeige von Kartenmaterial auf dem Display stellt allerdings höhere Anforderungen an die Systemleistung des Smartphones. Das Format der Karte spielt hierbei eine untergeordnete Rolle, solange Darstellung und Positionsschätzung parallel berechnet werden können. Lediglich das Importieren in die Applikation und die Darstellung müssen möglich sein. Dazu könnte ein referenziertes Bild, beispielsweise eine Rastergrafik im *.png-Format, ausreichen.

Gebäudepläne als Stützinformation in der Positionsschätzung zur Indoornavigation

In den bislang diskutierten Filteralgorithmen (KF, PF) können Gebäudepläne als Stützinformation – für die Positionsschätzung, als Richtungs- oder Positionskorrektur – einfließen. Bei einer MEMS-basierten Positionsschätzung kann in einem schmalen Gang oder Flur die Richtungsinformation einer naheliegenden Wand zur Korrektur der aus dem Gyroskop bestimmten Richtung herangezogen werden. In Abb. 2.8 ist der Ausschnitt einer verwendeten Vektorkarte zu sehen. Die schwarzen Linien stammen aus einem CAD-Plan und stellen die Wände des Gebäudes dar. Sie können zur Stützung der Position herangezogen werden. Daher muss für die Positionsschätzung das Kartenformat eine einfach handhabbare Struktur aufweisen. Das spricht für das Vektorformat, da es einfache geometrische Rechenfunktionen wie den Geradenschnitt zulässt.

Die Erkennung einer möglichen Kollision ist auch mit Rasterkarten relativ einfach umzusetzen: schwarze Pixel beispielsweise, die eine Wand repräsentieren, können anhand des Farbwertes zugeordnet werden. Zur Stützung der Orientierung sind mehrere solcher Pixel notwendig, damit die Kanten bzw. Wände per Algorithmus detektiert werden können und sich eine Richtung berechnen lässt.

Für den Einsatz der Karte zur Korrektur der Positionsschätzung wird in dieser Arbeit das Vektorformat bevorzugt. Die daraus abzuleitenden Stützinformationen

sind für die Indoor-Navigation besser geeignet als die Rasterdaten. Sie lassen sich einfach und schnell in das xml-Format transformieren.

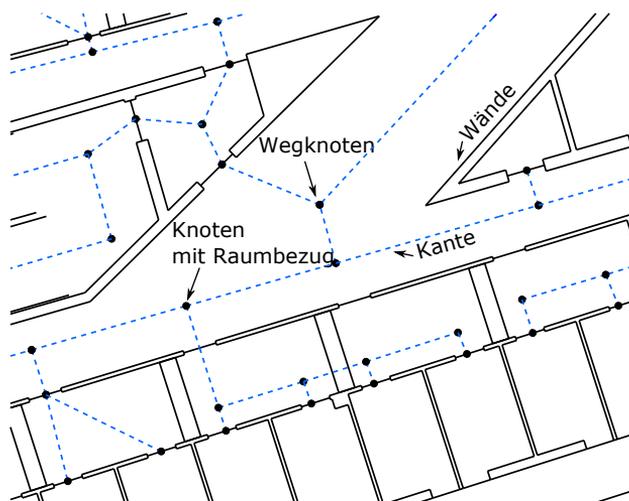


Abb. 2.8: Ausschnitt des vierten Obergeschosses aus dem Gebäudeplan der HCU mit Routing-Graph (blau gestrichelt: Kanten des Routing-Netz, schwarz: Wände und Türen)

Erstellung des Routing-Netzes

Das Routing ist ein bekanntes Verfahren zur Berechnung der kürzesten, schnellsten oder ökonomischsten Verbindung zwischen zwei Punkten. Bei einer Navigation ist das die Verbindung zwischen der aktuellen Position und dem Zielpunkt, und zwar über je nach Fahrzeug nutzbare Verkehrswege. Voraussetzung ist daher ein vorhandenes, topologisch erfasstes Wegenetz, das aus Kanten und Knoten besteht [Blankenbach, 2007] (Abb. 2.8).

Auf Basis dieses Kanten-Knoten-Netzes im Vektorformat erfolgt die Wegeberechnung über einen Algorithmus. Als Beispiele seien Algorithmen wie Dijkstra, A*, Floyd-Warshall und Bellmann-Ford genannt. Sie alle basieren auf der Graphentheorie [Turau, 2009], die eine Berechnung beinhaltet, die auf den Kosten der Kanten basiert. Die Kosten auf der Kante können für die Länge der Kante, die Kosten des Energieverbrauchs oder die Zeit stehen, die für das Zurücklegen einer Wegstrecke erforderlich ist. Die Umsetzung zweier Routing-Algorithmen zum

Indoor-Einsatz erfolgt zum Abschluss dieser Arbeit in Kapitel 6.1.1 *Routenberechnung* am Beispiel des HCU-Gebäudes.

Für die Navigation ist ein Routing-Netz zwingend erforderlich, daher erfordert die Verwendung in der Positionsschätzung keinen zusätzlichem Aufwand in der Implementierung im Gebäude. Das Kanten- und Knotenmodell spiegelt die typischen Laufwege in einem Gebäude wider und eignet sich zur Stützung der Positionsschätzung. Allerdings spiegelt das Routing-Netz nicht unbedingt die zurückgelegten Strecken exakt wieder, wie es beispielsweise in Abb. 2.8 am zentralen Wegknoten sichtbar ist. Diese Abweichung liegt an der starken Generalisierung des Netzes. Wenn das Routing-Netz zu Stützungszwecken eingesetzt wird, muss dieser Effekt kompensiert werden, um zusätzliche Unsicherheiten aufgrund fehlerhafter Korrekturen zu vermeiden.

Das Routing-Netz für das HCU-Gebäude wird auf Grundlage von CAD-Daten manuell erstellt. Dabei wurden die kürzesten Verbindungen zwischen Räumen ausgewählt und die Kanten weitgehend zentriert in den Gängen positioniert. Für den Einsatz in der Positionsschätzung wird eine Textdatei, in der in jeder Zeile die Anfangs- und Endkoordinaten der Kante enthalten sind, erstellt. Jede dieser Koordinaten repräsentiert einen Knoten im Routing-Netz.

2.4 Graphentheorie und Routingalgorithmen

Algorithmen für das Routing lassen sich der Gruppe der Greedy-Algorithmen zuordnen. Greedy-Algorithmen – zu Deutsch: gierige Algorithmen – stehen für die Suche nach einer Gesamtlösung durch die iterative Erweiterung einer lokalen Lösungssuche. Dazu werden parallel Teillösungen weiterentwickelt. Greedy-Algorithmen sind vergleichsweise einfach zu entwerfen und können recht effizient in Applikationen implementiert werden [Turau, 2009]. Aufgrund der iterativ fortschreitenden Arbeitsweise sind jedoch die Berechnungen mit ihnen aufwendig. Zur Effizienzsteigerung gibt es je nach Ansatz unterschiedliche Optimierungsmöglichkeiten, beispielsweise die Anwendung von Annahmen zur Laufzeitoptimierung.

Grundlage der Greedy-Algorithmen bildet die Graphentheorie. Ein Graph ist definiert durch n -Knoten, die durch Kanten verbunden sind. Einsatzgebiete für Gra-

phen sind Kommunikationsnetze, Wegplanungen, Suchmaschinen und objektorientierte Programmiersprachen. Für die Umsetzung eines Routings können ungerichtete und gerichtete Graphen Einsatz finden. Bei gerichteten Graphen besitzen die Kanten eine festgeschriebene Bewegungsrichtung. Es können aber auch Mischungen aus gerichteten und ungerichteten Kanten im Routing umgesetzt werden [Lederer, 2009]. Im Routing werden die Kanten nach ihren Kosten gewichtet, diese können beispielsweise im Zeitaufwand oder der Entfernung bestehen.

Gegenüber der Straßennavigation ist die Umsetzung eines Routings im Gebäude weniger kompliziert, da hier weniger Datensätze vorliegen. Zudem ist die Berücksichtigung unterschiedlicher Geschwindigkeiten für Fußgänger im Gebäude nicht notwendig. Insofern können die Kosten für die Routenberechnung dem Aufwand zur Überwindung der vorhandenen Infrastruktur, üblicherweise Flure, Fahrstühle und Treppen, gleichgesetzt werden.

Für die Navigation im HCU-Gebäude werden zwei Algorithmen untersucht: der Dijkstra und der Floyd-Warshall Algorithmus. Beide eignen sich zur Routenberechnung in Gebäuden, haben allerdings eine unterschiedliche Vorgehensweise zur Routenberechnung. Der Dijkstra Algorithmus setzt positive Kantenbewegungen voraus.

Ein Beispiel eines ungerichteten Wegepunktenetzes, wie es für die HCU aussehen könnte, zeigt Abb. 2.9. Die Knoten repräsentieren Weggabelungen und Kreuzungen in dem Gebäude. Die Kanten stehen für Gänge ohne Abzweigungen. Damit eine Verknüpfung der Stockwerke stattfinden kann, müssen die Stockwerke bei Treppen und Fahrstühlen mit Kanten verbunden werden, wie beispielsweise Knoten 2.1 und 2.2 demonstrieren. In diesem Beispiel weist die Nachkommastelle der Knotenbezeichnung auf die Möglichkeit eines Stockwerkswechsels hin. Gesucht ist die kürzeste Verbindung zwischen Knoten 1 und Knoten 8.

Die nach der Routenberechnung kürzeste Verbindung ist mit 1 - 2.1 - 2.2 - 8 gefunden. Abb. 2.10 zeigt die Routenfindung am Beispiel des Dijkstra Algorithmus. Beim Dijkstra Algorithmus werden zunächst die Kosten von der Startposition zu den Nachbarknoten berechnet. Anschließend wird in jedem folgenden Berechnungsschritt nur der günstigste Teilweg fortgeführt. Durch die Übernahme der zuvor nicht fortgeführten Knoten kann in jedem Berechnungsschritt zudem die

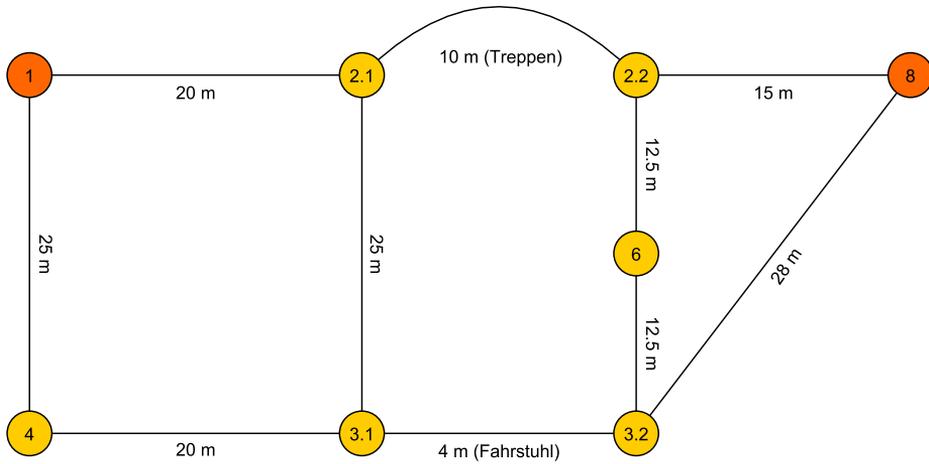


Abb. 2.9: Beispiel-Graph als Grundlage für einen Routingalgorithmus. Kosten sind Entfernungen, Stockwerksübergänge werden durch Kommazahlen repräsentiert [Willemsen u. a., 2014a].

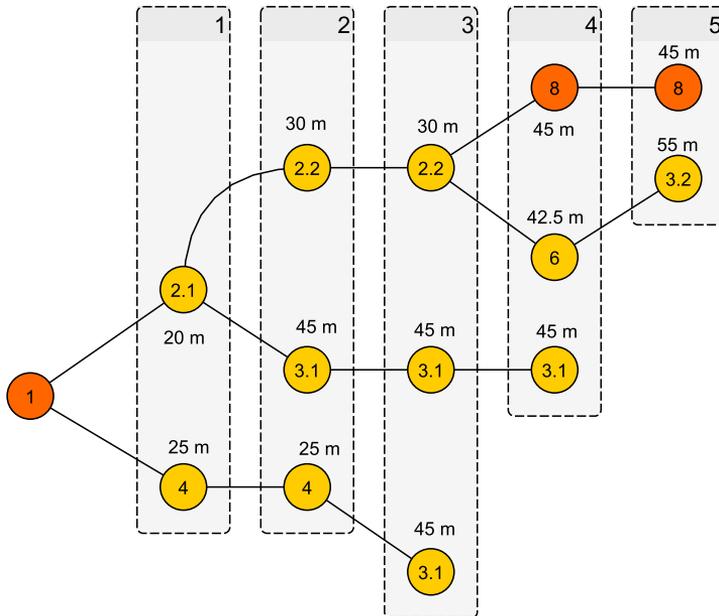


Abb. 2.10: Berechnungsschritte eines Dijkstra Algorithmus angewendet auf den Beispiel-Graph aus Abb. 2.9 [Willemsen u. a., 2014a].

Auswahl eines vorher ausgeschlossenen Knotens erfolgen. Die Berechnungszeit für die Zielführung hängt maßgeblich von der Anzahl der Knoten ab, da üblicherweise auch die mögliche Anzahl an Verbindungen mit steigender Knotenanzahl ansteigt. Für den einheitlichen Vergleich der Laufzeit zwischen Algorithmen kann die O-Notation Anwendung finden, die in Abhängigkeit der Eingaben n die maximal notwendige Anzahl an Arbeitsschritten aufführt [Turau, 2009]. Nach O-Notation beträgt die Laufzeit: $O(|n^2|)$.

Der Floyd-Warshall Algorithmus ist eine Erweiterung des Warshall-Algorithmus und verwendet eine erweiterte Adjazenzmatrix, auch als Pfadmatrix bezeichnet. Während beim Dijkstra Algorithmus für jede Wegeberechnung die Daten aufbereitet und abgefragt werden müssen, wird beim Floyd-Warshall Algorithmus eine Berechnung aller Wegemöglichkeiten vor einem Einsatz zur Routenberechnung in der Pfadmatrix gespeichert. Auf der Anwendungsplattform werden nur noch Abfragen mit Hilfe der erweiterten Adjazenzmatrix durchgeführt. Eine regelmäßige aufwendige Berechnung während der Navigation entfällt und macht den Algorithmus besonders für große Netze interessant. Von Nachteil ist die notwendige Neuberechnung der Pfadmatrix nach Änderung in der Infrastruktur des Gebäudes [Turau, 2009]. Die Laufzeit des gesamten Algorithmus beträgt: $O(|n^3|)$.

Neben den beschriebenen Algorithmen existieren weitere Algorithmen zur Routenfindung, beispielsweise der A*-Algorithmus. Er ist eine Modifikation des Dijkstra Algorithmus und bestimmt den kürzesten Weg zwischen zwei Punkten vergleichsweise schnell. Ziel ist es, die Anzahl der besuchten Knoten möglichst gering zu halten [Turau, 2009]. Dazu wird eine Schätzfunktion $f(i)$ in den Algorithmus eingebunden. Die Schätzfunktion kann beispielsweise die direkte Verbindung zwischen Start- und Ziel-Punkt sein und dient zur Gewichtung einzelner Knoten im aktuellen Berechnungsschritt. Die Entscheidungsgrundlage basiert somit nicht mehr nur auf den aktuellen aufsummierten Kosten.

2.5 Versuchshardware

Für die Algorithmenentwicklung zur smartphonebasierten Positionsschätzung unter Innenraumbedingungen wurden ein Samsung Galaxy Nexus und ein Google

Nexus 4 als Versuchsgeräte ausgewählt. Beide Smartphones werden mit dem Betriebssystem Android betrieben. Dies macht das Erstellen eigener Applikationen (Apps) vergleichsweise einfach, da Android eine offene Programmierschnittstelle bietet. Im Folgenden werden die Smartphones und ihre für diese Arbeit interessanten Sensoren detaillierter vorgestellt. Dazu werden zufällig ausgewählte Rohdaten gezeigt, die zwar als Einzelergebnisse anzusehen sind, aber den typischen Eigenschaften des Sensors, wie beispielsweise Messrauschen und der Auflösung entsprechen.

2.5.1 Smartphones mit integrierten MEMS

Samsung Galaxy Nexus

Die Vermarktung des Samsung Galaxy Nexus (Abb. 2.11) startete im 4. Quartal 2011. Der Einführungspreis liegt bei 679 € und ist daher eher den Smartphones im höheren Preissegment zuzuordnen. Das Smartphone enthält einen Dual Core Prozessor (1,2 Ghz) und 1 Gb Arbeitsspeicher [Samsung, 2011]. Es wurde für diese Arbeit ausgewählt, da es zusätzlich zu den anderen notwendigen Sensoren erstmals auch einen Barometer hat, der sich zur Stockwerkserkennung eignen könnte.



Abb. 2.11: Testgerät Samsung Galaxy Nexus, Marktverfügbarkeit 2011 (<http://www.pcmweb.nl/nieuws/samsung-galaxy-nexus-met-ice-cream-sandwich.html>).

Für eine intuitive Handhabung und möglichst viele Anwendungsmöglichkeiten ist eine große Anzahl von Sensoren in modernen Smartphones verbaut. So wird z.B. ein Beschleunigungssensor zur Erkennung der Ausrichtung und Drehung des Bild-

Tab. 2.2: Für die Positionsschätzung interessante MEMS-Sensoren im Smartphone Samsung Galaxy Nexus

Sensortyp	Name	Hersteller
3-Achs Gyroskop	MPU-3050	InvenSense
3-Achs Beschleunigungsmesser	BMA 250	Bosch
Barometer	BMP 180	Bosch
3-Achs Magnetfeldsensor	YAS 530	Yamaha
Bluetooth, GNSS, WLAN, Kamera		

schirms benötigt. Ein Abstands- oder Annäherungssensor erkennt Objekte vor dem Display, um das Display während eines Telefonats zu verdunkeln um so den Energieverbrauch zu senken. Viele dieser Sensoren sind in MEMS-Bauweise gefertigt. Für die MEMS-INS-basierte Positionsschätzung sind besonders Beschleunigungsmesser, Gyroskope, Magnetfeldsensoren und Barometer von Interesse. Ein Gyroskop registriert Winkelgeschwindigkeiten und wird etwa bei Spielen verwendet. Der Magnetfeldsensor wird in vielen Applikationen als Kompass oder, wie auch das Barometer, für das Assisted-GPS zur schnelleren Initialisierung einer GNSS-Position eingesetzt. Tab. 2.2 zeigt die im Samsung Galaxy Nexus verbauten Sensoren.

Die vier primär ausgewählten Sensoren stammen von drei unterschiedlichen Herstellern. Nur Bosch ist mit dem Barometer und dem Beschleunigungsmesser zweimal vertreten. Die Steuerung der Sensoren wird vom Betriebssystem übernommen. Für infrastrukturbasierte Anwendungen sind auch Bluetooth und WLAN interessant. Die Kamera im Smartphone kann ebenfalls je nach Anwendung zur Positionsschätzung beitragen.

LG E960 / Google Nexus 4

Als zweites Versuchsgerät wird das Google Nexus 4 (Abb. 2.12) herangezogen, das seit 2012 auf dem Markt ist. Während Google beim Galaxy Nexus mit Samsung kooperierte, ist das Gerät Nexus 4 Produkt einer Kooperation des Internetriesen mit dem südkoreanischen Unternehmen LG. Mit 299 € Einführungspreis ist das Smartphone dem mittleren Preissegment zuzuordnen. Es ist mit einem Quad-Core-Prozessor, 2 GB Arbeitsspeicher und den ebenfalls im Galaxy Nexus

vorhandenen Sensortypen ausgestattet und ist, trotz des niedrigen Preises, mit Smartphones des Premiumsegments vergleichbar [LG, 2016].

Der günstige Marktpreis und die Verfügbarkeit aller notwendigen Sensoren waren für die Wahl ausschlaggebend. Es erlaubt zudem die Überprüfung der neueren Generation von Sensoren gegenüber dem Galaxy Nexus hinsichtlich der Datenqualität. Die im Testgerät Google Nexus 4 verbauten Sensoren sind in Tab. 2.3 aufgeführt.



Abb. 2.12: Testgerät LG E960 verfügbar als Google Nexus 4, Markteinführung 2012 [LG, 2016].

Tab. 2.3: Für die Positionsschätzung interessante MEMS-Sensoren im Google Nexus 4 / LG E960

Sensortyp	Name	Hersteller
3-Achs Gyroskop	MPU-6050	InvenSense
3-Achs Beschleunigungsmesser		
Barometer	BMP 180	Bosch
3-Achs Magnetfeldsensor	AK 8963	AKM
Bluetooth, GNSS, WLAN, Kamera		

Der Beschleunigungsmesser und das Gyroskop befinden sich im Vergleich zum Samsung Galaxy Nexus in einem Modul und stehen für eine neue Sensorgeneration des Herstellers InvenSense. In beiden Smartphones findet sich das Barometer BMP180 von Bosch.

Koordinatensystem

Gyroskop, Beschleunigungsmesser und Magnetfeldsensor sind als dreiachsige Sensoren im Smartphone verbaut. In Android-betriebenen Smartphones sind sie durch ein festgelegtes Koordinatensystem definiert (Abb. 2.13). Dies ist notwendig, damit alle Applikationen Smartphone-übergreifend funktionieren können. Die y-Achse zeigt in Längsrichtung nach oben aus dem Smartphone, die z-Achse steigt positiv aus dem Display heraus, somit verläuft die xy-Ebene parallel zum Display.

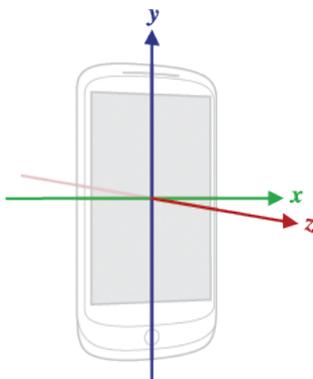


Abb. 2.13: Koordinatensystem von Android Smartphones (http://developer.android.com/images/axis_device.png).

Aufgrund der Kompaktheit und des Designs von Smartphones sind die Sensoren je nach Sensormodell an unterschiedlichen Stellen verbaut. Sie sind nicht im Koordinatenursprung zu finden, was ein Blick auf die Hauptplatine des Nexus 4 in Abb. 2.14 verdeutlicht. Das gelbe Rechteck in der Abbildung identifiziert den MPU-6050-Baustein, in dem Beschleunigungsmesser und Gyroskop verbaut sind. Im unteren Bereich im silbernen Kästchen befindet sich das Barometer [www.ifixit.com, 2015]. Im Verlauf der vorliegenden Untersuchung wird dieser Nachteil bei der Kalibrierung des Beschleunigungsmessers deutlich.

MEMS beider Testgeräte im Vergleich

Für eine erste Einschätzung der in beiden Smartphones verbauten Sensoren werden die Messdaten einer Langzeitmessung in Ruhelage vorgestellt. Ziel ist eine Einschätzung der Rohdatenqualität beider Sensorgenerationen und der Unterschiede beider Android-basierten Geräte. Dazu wurde eine Applikation entwickelt, die alle Messdaten in Echtzeit in der jeweils höchsten Messrate speichert und in einer

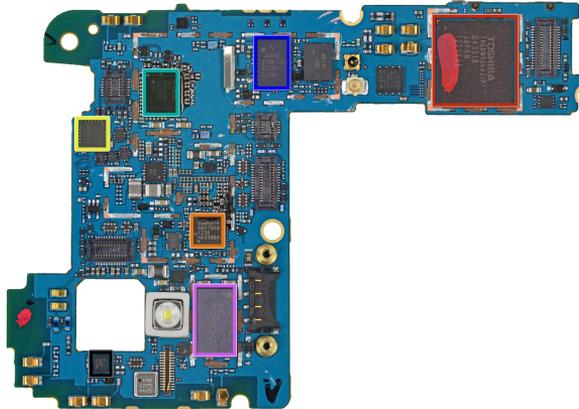


Abb. 2.14: Sensorenübersicht der Hauptplatine im Google Nexus 4. Die markierten Rechtecke fokussieren verschiedene Sensoren, beispielsweise in Gelb die der MPU-6050 (Quelle: <https://www.ifixit.com/Teardown/Nexus+4+Teardown/11781>).

Textdatei auf dem Smartphone hinterlegt. Im Rahmen der vorliegenden Arbeit dient die Applikation außerdem dazu, um die gewünschten Algorithmen mit den gewonnenen Daten in Matlab, einer plattformunabhängigen Software zur Lösung und Darstellung mathematisch-technischer Probleme, zu entwickeln.

Gyroskop

Für einen Vergleich der dreiachsigen Gyroskope der Testgeräte wurden die ersten 100 Sekunden in Ruhelage aufgezeichnet. Abb. 2.15 und Abb. 2.16 zeigen die Drehraten des jeweiligen Smartphones. Die Darstellung der Achsen erfolgt nach dem RGB-Farbschema (Rot = X, Grün = Y, Blau = Z). Während das Galaxy Nexus die Messdaten mit 100 Hz registriert, arbeitet der Sensor im Nexus 4 mit 200 Hz. Beide Gyroskope stammen von InvenSense Inc., dem führenden Hersteller MEMS-basierter Bewegungssensoren. Der Sensor MPU-6050 im Nexus 4 ist Nachfolger vom MPU-3050 im Galaxy Nexus. Mittlerweile sind zwei weitere Nachfolgemodelle auf dem Markt erhältlich, der MPU-9150, der über einen zusätzlich integrierten Magnetfeldsensor verfügt, und dessen Nachfolger, der MPU-9250 (Stand: Feb. 2016).

Die Abbildungen zeigen bei beiden Geräten einen Korrekturingriff, in Abb. 2.15

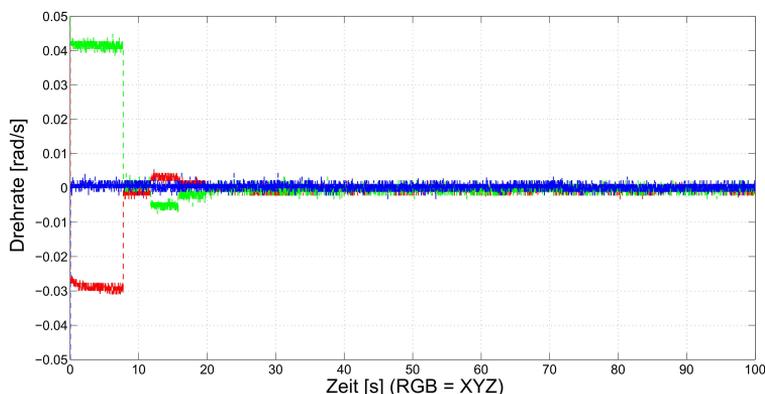


Abb. 2.15: Drehraten des Gyroskops im Samsung Galaxy Nexus, sichtbar sind die ersten Sekunden nach dem Start (Achsenzuordnung: RGB = XYZ).

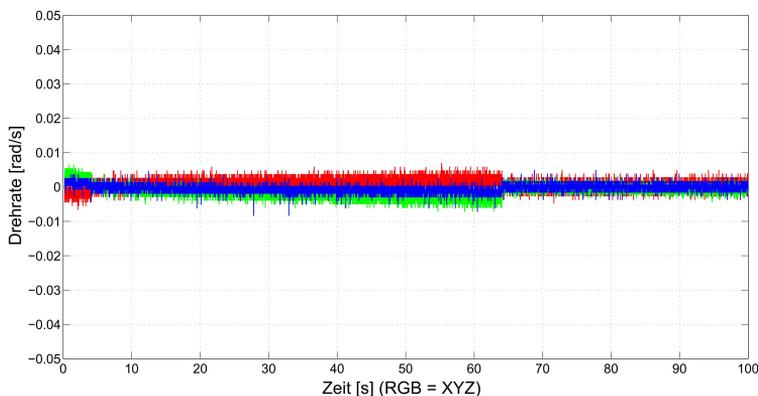


Abb. 2.16: Drehraten des Gyroskops im Google Nexus 4, sichtbar sind die ersten Sekunden nach dem Start (Achsenzuordnung: RGB = XYZ).

erstmalig nach ca. 8 Sekunden und in Abb. 2.16 erstmalig nach ca. 6 Sekunden. Diese Korrektur wird durch einen so genannten ZUPT-Befehl (Zero Velocity UPdaTe) hervorgerufen, der in einer Ruhelage nach einer gewissen Zeitspanne die detektierte Drehrate auf null zurücksetzt. Drehraten gleicher Größe, die über einen längeren Zeitraum stabil gehalten werden, werden vom Betriebssystem als Offset-Fehler interpretiert und per ZUPT korrigiert.

Während sich beim Galaxy Nexus ein Fehler unmittelbar in den Messdaten nie-

derschlägt (siehe Zeitraum 0-20 Sekunden), ist bei der MPU-6050 vom Nexus 4 ein stetiger Anstieg zu verzeichnen, bis ein Grenzwert erreicht wird und eine Korrektur stattfindet. Bei genauerer Betrachtung fällt das höhere Messrauschen der MPU-6050 auf. Zurückzuführen ist es auf die höhere Abtastrate. Laut Datenblatt des Sensorherstellers InvenSense [InvenSense, 2015] kann der MPU-6050 mit bis zu 1.000 Hz Ausgaberate programmiert werden. Die vergleichsweise geringe Abtastung des Sensors im Smartphone könnte für die Verringerung des Leistungs- und Energieverbrauchs oder aufgrund der geplanten Anwendungen, die keine höhere Abtastraten benötigen, vom Betriebssystem begrenzt sein.

Beschleunigungsmesser

Zum Vergleich der Beschleunigungsmesser sind in Abb. 2.17 die Daten der jeweiligen x-Achsen beider verbauter Sensoren über einen Zeitraum von 800 Sekunden wiedergegeben. Es sei an dieser Stelle darauf hingewiesen, dass der Vergleich beider Sensoren nicht als Vergleich einer Produktreihe missverstanden werden darf, denn im Galaxy Nexus ist der Beschleunigungsmesser BMA 250 der Firma Bosch eingebaut, während im Nexus 4 die MPU-6050 von InvenSense Beschleunigungsmesser und Gyroskop kombiniert. In der Abbildung sind die Messergebnisse des Galaxy Nexus grün dargestellt. Sie weisen gegenüber den Daten vom Nexus 4 – im Schaubild: schwarz – ein deutlich höheres Messrauschen auf. Außerdem ist eine leichte Drift im Datensatz des Galaxy Nexus erkennbar. Die Vermutung liegt nahe, dass die integrierte Temperaturmessung im MPU-6050 die Drift aufgrund der Aufwärmung des Smartphones minimieren kann. Der MPU-6050 weist eine relativ hohe Stabilität bei geringerem Rauschen auf, obwohl die Abtastrate für den Sensor im Nexus 4 bei 200 Hz liegt, während der Sensor im Galaxy Nexus mit nur 100 Hz arbeitet.

Barometer

Mit Hilfe der barometrischen Höhenformel können metrische Höhen aus Luftdruckmessungen bestimmt werden. Formel (23) zeigt die mit Werten gefüllte aufgelöste Grundgleichung aus Kahmen [2006] zur Berechnung von Höhen relativ zum mittleren Meeresspiegelniveau ohne Temperaturkorrektur. Der Luftdrucksensor BMP-180 hat, den Angaben des Herstellers zufolge [Sensortec, 2016], eine Genauigkeit von 1 hPa in einem Messbereich von 300-1100 hPa. Dies entspricht

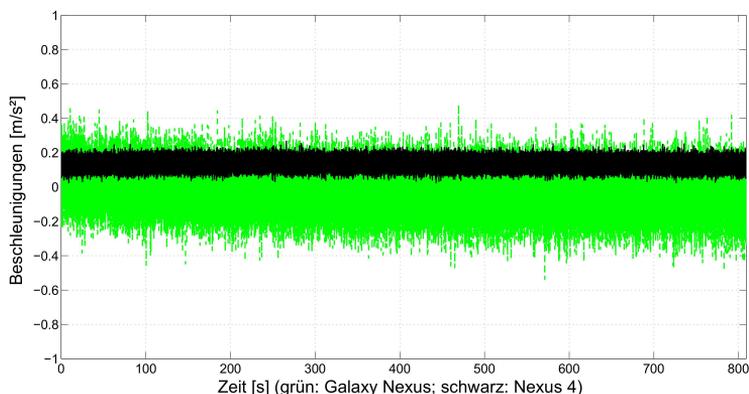


Abb. 2.17: Rohdaten der x-Achsen der Beschleunigungsmesser in beiden Testgeräten einer gleichzeitigen Messung über einen Zeitraum von 800 Sekunden (schwarz: Nexus 4; grün: Galaxy Nexus).

einer Höhengenaugigkeit von ca. 8 m. Je nach äußeren Einflüssen sind, so die Herstellerangaben, Maximalabweichungen von bis zu 4,5 hPa möglich. Um dennoch Stockwerke in Gebäuden unterscheiden zu können, werden nur relative Messwerte mit dem Barometer verarbeitet. Der Hersteller gibt dafür eine Genauigkeit von $\pm 0,12$ hPa (1 m) an.

In Abb. 2.18 werden Messergebnisse beider Smartphones in Ruhelage für 800 Sekunden gezeigt. Die Luftdruckmessungen sind in Meter umgerechnet und von der zuvor berechneten ersten Höhe subtrahiert worden. Das Galaxy Nexus registriert Messdaten mit ca. 10 Hz, während das Nexus 4 mit ca. 30 Hz Daten aufzeichnet. In den Standardeinstellungen [Sensortec, 2016] kann der Sensor nach 5 ms einen Wert für den Luftdruck ausgeben – dies würde einer möglichen Messrate von maximal 200 Hz entsprechen. Da der baugleiche Sensor in beiden Smartphones mit unterschiedlichen Abtastraten Sensordaten ausgibt, wurden zu besseren Vergleichbarkeit in Abb. 2.18 die Daten jeweils zu 1 Hz geglättet. Beide Datensätze (grün: Galaxy Nexus, schwarz: Nexus 4) zeigen ein ähnliches Rauschverhalten im Bereich von ± 1 m Streuung. Allerdings steigt die Differenz zwischen beiden ermittelten Höhen zum Ende der Datenreihe an und überschreitet 1 m.

$$h_b = \frac{288,15 - 288,15 * 5,255 \sqrt{\frac{P(h)}{1013,25}}}{0.0065} \quad (23)$$

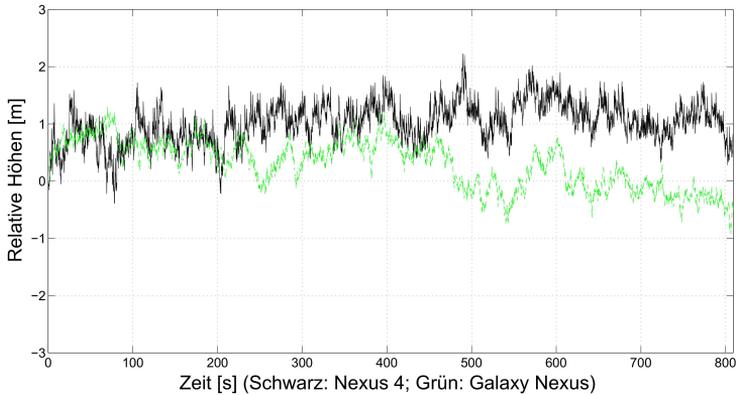


Abb. 2.18: Relative Höhen abgeleitet aus Luftdruckmessungen für eine Datenaufzeichnung von 800 Sekunden Dauer (schwarz: Nexus 4; grün: Galaxy Nexus).

Magnetfeldsensor

Die x-Achse beider Magnetfeldsensoren wird in Abb. 2.19 über einen Zeitraum von 800 Sekunden (grün = Galaxy Nexus und schwarz = Nexus 4) dargestellt. In der x-Achse zeigt sich beim Sensor des Galaxy Nexus eine deutliche Drift. Sie wird vermutlich durch die Erwärmung des Smartphones hervorgerufen. Beide Smartphones waren beim Test bestmöglich von Störeinflüssen durch äußere Magnetfelder abgeschirmt. Nur eine Einwirkung beider Smartphones aufeinander ist nicht auszuschließen, da die Messungen zeitgleich an derselben Position durchgeführt wurden, um äußere Einflüsse gleichzuschalten. Die Drift des YAS530 (Galaxy Nexus) beträgt in diesem Beispieldatensatz nach 800 Sekunden $5 \mu T$. Die Referenzwerte für das Erdmagnetfeld in Hamburg liegen räumlich bei ca. $50 \mu T$ und für die Lage (Nordsuche) bei $20 \mu T$ [Willemsen u. a., 2014c]. Der mögliche Messbereich liegt bei $\pm 4900 \mu T$ für den AK8963 von AKM (im Nexus 4) und für den YAS530 von Yamaha bei $\pm 800 \mu T$ [Yamaha, 2016] [AKM, 2016].

Magnetfeldsensoren dienen oft der Bestimmung der Richtung zu magnetisch Nord. Auf dem Markt gibt es dazu eine Vielzahl unterschiedlicher Kompass-Applikationen. Für die Positionsschätzung per Inertialsensoren sind absolute Richtungsmessun-

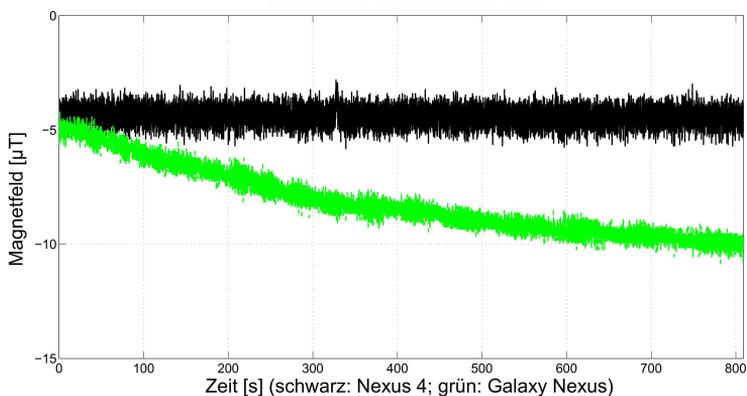


Abb. 2.19: Rohdaten der x-Achsen beider Magnetfeldsensoren der Testgeräte über einen Zeitraum von 800 Sekunden (schwarz: Nexus 4; grün: Galaxy Nexus).

gen zur Korrektur einer aus Drehraten berechneten Richtung aufgrund der Fehlercharakteristika eines Gyroskops besonders geeignet. Zur Bestimmung der Nordrichtung muss der dreiaxige Magnetfeldsensor zuerst durch den Beschleunigungssensor in den Horizont transformiert werden, sodass die xy-Ebene des Sensors ausschließlich die Nordkomponenten des Magnetfeldes detektiert. Anschließend kann aus der x- und y-Achse des Sensors die magnetische Nordrichtung berechnet werden. Abb. 2.20 zeigt die aus dem Datensatz in Ruhelage errechnete magnetische Nord-Richtung. Die Daten des Galaxy Nexus (grün) weisen ein erhöhtes Rauschen im Vergleich zu den Rohdaten der x-Achse (Abb. 2.19) auf. Grund ist der Beschleunigungsmesser. Dessen Daten sind stark verrauscht und beeinflussen durch die Berechnung der Horizontierung des Magnetfeldsensors die Richtung zu magnetisch Nord. Der Einfluss der Drift auf der x-Achse des Galaxy Nexus ist weiterhin klar erkennbar und wirkt sich mit einem Winkelfehler von $0,2 \text{ rad}$ (12°) aus. Der Einsatz von Magnetfeldsensoren zur Bestimmung der Nordrichtung in Gebäuden ist daher besonders wegen der dort künstlich erzeugten Magnetfelder problematisch.

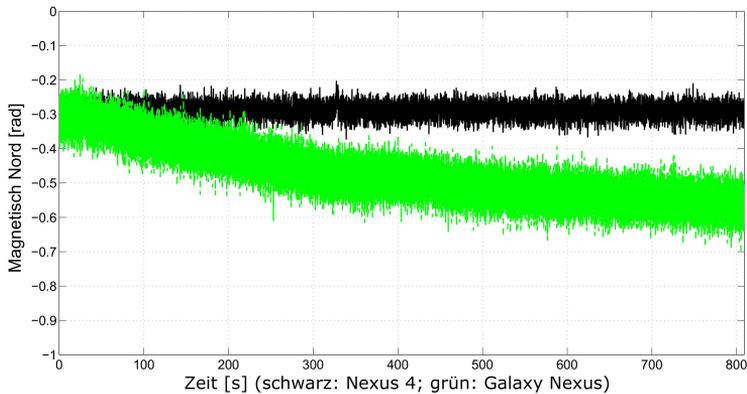


Abb. 2.20: Magnetisch Nord, berechnet aus den Rohdaten in Abb. 2.19 (schwarz: Nexus 4; grün: Galaxy Nexus).

2.5.2 Android als Entwicklungsumgebung

Im November 2007 wurde die erste Version des Android Software Development Kits (Android SDK) vorgestellt. Dies war der Auftakt der freien Entwicklung von Anwendungen für Hardware auf Basis des Betriebssystems Android. Gleichzeitig entstand die erste offene Plattform für mobile Geräte. Die Open Handset Alliance (OHA) verfolgte unter Leitung von Google mit Android das Ziel einer kostengünstigen Entwicklung und Verteilung innovativer Produkte sowie die rasche Markteinführung auf mobilen Geräten [Künneht, 2012].

Android ist ein Betriebssystem, das auf der Programmiersprache Java aufgebaut ist. Mit Hilfe der Android-SDK und einer Entwicklungsumgebung wie Eclipse lassen sich Applikationen für Smartphones entwickeln. Andere verbreitete Betriebssysteme von Smartphones sind das von Microsoft entwickelte Windows Phone und das mobile Betriebssystem iOS von Apple. Ein Nachteil dieser Betriebssysteme ist der für Entwickler vergleichsweise höhere Aufwand, um mit der Entwicklung einer App starten zu können. Samsungs inzwischen eingestellte Eigenentwicklung Bada, das BlackBerry-OS, und die ebenfalls vom Markt genommene Symbian-Plattform von Nokia sind weitere Betriebssysteme für Smartphones, die hier der Vollständigkeit halber genannt seien.

Die große Community von Entwicklern, die überwältigende Akzeptanz und Markt-

dominanz – 75 % aller Smartphones laufen mit Android [IDC, 2013]) – und die gegenüber anderen Smartphone-Betriebssystemen relativ einfache Umsetzung eigener Applikationen sind der Grund dafür, dass in der vorliegenden Untersuchung Android-betriebene Smartphones Verwendung finden.

Die Untersuchungen dieser Arbeit basieren weitestgehend auf einem Post Processing mit Matlab-Programmen und real erzeugter Messdaten. Daher wird eine Applikation zur Rohdatenspeicherung von Beschleunigungssensor, Gyroskop, Barometer, Magnetfeldsensor und WLAN Daten entwickelt. Die Rohdaten werden mit einem Zeitstempel markiert, damit im Post Processing eine möglichst echtzeitorientierte Auswertung erfolgen kann. Weitere Applikationen entstehen je nach Stand der Entwicklung, um die Performance entwickelter Algorithmen auf den Testgeräten zu untersuchen.

2.5.3 Arbeiten auf Sensorebene am Beispiel von Arduino

Die Möglichkeiten der Einsetzbarkeit von MEMS-INS sind vielfältig. Viele Projekte sind allerdings aufgrund spezieller Anforderungen und der Notwendigkeit flexibler Schnittstellen auf Eigenentwicklungen angewiesen. In Sternberg u. a. [2015] werden einige Beispiele genannt: ein Einsatz zur Steuerung von Audio- und Videosequenzen in einem Kunstprojekt, umgesetzt mit den hier vorstellten Sensoren und einer Bluetooth-basierten Kommunikation, sowie ein sich per Schrittmotor selbst ausrichtender Reflektor.

Da die abgreifbaren Sensordaten in den Testgeräten durch das Betriebssystem des Smartphones vorverarbeitet werden, wie beispielsweise der ZUPT am Gyroskop, ist es notwendig, eine eigene Sensorplattform zu entwickeln, um die Sensoren rein auf der Hardwareebene zu untersuchen. Dazu werden die Sensortypen im Nexus 4 in einer Eigenentwicklung mit der Physical-Computing-Plattform Arduino kombiniert.

Auf dem Markt existieren unterschiedliche Plattformen für Physical Computing, die es erlauben, Sensoren anzuschließen und Sensordaten zu verarbeiten. Beispiele sind: .NET Micro Framework, Arduino, .NET Gadgeteer, Raspberry Pi, pcDduino und Tinkerforge. Die bekanntesten sind Arduino und Raspberry Pi [Keller u.

Willemsen, 2014]. Diese Sensoren erfreuen sich einer großen Beliebtheit bei Modellbauern, Künstlern und Bastlern. Während Raspberry Pi ein eigenes, Linux-basiertes Betriebssystem besitzt, wird bei Arduino eine vereinfachte c-basierte Entwicklungsumgebung verwendet, um Funktionen an den Arduino-Mikroprozessor zu übertragen. Die Arduino-Technologie wird daher auch als Cross-Plattform Entwicklungsumgebung bezeichnet. Diese Mikrocontroller haben analoge und digitale Ein- und Ausgänge, an denen man die Sensoren direkt anschließen und mit Strom versorgen kann. Die Anschaffung der Mikrocontroller in Arduino-Bauweise ist mit 10-30 €, je nach Variante, vergleichsweise günstig.

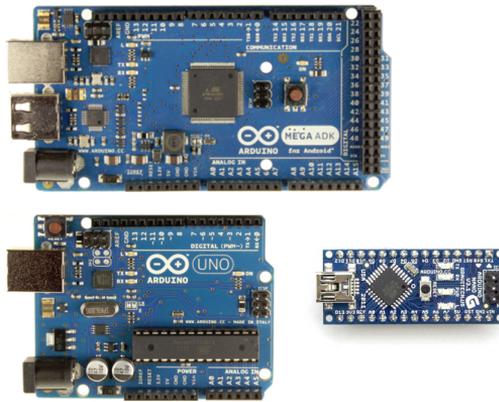


Abb. 2.21: Arduino Mikroprozessoren: Arduino Mega oben; Arduino Uno links; Arduino Nano rechts. (<http://arduino.cc/en/Main/Products>)

Die typischen Varianten des Arduino sind der Nano, Uno und Mega (Abb. 2.21). Sie unterscheiden sich hauptsächlich in der Anzahl der Anschlüsse, dem Speicher und in ihrer Rechenleistung. Für Arduino Mikroprozessoren existieren als Modifikationen für gezielte Anwendungen sogenannte Shields. Diese Shields liefern direkte Funktionen, beispielsweise GSM, LAN, WLAN oder GPS, und können zur Hardwareerweiterung an den Arduino angeschlossen werden. Mit Hilfe von Bibliotheken lassen sich diese Funktionen relativ einfach nutzen. Softwaregrundlage ist die Open-Source Arduino-IDE. Mit dieser Entwicklungsumgebung können Programme erstellt und auf den Arduino per USB-Schnittstelle hochgeladen werden. Bei jedem Start des Arduino wird das Programm in einer sich wiederholenden Schleife ausgeführt [Arduino-Guide, 2015].

Aufbau einer Sensorplattform auf Basis der Sensoren im Nexus 4

Für die Erzeugung der Sensordaten auf Rohdatenebene wird ein Arduino Nano Board mit den Sensortypen MPU-6050 und dem BMP-180 des Nexus 4 verbunden. Auf einer SD-Karte werden die gewonnenen Sensordaten gespeichert, um sie anschließend mit Matlab auszuwerten. Abb. 2.22 zeigt das auf einer Streifenrasterplatine gelötete Sensorboard. Von links nach rechts sind das SD-Karten Shield mit Mikro-SD-Karten-Steckplatz, der Arduino Nano, der MPU-6050 und der BMP-180 Luftdrucksensor zu sehen. Die Stromerzeugung wird mit einer 9V-Blockbatterie sichergestellt. Die Laufzeit beträgt mit einer 200 mAh Batterie etwa vier Stunden.

Das auf dem Arduino Nano entwickelte Programm liest die Sensorrohdaten für den MPU-6050 und BMP-180 aus und speichert sie auf der SD-Karte. Für weitere Informationen zum Aufbau der Hardware und für die Entwicklung des Programms sei an dieser Stelle auf die einschlägige Literatur [Schmidt, 2012], [Margolis, 2012]), [McComp, 2013]) und die zahllosen Beiträge im Internet verwiesen.

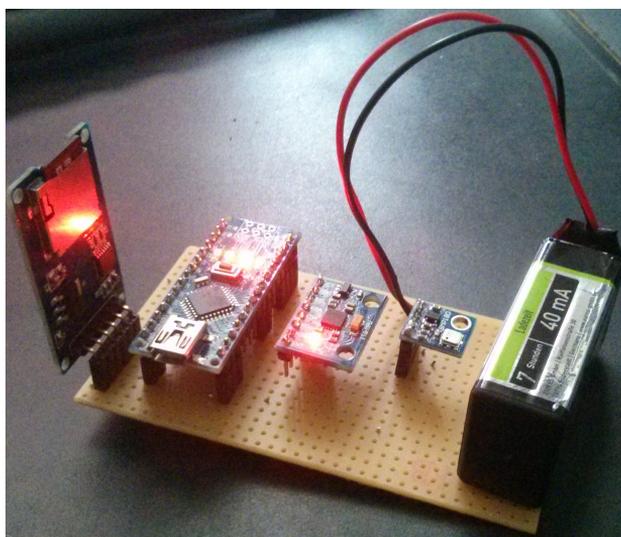


Abb. 2.22: Versuchsaufbau für die im Nexus 4 verbauten MEMS-Sensoren basierend auf einem Arduino Nano (von links: SD-Karten-Shield, Arduino Nano, MPU-6050, BMP180, 9V Blockbatterie).

Der MPU-6050 bietet vier Messbereiche für das Gyroskop wie für den Beschleunigungssensor an. Je nach ausgewähltem Messbereich sinkt bzw. steigt die Auflösung, da der Sensor eine Sensitivität von 16-bit bereitstellt. Das bedeutet, dass sich der Messbereich auf 65.536 Zustände verteilt [InvenSense, 2015]. In Tab. 2.4 werden die Messbereiche mit ihrer Auflösung zusammengefasst. Zu Beginn einer Messung muss der Sensor beim Setup auf den gewünschten Messbereich eingestellt werden. Für die Daten der folgenden Tabelle wurden jeweils die kleinsten Messbereiche ausgewählt, um die höchste Auflösung zu gewährleisten. Je nach Anwendung kann dabei eine hohe Auflösung oder ein hoher Messbereich von Interesse sein. Bei schnellen Drehbewegungen ist ein hoher Messbereich des Gyroskops notwendig, um die schnellen Drehungen zu detektieren – aufgrund der reduzierten Auflösung werden andererseits langsame Drehungen prozentual stärker verfälscht. Dies zeigt wie wichtig es ist, den Messbereich an den jeweiligen Anwendungsfall anzupassen. Die maximale Messrate des Sensors beträgt für beide Sensoren im MPU-6050 1.000 Hz.

Tab. 2.4: Auflösung und Arbeitsbereich von Gyroskop und Beschleunigungssensor im MPU-6050 ([InvenSense, 2015]).

Gyroskop		Beschleunigungssensor	
Messbereich [°/s]	Auflösung [°/s]	Messbereich [m/s ²]	Auflösung [m/s ²]
±250	0,0076	±2	0,00006
±500	0,0153	±4	0,00012
±1000	0,0305	±8	0,00024
±2000	0,0610	±16	0,00048

Die Analyse des Luftdrucksensors kommt im Test zu Ergebnissen, die mit denen des Nexus 4 vergleichbar sind. Daher wird an dieser Stelle auf eine detaillierte Darlegung weiterer Ergebnisse verzichtet. In Abb. 2.23 sind die Rohdaten der Messung des Gyroskops für das MPU-6050-Modul zu sehen (XYZ = RGB). Deutlich zu erkennen ist, dass die Daten keine sichtbare Drift aufweisen. Durch den integrierten Temperatursensor werden Drift-Abweichungen aufgrund von Temperatureinflüssen an die Rohdaten korrigiert. Der bei der Analyse der Testgeräte aufgedeckte ZUPT schlägt sich hier nicht nieder, obwohl eindeutig Nullpunkt-

fehler vorliegen, die über einen längeren Zeitraum stabil bleiben. Außer einigen Ausreißern zu Beginn der Messung weisen die Daten der jeweiligen Achsen ein Rauschen von $\pm 0,2 \text{ }^\circ/\text{s}$ auf. Die erreichte Messrate beim Arduino Nano beträgt 150 Hz.

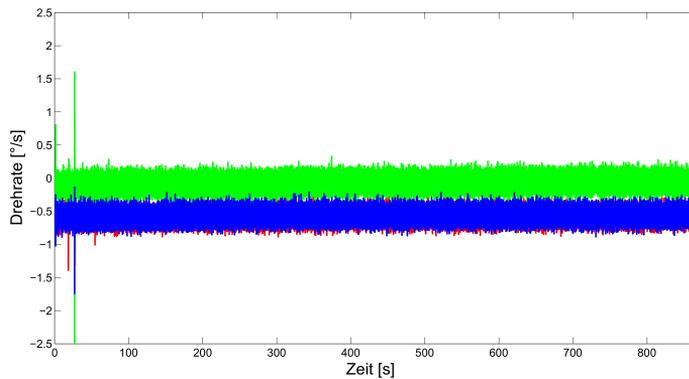


Abb. 2.23: Rohdaten des Gyroskops aus dem MPU-6050 im Arduino gesteuerten Versuchsgerät (RGB = XYZ).

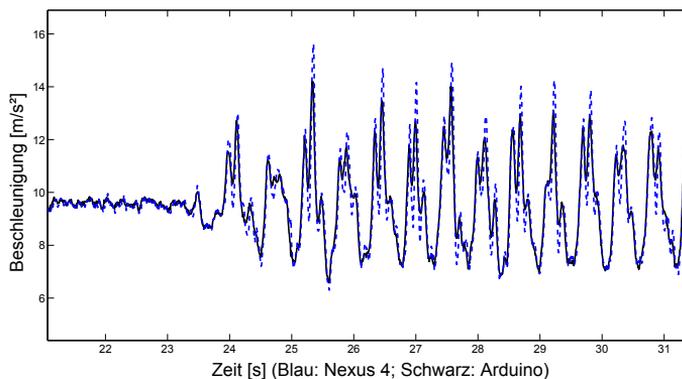


Abb. 2.24: Vergleich der z-Achse des Arduino Moduls mit der z-Achse des Beschleunigungsmessers aus dem Testgerät Nexus 4. Die Daten entstammen einem Testlauf im HCU-Gebäude bei dem beide Sensoren zusammen in einer Hand getragen wurden (schwarz: Arduino; blau: Nexus 4).

Für einen direkten Vergleich der Beschleunigungsmesser im Nexus 4 und der Eigenentwicklung wurde eine Strecke im HCU-Gebäude zeitgleich mit physisch verbundenen Sensoren abgelaufen. Zum Vergleich werden die z-Achsen beider Sen-

soren in Abb. 2.24 gegenübergestellt. Dabei wurde die Zeitachse manuell um ca. eine Sekunde verschoben, da es nicht möglich war, beide Applikationen synchron zu starten. Die Messfrequenz des Arduino-Moduls beträgt in diesem Versuch ca. 150 Hz und die Messfrequenz bei dem Smartphone Nexus 4 beträgt ca. 200 Hz . Beide Frequenzen rauschen aufgrund weiterer paralleler Prozesse wie beispielsweise der Speicherung der registrierten Daten. Abb. 2.24 zeigt deutlich die Ähnlichkeit beider Signale. Trotz unterschiedlicher Abtastrate sind die Schrittmuster relativ deckungsgleich, nur an den Spitzen wird die geringere Abtastrate beim Arduino-Modul erkennbar.

Es bleibt zu vermuten, dass der Beschleunigungsmesser im Smartphone keine Einwirkungen erfahren hat. Die Drehrate des Gyroskops unterliegt allerdings im Smartphone einer ZUPT-Korrektur. Weitere Glättungen der Daten aufgrund von internen Mittellungen waren nicht messbar.

3 Untersuchungen und Kalibrierung von MEMS im Smartphone

Dieser Abschnitt behandelt die Einschätzung zur Qualität und der Anwendbarkeit von MEMS-Sensoren zur Indoor-Navigation. Neben der Genauigkeit der Sensordaten für den Einsatz in den Fusionsfiltern soll ermittelt werden, ob sich ihre Leistung per Kalibrierung verbessern lässt. Zur Auswertung der dazu registrierten Sensordaten werden das Kalman Filter und die Methode der kleinsten Quadrate (MdkQ) eingesetzt.

3.1 Stockwerkserkennung mit Barometer BMP-180

In beiden Versuchsgeräten, dem Nexus 4 und dem Galaxy Nexus, ist der Luftdrucksensor BMP-180 von Bosch verbaut. Um äußere Einflüsse wie Wetterdrift und Temperaturänderungen zu korrigieren, wird üblicherweise ein zweiter ortsfester Luftdrucksensor für den ingenieurgeodätischen Einsatz genutzt. Die erfassten Luftdruckveränderungen, die nicht auf Höhenänderungen zurückzuführen sind, lassen sich so erfassen und bei den Daten des sich bewegenden Gerätes korrigieren. Luftdruckmessungen per Smartphone können nicht auf ein solches Referenzbarometer zurückgreifen. Daher unterliegen die mit den Luftdruckmessungen des MEMS-Barometers ermittelten Höhen sämtlichen äußeren Einflüssen.

Um die Eignung des BMP-180 zur Stockwerksunterscheidung in der Praxis zu erproben, wurde ein Treppenhaus im Gebäude D des ehemaligen HCU-Gebäudes, das vier Stockwerke miteinander verbindet, mit den Testgeräten abgelaufen. Aus Vermessungsübungen mit Studierenden sind die Stockwerkshöhen Zentimetergenau bekannt. Während der Versuche registrierten die Smartphones fortlaufend Messdaten. Die Geräte wurden in jedem Stockwerk vier Minuten lang positioniert, dabei sind die Stockwerke von oben nach unten und umgekehrt abgelaufen worden (in der Reihenfolge 4-3-2-1-1-2-3-4). Aus den ersten Luftdruckmessungen im vierten Stockwerk wird ein Mittelwert gebildet, der als Bezugswert dient, um relative Höhen mit der Formel (23) aus allen Daten zu berechnen.

Die Ergebnisse für das Galaxy Nexus sind in Abb. 3.1 und für das Nexus 4 in Abb.

3.2 dokumentiert. Die blauen Linien stehen für die Referenzhöhen. Die Luftdruckmessungen, in relative Höhen umgerechnet, sind schwarz abgebildet. Wegen der unterschiedlichen Messraten der Sensoren – beim Galaxy Nexus $12,5\text{ Hz}$ und beim Nexus 4: 25 Hz – wurde zusätzlich das gleitende Mittel für eine Sekunde ermittelt. Es ist grün dargestellt.

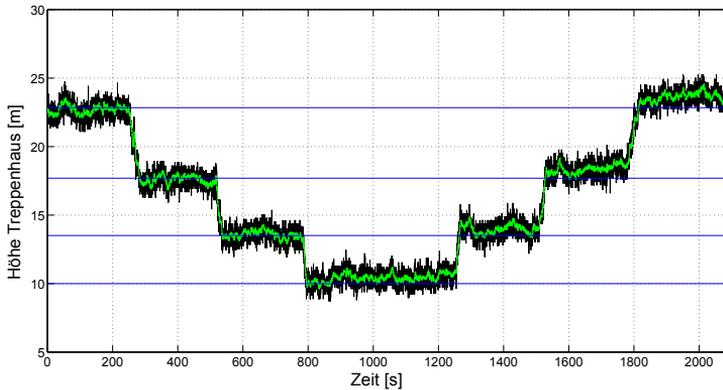


Abb. 3.1: Relative Höhen des Treppenhauses im ehemaligen HCU-Gebäude aus Luftdruckmessungen des Galaxy Nexus (schwarz: relative Höhen, grün: geglättet auf Basis der Messungen der vorherigen Sekunde, blau: Referenzstockwerkshöhen).

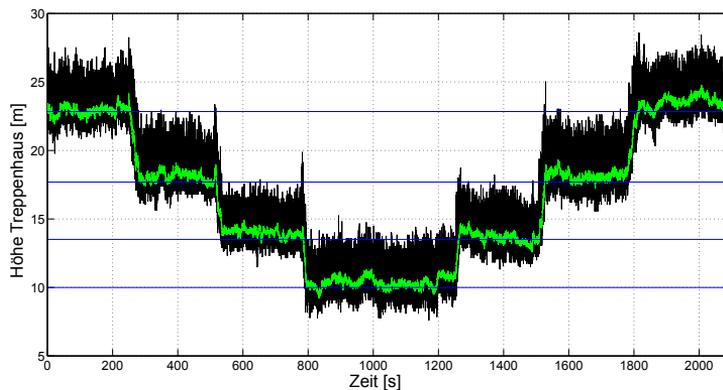


Abb. 3.2: Relative Höhen des Treppenhauses im HCU-Gebäude aus Luftdruckmessungen des Nexus 4 (schwarz: relative Höhen, grün: geglättet auf Basis der Messungen der vorherigen Sekunde, blau: Referenzstockwerkshöhen).

Die nicht geglätteten Höhen weisen bei beiden Geräten ein Rauschen im Bereich ± 1 m auf. Dabei weisen die Ergebnisse des Nexus 4 eine hohe Anzahl an Ausreißern auf, die allerdings nur in eine Richtung zeigen. Die geglätteten Datensätze zeigen, dass eine Stockwerkserkennung mit einer Auflösung von ± 1 m im Testgebäude möglich ist.

Tab. 3.1 fasst die Ergebnisse der Höhenbestimmung in Zahlen zusammen. Gegenübergestellt sind hier die Differenzen zu den Referenzhöhen sowie die Standardabweichungen der Ist-Höhen, berechnet aus den jeweiligen vierminütigen Messungen, um das Rauschverhalten der berechneten Höhen innerhalb dieser Testmessung in Zahlen zu verdeutlichen. Auch dieser Überblick belegt die prinzipielle Möglichkeit der Stockwerksunterscheidung – macht allerdings deutlich, dass die Abweichung der ermittelten Höhenunterschiede mit der Dauer der Beobachtung steigt. Diese Abweichungen können aus der Aufwärmung des Smartphones während der Messdatenregistrierung oder aus der Wetterdrift resultieren. Beim Galaxy Nexus ist diese Drift in der Höhendifferenz stärker ausgeprägt. Sie beträgt zwischen Anfang und Ende der Messung 0,93 m. Das Gerät Nexus 4 unterliegt unregelmäßigen Schwankungen in der Höhendifferenz, aber der maximale Fehler in der gesamten Messreihe beträgt nur 0,64 m.

Um den Einfluss durch die Geräteerwärmung genauer einschätzen zu können, werden Langzeitmessungen durchgeführt. Dabei werden ein hochgenaues Barometer, ein Setra B470T, und zusätzliche Temperatursensoren eingesetzt. Das Referenzbarometer erreicht im Arbeitsbereich von 800 bis 1100 hPa eine absolute Genauigkeit von $\pm 0,02$ % (0,06 hPa). Umgerechnet bedeutet dies eine metrische absolute Abweichung von ca. 0,5 m. Während der Langzeitmessung werden die Umgebungstemperatur sowie die Gehäusetemperatur des Smartphones erfasst. Zur Messung der Gehäusetemperatur dient ein mit Klebeband am Smartphone angebrachter Temperaturfühler, der, mit zusätzlichem Dämmmaterial versehen, den Einfluss der Außentemperatur weitgehend abschirmt. Die Abtastrate von Referenzluftdruck und beiden Temperaturmessungen liegt bei 1 Hz. Da die Versuchsreihen aus technischen Gründen nicht zeitgleich durchgeführt werden können, ist die Vergleichbarkeit der Luftdruckmessungen beider Testgeräte nur im relativen Vergleich zum Referenzwert gegeben.

Tab. 3.1: Vergleich der berechneten Stockwerkshöhen aus den Luftdruckmessungen beider Testgeräte zu Referenzhöhen.

Stockwerk	Galaxy Nexus		Nexus 4	
	Diff. (Ist - Soll) [m]	StdAbw. d. Ist-Messung pro Stockwerk [m]	Diff. (Ist - Soll) [m]	StdAbw. d. Ist-Messung pro Stockwerk [m]
4	-0,13	0,30	0,04	0,32
3	-0,10	0,29	0,39	0,33
2	0,08	0,26	0,49	0,27
1	0,42	0,28	0,34	0,36
2	0,46	0,32	0,14	0,35
3	0,66	0,30	0,45	0,31
4	0,80	0,29	0,68	0,42

Die Ergebnisse einer anderthalbstündigen Messung für das Galaxy Nexus geben die Abb. 3.3 und 3.4 wider. Die Abweichung zur Referenz beträgt -2,22 hPa im Mittel über den gesamten Messbereich – bei einer Standardabweichung von $\pm 0,036$ hPa. Ein Einfluss durch die Aufwärmung des Gehäuses war bei dieser Luftdruckmessung nicht nachweisbar. Das Gehäuse erwärmt sich von 21,8 °C bis auf 28.5 °C. Der Korrelationskoeffizient für die Referenzmessung und dem Galaxy Nexus Datensatz liegt bei 0.98. Das bestätigt die heuristische Annahme einer Unabhängigkeit der Messungen von der Gehäusetemperatur. Eine vorangegangene Untersuchung in [Sternberg u. a., 2012], die eine auffällige Temperaturdrift ausmachte, ließ sich nicht bestätigen. Allerdings gibt es einen Unterschied, der zwischen den damaligen und den aktuellen Untersuchungen auszumachen ist: das Update des Betriebssystems von Android 4.0 auf Android 4.2. Zu vermuten ist, dass damit Optimierungen auf Betriebssystemebene vorgenommen wurden.

Die Auswertung der Daten zum Nexus 4 (Abb. 3.5, Abb. 3.6) zeigt wie beim Galaxy Nexus keine Abhängigkeit der Luftdruckmessung von der Gehäusetemperatur. Zur Messdatenregistrierung wurde eine Beobachtungszeit von einer Stunde gewählt. Während dieser Zeit stieg die Temperatur des Nexus 4 von 22,8 °C bis auf 28,8 °C. Der Korrelationskoeffizient aus der Gegenüberstellung der Referenz

3.1 Stockwerkserkennung mit Barometer BMP-180

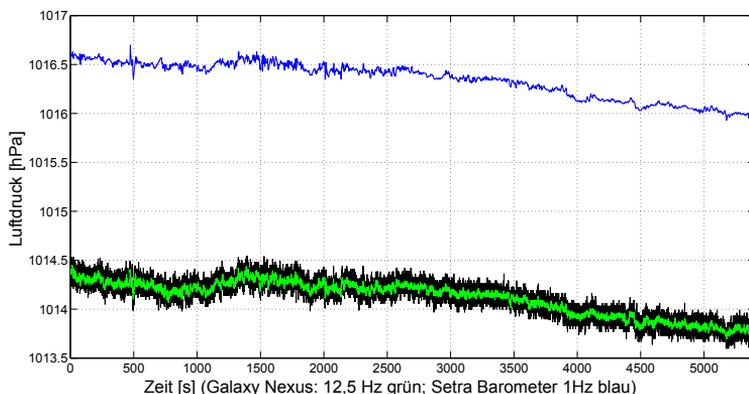


Abb. 3.3: Langzeitmessung des Luftdrucksensors im Galaxy Nexus im Vergleich zum Referenzgerät (Setra B470T).

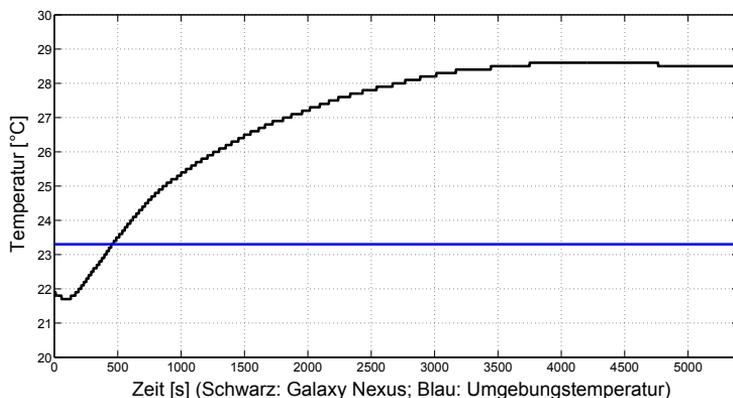


Abb. 3.4: Temperatur während der Langzeitbeobachtung des Luftdrucksensors im Galaxy Nexus aus Abb. 3.3 (schwarz: Gehäusetemperatur; blau: Umgebungstemperatur).

zu den Nexus 4-Daten liegt bei 0,96 und beweist, dass keine Drift aufgrund der Erwärmung des Smartphones messbar ist. Gegenüber dem Samsung Galaxy Nexus fällt der Abstand zum Referenzluftdruck bei diesem Gerät geringer aus. Er liegt bei nur 0,45 hPa mit einer Standardabweichung von $\pm 0,037$ hPa.

Die Untersuchungsergebnisse zeigen, dass Luftdrucksensoren in Smartphones prinzipiell zur Stockwerksunterscheidung geeignet sind. Auch die Erwärmung der Smartphones scheint keine Auswirkungen auf die Sensorstabilität zu haben. Die

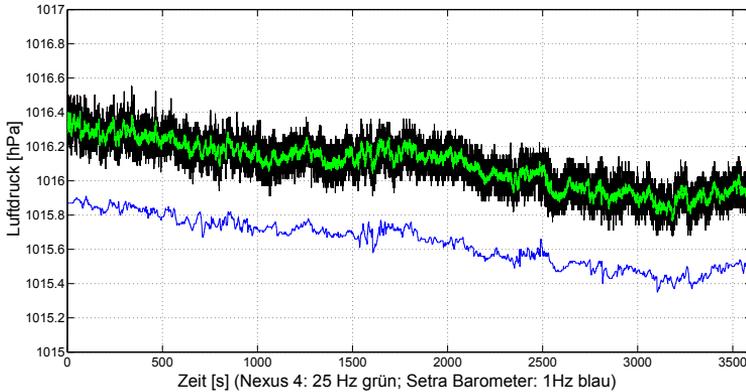


Abb. 3.5: Langzeitmessung des Luftdrucksensors im Nexus 4 im Vergleich zum Referenzgerät (Setra B470T).

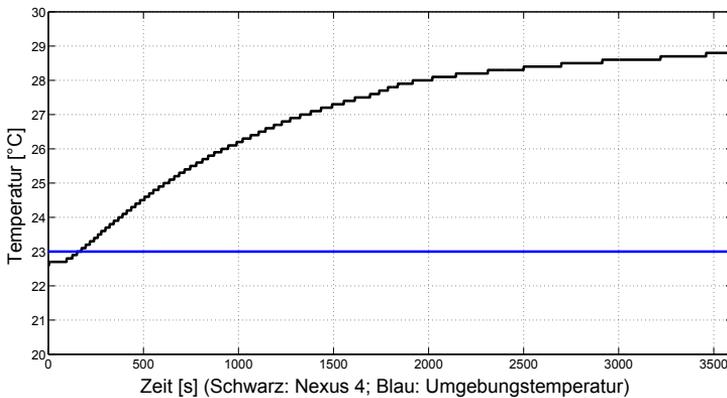


Abb. 3.6: Temperatur während der Langzeitbeobachtung des Luftdrucksensors im Nexus 4 aus Abb. 3.5 (schwarz: Gehäusetemperatur; blau: Umgebungstemperatur).

Luftdrucksensoren erlauben eine kurzfristige relative Höhenunterscheidung mit einer Genauigkeit von ± 1 m. Nach diesen Ergebnissen ist eine Kalibrierung des Barometers für die Stockwerkserkennung nicht erforderlich, da äußere Einflüsse wie die Wetterdrift oder Temperaturveränderung einen größeren Einfluss haben. Diese Temperaturveränderungen können im Gebäude durch unterschiedliche Raumklimata produziert werden. Die Auswirkungen beider Umgebungsgrößen – Wetterdrift und Temperaturveränderungen – lassen sich durch regelmäßige Reinitialisierungen des Bezugsluftdrucks während der Positionsschätzung minimieren.

3.2 Kalibrierung eines Gyroskops

Dazu muss allerdings die genaue Höhe bzw. das aktuelle Stockwerk bekannt sein, um weitere Unsicherheiten zu vermeiden.

3.2 Kalibrierung eines Gyroskops

Dreiachsige Gyroskope registrieren die Drehraten um die jeweiligen Achsen. Daher lassen sie sich zur Orientierungsbestimmung bei der Positionsschätzung mit Smartphones verwenden. Die zuvor diskutierten potentiellen Fehlerquellen wirken sich dabei allerdings auf die Rohdaten aus und verfälschen die aus Drehraten integrierten Drehwinkel.

Im Folgenden werden nun zwei Ansätze zur Kalibrierung eines Gyroskops im Smartphone vorgestellt, die mithilfe einer Totalstation erfolgen. Die erste Kalibrierung gilt der Dreh-rate auf Rohdatenebene, die zweite den Drehwinkeln. Die Untersuchung von Low-Cost-Inertialsensoren mittels Totalstation ist nicht neu. In Artese [2008] und Sternberg u. a. [2012] wird ihr Einsatz zur Kalibrierung eines MEMS Gyroskops vorgestellt.



Abb. 3.7: Modifizierte Totalstation Leica TCRP 1105+ zur Kalibrierung von Gyroskopen im Smartphones [Sternberg u. a., 2012].

Der Versuchsaufbau besteht aus einem Leica TCRP 1105+, das so modifiziert wurde, dass an dem drehbaren Gehäuse ein Smartphone befestigt werden kann (Abb. 3.7). Durch die vorhandene GeoCom-Schnittstelle kann die Totalstation vergleichsweise einfach per Matlab oder per C++ angesteuert werden. In der vorliegenden Versuchsanordnung wird die Totalstation mit einer Matlab-Routine über einen angeschlossenen PC gesteuert. Die Maximalgeschwindigkeit der verwendeten Totalstation liegt bei $0,78 \text{ rad/s}$ (50 gon/s).

Wegen der Veränderung am Aufbau der Totalstation können die in der Prüfroutine eingestellten Drehgeschwindigkeiten verfälscht sein, daher müssen die Drehraten der Totalstation vor der Kalibrierung überprüft werden. Zur Kalibrierung bieten sich verschiedene Ansätze an [Sternberg u. a., 2012]. Eine Möglichkeit besteht in der Anbringung einer lichtempfindlichen Diode gegenüber der Totalstation mit einer exakten Stoppuhr. Bei jeder vollendeten 360° -Drehung wird die Diode durch den aktivierten Laser der Totalstation beleuchtet und gibt einen Zeitstempel aus. Eine weitere Möglichkeit ist die Verfolgung der Drehung durch ein Koordinatenmessgerät, das zu jeder Koordinate einen Zeitstempel speichert. Für diese Untersuchung wurde dazu der Lasertracker Leica AT-901MR verwendet (Abb. 3.8).

Zum Einsatz kommt der AT-901MR in Verbindung mit einem $0.5''$ Corner Cube Reflektor (CCR). Die 3D-Punktgenauigkeit des Lasertrackers beträgt $\pm 15 \mu\text{m} + 6 \text{ ppm}$ (MPE). Die interne Uhr weist eine Drift von $20 \mu\text{s/s}$ auf (laut Leica TrackerTriggerInterface-Datenblatt). Bei Annahme dieser Genauigkeiten, einer Messentfernung von ca. 2 m und der Minimierung systematischer Fehler durch einen geringen Messbereich ist ein maximaler Fehler bei maximaler möglicher Geschwindigkeit von 0.05 mrad/s zu erwarten [Sternberg u. a., 2012]. Die Messungen wurden für Drehraten zwischen 0,1 und 0,6 rad in 0,1 rad Schritten durchgeführt, um eine Maßstabskorrektur für die an der Totalstation eingestellten Drehgeschwindigkeiten zu berechnen (Tab. 3.2).

Die Differenzen der Drehraten aus Lasertrackerdaten zu den eingestellten Drehraten an der Totalstation in Tab. 3.2 belegen eine Verlangsamung der Drehgeschwindigkeit, die vermutlich durch das erhöhte Gewicht des Systems zu erklären



Abb. 3.8: Versuchsaufbau für die Kalibrierung der Drehgeschwindigkeiten des Leica TCRP 1105+ mit einem Lasertracker Leica AT901-MR [Sternberg u. a., 2012].

ist. Der Maßstab konnte mit einer Standardabweichung von $\pm 0,0008$ bestimmt werden.

In den folgenden Untersuchungen werden die eingestellten Drehraten gemäß dem berechneten Maßstab korrigiert, aber im Folgenden weiterhin mit den eingestellten Drehraten bezeichnet. Der Einsatz der Totalstation als Referenz erlaubt zwei Möglichkeiten zur Kalibrierung. Im ersten Ansatz werden die gemessenen Drehraten direkt verglichen, der zweite basiert auf einem Winkelvergleich. Dazu werden Winkelschritte mit der Totalstation nacheinander abgefahren und mit den integrierten Drehraten verglichen.

Kalibrierung mit Totalstation auf Rohdatenebene

Zur Kalibrierung der Totalstation mit Referenzdrehraten wird das Smartphone in der aufgesetzten Halterung an der Totalstation positioniert und arretiert. Ist die

Tab. 3.2: Vergleich der Drehraten aus Lasertrackerdaten mit den eingestellten Drehraten an der Totalstation [Sternberg u. a., 2012].

Soll (rad/s)	Ist (rad/s)	Differenz (rad/s)
0.1	0.0962	-0.0038
0.2	0.1925	-0.0075
0.3	0.2884	-0.0116
0.4	0.3843	-0.0157
0.5	0.4810	-0.0190
0.6	0.5775	-0.0225
Maßstab	0.9629 (± 0.0008)	

Totalstation horizontal ausgerichtet, kann die Prüfroutine zur Steuerung der Totalstation beginnen. Dabei zeichnet die eigens erstellte Smartphone-Applikation die Gyroskop-Daten auf. Die Drehgeschwindigkeiten wurden zwischen -0,7 rad bis 0,7 rad in 0,1 rad Schritten abwechselnd eingestellt. Zwischen den Geschwindigkeitsübergängen wurden Stillstandszeiten eingefügt.

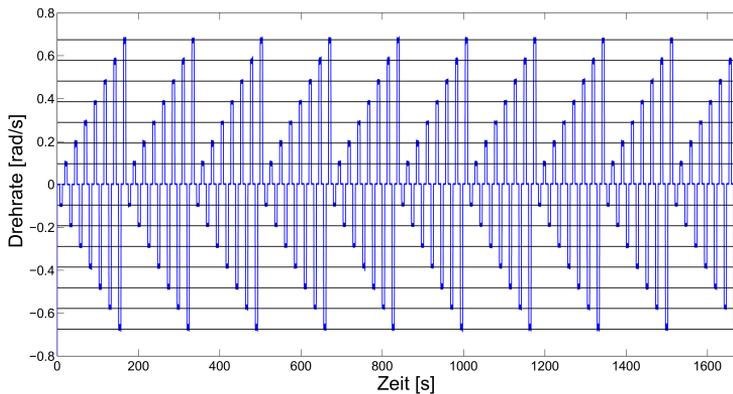


Abb. 3.9: Drehraten der z-Achse des Gyroskops im Galaxy Nexus, erzeugt durch Drehung der Totalstation (blau: Drehraten Gyroskop; schwarze horizontale Linien: Referenzgeschwindigkeiten).

Die Dauer einer Epoche wurde auf 6 Sekunden begrenzt, da der in Kapitelabschnitt 2.5 *Versuchshardware* beschriebene ZUPT-Befehl bei längeren gleichbleibenden Drehraten am Smartphone eine Korrektur der Drehrate vornimmt. Der

3.2 Kalibrierung eines Gyroskops

beschriebene Ablauf wurde im Versuch zehnmal unmittelbar hintereinander wiederholt. Abb. 3.9 zeigt einen Datensatz für eine Prüfung der z-Achse des Galaxy Nexus. Eine Detailansicht (Abb. 3.10) verdeutlicht das Rauschverhalten des Gyroskops.

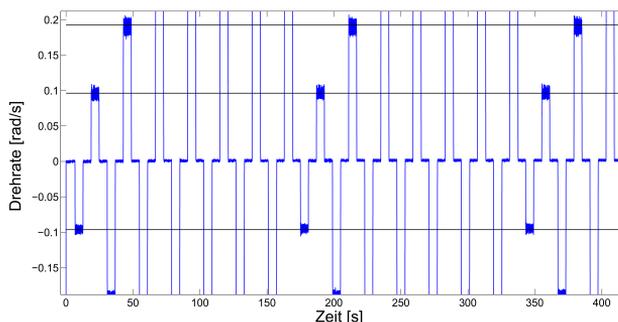


Abb. 3.10: Detailansicht aus Abb. 3.9 für die Drehratenmessung der z-Achse des Gyroskops im Galaxy Nexus (blau: Drehraten Gyroskop; schwarze horizontale Linien: Referenzgeschwindigkeiten).

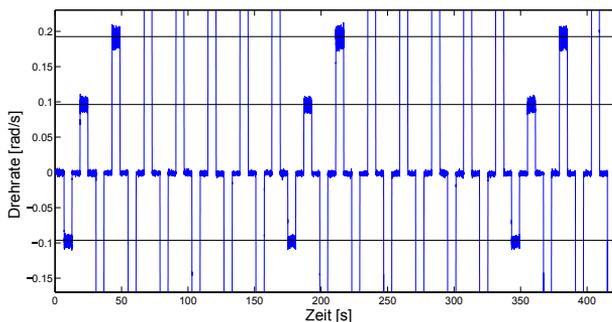


Abb. 3.11: Detailansicht der Drehratenmessung auf der Totalstation der z-Achse des Gyroskops im Nexus 4 (blau: Drehraten Gyroskop; schwarze horizontale Linien: Referenzgeschwindigkeiten).

Beim Nexus 4 wurden gleiche Testbedingungen wie beim Galaxy Nexus geschaffen. Aus Gründen der Vergleichbarkeit ist der Teilausschnitt der Prüfung hier vergrößert dargestellt (Abb. 3.11). Gegenüber dem Galaxy Nexus lässt sich ein erhöhtes Messrauschen der Drehraten verzeichnen. Allerdings arbeitet der Sensor auch mit einer zweifach höheren Messrate (200 Hz). Bei beiden Smartphones ist

das Rauschen in Ruhelage deutlich geringer als während der Rotationsbewegungen. Das deutet auf eine ungleichmäßige Rotation der Totalstation bei eingestellter Drehgeschwindigkeit hin.

Einen 6-Sekunden Block bei der Gyroskop-Kalibrierung zeigt Abb. 3.12. Das vermeintlich hohe Rauschen ist deutlich sichtbar ein periodischer Verlauf, der bei allen eingestellten Drehraten zu verzeichnen ist. Mit der Fourieranalyse konnte bei steigender Drehgeschwindigkeit der Totalstation ein Anstieg der Frequenz des Signals festgestellt werden. Für die Geschwindigkeiten 0,1 rad/s, 0,2 rad/s und 0,3 rad/s liegen die Frequenzen bei 2,8 Hz, 5,6 Hz und 8,5 Hz. Das bedeutet, dass die Schwingungen in Korrelation mit der Drehgeschwindigkeit stehen. Aus den Daten lässt sich ableiten, dass sich die Schwingung in 2° -Abständen wiederholen. Da die Motorregelung versucht, die über die GeoCom-Schnittstelle eingestellte Drehrate zu halten, muss sie ständig die Geschwindigkeit anpassen. Dieser Effekt macht deutlich, in welcher Qualität Drehraten im Gyroskop detektiert werden können.

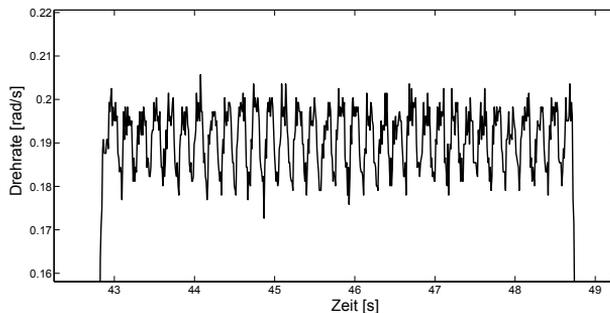


Abb. 3.12: Detailansicht der Drehraten während einer fest eingestellten Drehgeschwindigkeit an der Totalstation für eine Kalibriermessung. Periodische Schwingung, hier bei 0,2 rad/s, aufgrund von Motorregelung.

Die Messung für die Testgeräte dauerte jeweils 28 Minuten. Für jeden 6-Sekunden-Block wurde aus den Daten der mittleren vier Sekunden ein Mittelwert generiert und der Einfluss durch die Motorregelung vernachlässigt, da die Referenzdrehgeschwindigkeiten aus der Lasertrackermessung auch der Motorregelung unterliegen. So enthält jeder Datensatz 140 Drehgeschwindigkeiten. Aus allen Messungen \tilde{r}_z wurde ein Maßstab m_z und ein Offset b_z mittels Geradenausgleichung (MdkQ) auf Basis des funktionalen Zusammenhangs nach Formel (24) berechnet.

$$\tilde{r}_z = \dot{r}_z * m_z + b_z \tag{24}$$

Die Ergebnisse der Berechnung von Maßstabskorrektur und Offset für die z-Achsen beider Testgeräte sind in Tab. 3.3 aufgeführt. Die Maßstabskorrektur des Galaxy Nexus weist gegenüber der Maßstabskorrektur des Nexus 4 einen fünffach höheren Wert auf. Vermutlich ist die neue Sensorgeneration im Nexus 4 weniger fehlerbehaftet oder das Betriebssystem stellt eine System-Kalibrierung zur Verfügung.

Einen weiteren Einfluss auf die ermittelten Größen hat die nicht parallele Ausrichtung der z-Achse des MEMS-Sensors zur Drehachse der Totalstation, da aus untersuchungstechnischen Gründen die Hülle des Smartphones als parallel zum Sensorkoordinatensystem verlaufend angenommen werden muss. Das besondere Design des Samsung Galaxy Nexus erschwert daher die Ausrichtung des Gyroskops, denn anders als das Nexus 4 hat das Gehäuse an der Rückseite eine nicht planare Oberfläche.

Tab. 3.3: Ergebnisse der Ausgleichung aus den Kalibriermessungen für die z-Achsen der Gyroskope in beiden Testgeräten.

Versuchsgerät	Maßstab		Offset [rad/s]	
	Ist	StdAbw. (95%)	Ist	StdAbw. (95%)
Galaxy Nexus	1.0104	0.0002	-0.0018	0.0001
Nexus 4	0.9980	0.0002	0.0013	0.0001

Abb. 3.13 und Abb. 3.14 zeigen die Verbesserungen der jeweiligen Ausgleichungen von Maßstab und Offset. Erkennbar beim Galaxy Nexus ist ein Einlaufverhalten des Sensors, das aus der Erwärmung des Geräts resultieren kann. In Sternberg u. a. [2012] wird der Einfluss der Temperaturdrift für das Galaxy Nexus berechnet, es ergibt sich eine Drift von 0,02 °/s/h.

Insgesamt belegen die Ergebnisse, dass eine Kalibrierung des Gyroskops beider Smartphones per Rotation möglich ist. Die Durchführung einer solchen Kalibrierung ist allerdings aufwendig und eine Totalstation steht meist nicht immer zur Verfügung. Auch beeinflusst die Motorregelung die Kalibrierung. Eine Alternative zur Kalibrierung per Rotation könnte beispielsweise ein Schallplattenspieler

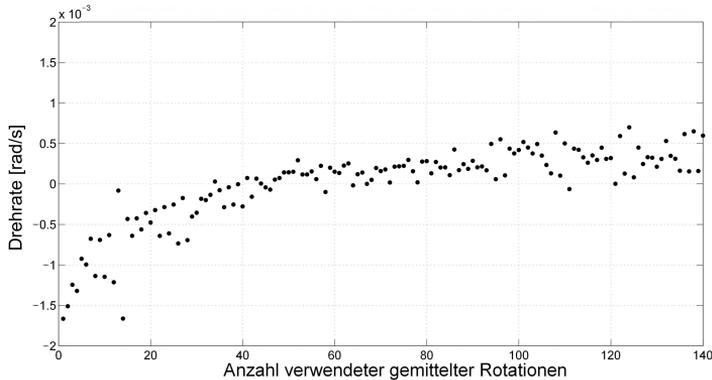


Abb. 3.13: Verbesserungen nach der Ausgleichung der Drehgeschwindigkeiten des Gyroskops im Galaxy Nexus.

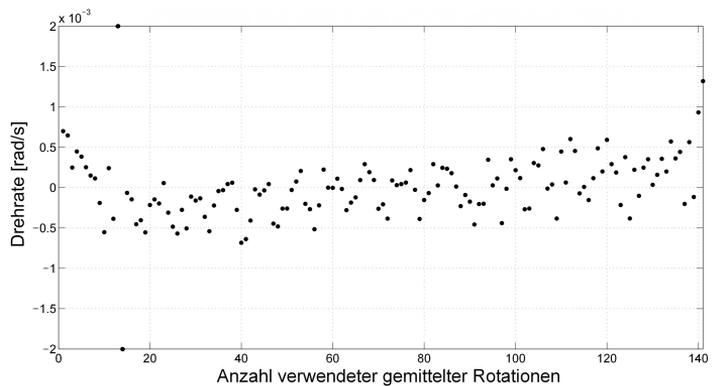


Abb. 3.14: Verbesserungen nach der Ausgleichung der Drehgeschwindigkeiten des Gyroskops im Nexus 4.

sein, denn die Drehgeschwindigkeiten dieser Geräte sind in aller Regel von hoher Präzision.

Der Nutzen einer Berücksichtigung der ermittelten Korrekturwerte lässt sich am Beispiel einer typischen Personenbewegung abschätzen. Bei dem angenommenen Richtungswechsel einer Person mit einer Geschwindigkeit von $180^\circ/\text{s}$ ergibt sich beim Galaxy Nexus ein Richtungsfehler von $+1,87^\circ$ und beim Nexus 4 von $-0,36^\circ$. Bedenkt man die zusätzlichen Einflüsse wie das sensorinhärente Rauschen, so kann auf die Korrektur des Maßstabs verzichtet werden.

Kalibrierung des Gyroskops mit definierten Drehwinkeln

Ein weitere Möglichkeit der Überprüfung des Gyroskops kann über einen Winkelvergleich erfolgen, wie es beispielsweise Sternberg u. a. [2012] zeigt. Danach erfolgt der Vergleich der Sensordaten auf Nutzungsebene, somit nach Integration der Drehraten. Der Versuchsaufbau entspricht dem Versuch mit festgelegten Drehgeschwindigkeiten.

Für den Versuch wird die Matlab-Routine editiert, um sukzessiv eine Folge von Positionen anzufahren und so definierte Referenzwinkel zu erhalten. Abb. 3.15 zeigt die Drehraten für einen Messablauf mit $\Pi/2$ (90°) Winkelschritten in verschiedene Drehrichtungen. Die Achsen des Gyroskops, hier beim Galaxy Nexus (RGB = XYZ), reagieren je nach Drehrichtung der Totalstation. Verwendet wird die maximale Drehgeschwindigkeit (50 gon/s oder 0.78 rad) der Totalstation, die bis zum Erreichen der neuen Position gehalten wird.

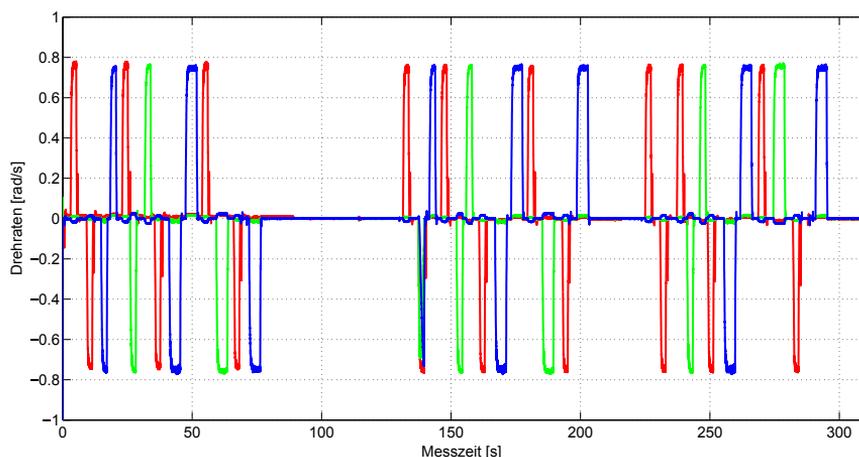


Abb. 3.15: Drehraten der Gyroskopkalibrierung im Galaxy Nexus für eine Kalibriermessung mit zuvor definierten Winkelschritten (RGB = XYZ).

Die in Abb. 3.15 aufgeführten Datensätze wurden anschließend integriert (Abb. 3.16). Die so ermittelten Drehwinkel sind immer relativ zur vorherigen Position zu betrachten, da die integrierten Drehraten aufsummiert werden. In der Abbildung fällt die x-Achse (rot) mit einer Drift auf, die nach 90 Sekunden den Drehwinkel um 1 rad beeinträchtigt. Zu Beginn haben die Achsen des Gyroskops im Galaxy

Nexus durch das sensorinhärente Rauschen erhöhte momentstabile Offsets, die das Betriebssystem im Smartphone mittels ZUPT korrigiert. Dies geschieht bei der x-Achse erst nach 90 Sekunden, da zuvor unterschiedliche Drehungen stattfinden und kein Stillstand über einen längeren Zeitraum möglich ist. Für die y- und die z-Achse wird gleich zu Beginn ein ZUPT ausgeführt, da hier die Drehungen erst nach ca. 20 Sekunden beginnen. Durch die Integration der Drehraten verhält sich der momentstabile Offset für den Winkel wie eine Drift. Den Einfluss eines manuell eingeführten ZUPT-Befehls zu Beginn der Winkelberechnung zeigt der rot-gestrichelte Datensatz für die x-Achse in Abb. 3.16. Für das Gyroskop im Nexus 4 wird vom Betriebssystem gleich zu Beginn ein ZUPT ausgeführt.

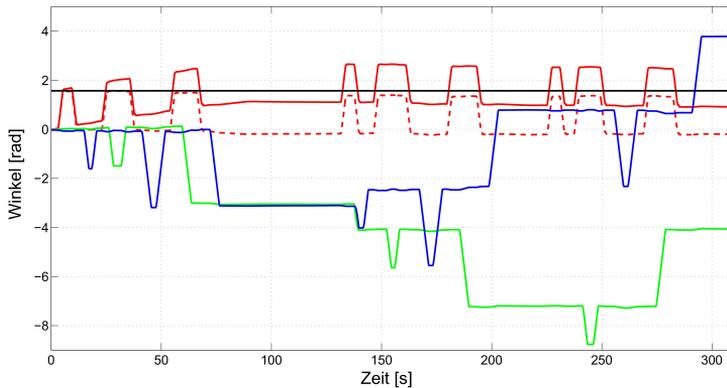


Abb. 3.16: Integrierte Drehraten aus der Kalibrierungsfahrt des Gyroskops in Abb. 3.15 mit Winkelschritten (RGB = XYZ; rot gestrichelt: um Drift bereinigte x-Achse).

Für einen Vergleich mit den Referenzwinkeln werden die Winkel an den einzelnen zeitlichen Positionen abgegriffen. Die Genauigkeit der Referenzwerte ergibt sich aus der Ansteuerungsgenauigkeit der Motorsteuerung. Da diese nicht bekannt ist, wird ein Wert anhand einer prismengestützten Distanzmessung abgeschätzt. Die Ansteuerungsgenauigkeit der Motorsteuerung wird mit 0.1 mrad angenommen, was einer Querabweichung von 10 mm in 100 m Entfernung entspricht. Die folgende Tab. 3.4 zeigt eine Auswahl aus dem Datensatz der abgegriffenen Winkel. Auffällig ist, dass fast alle berechneten Winkel kleiner als die Referenzwinkel sind. Die Nicht-Orthogonalität der Sensorachsen zueinander und eine fehlerhafte

3.3 Kalibrierung eines Beschleunigungsmessers

Ausrichtung der Drehachsen des Gyroskops um die Referenzdrehachsen der Totalstation erzeugen immer eine Registrierung der Drehgeschwindigkeiten in zwei oder allen drei Sensorachsen. In Abb. 3.15 ist dieser Effekt bei genauerer Betrachtung der Geschwindigkeiten nahe 0 rad/s erkennbar. Das bedeutet, dass stets zu geringe Drehraten auf der Bezugsachse registriert werden und dass die daraus abgeleiteten Drehwinkel kleiner sind, als die der Referenzmessung. Die ermittelten Differenzen betragen bis zu 0,03 rad ($1,7^\circ$) nach der Berücksichtigung der ZUPT-Korrekturen.

Tab. 3.4: Soll-Ist-Vergleich für berechnete Winkel aus der Kalibrierung des Gyroskops auf Basis von Winkelmessungen mithilfe der Totalstation; ausgewählte Winkel, ermittelt aus den Daten in Abb. 3.16.

Sensorachse	Istwinkel [rad]	Sollwinkel [rad]	Differenz [rad]
x	1,550	1,570	-0,020
x	1,542	1,570	-0,028
x	1,546	1,570	-0,024
x	1,552	1,570	-0,018
y	3,130	3,140	-0,010
y	1,570	1,570	0,000
z	1,555	1,570	-0,015
z	3,110	3,140	-0,030

Die Untersuchung der Gyroskope beider Smartphones belegt, dass eine Kalibrierung auf Rohdatenebene funktioniert. Die berechneten Korrekturen zeigen aber gleichzeitig, dass die Korrektur der Gyroskop-Daten zwar durchaus sinnvoll sein kann, allerdings aufgrund sich stärker ausprägender Einflüsse wie dem farbigen Rauschen, das sich als momentstabile Drift in den integrierten Drehraten niederschlägt, zu vernachlässigen sind. In dieser Arbeit wird der ZUPT-Befehl zur Minimierung des Drehraten-Offsets im Gyroskop favorisiert und kommt durchgängig vor jeder Messung zum Einsatz.

3.3 Kalibrierung eines Beschleunigungsmessers

Die Magnitude eines dreiachsigen Beschleunigungsmessers zeigt in Ruhelage die Erdbeschleunigung an. Dies wird in Smartphones genutzt, um den Inhalt des Dis-

plays, je nach Position des Smartphones, im Quer- oder Hochformat darzustellen. Eine Horizontierung des Koordinatensystems ist ebenfalls möglich. Sie ist für eine Positionsschätzung von Interesse, weil sie den Übergang von einer räumlichen Orientierung zu einer Drehung um die horizontierte z-Achse erlaubt.

Zur Horizontierung der dreiachsigen Smartphonesensoren sind die Neigungen r_x und r_y um die x- und y-Achsen erforderlich. Mit der daraus aufgestellten Rotationsmatrix R_{xy} können der Beschleunigungssensor, der Magnetfeldsensor und das Gyroskop parallel zum Horizont rotiert werden.

Sind die Messungen auf den Sensorachsen fehlerhaft, so ist die Verebnung der Sensorkoordinatensysteme in einem nicht horizontalen System die Folge. Alle horizontalen Sensordaten generieren dann Daten in einem falschen Koordinatensystem und detektieren auf verschiedenen Achsen Drehungen um eine Achse. Eine Überprüfung des Beschleunigungsmessers und gegebenenfalls eine Kalibrierung der Sensorachsen sind daher erforderlich.

Zur Kalibrierung des Beschleunigungsmessers wird ein ganzheitlicher Ansatz angestrebt, mit dessen Hilfe man alle drei Achsen gleichzeitig auswerten kann. So sollen mögliche Auswirkungen von Fehlausrichtungen für die einzelnen Achsen verringert werden. Da die Magnitude der drei fehlerfreien Achsen \bar{a}_x , \bar{a}_y und \bar{a}_z in Ruhelage den örtlichen Schwerewert g beschreibt, kann ein funktionaler Zusammenhang verwendet werden, der g einbezieht. Als funktionaler Zusammenhang dient der räumliche Pythagoras nach Formel (25).

$$g^2 = \bar{a}_x^2 + \bar{a}_y^2 + \bar{a}_z^2. \quad (25)$$

Lokale Referenzwerte für die Bundesrepublik Deutschland liefert die Physikalisch-Technische Bundesanstalt (PTB). Danach beträgt der Referenzwert Hamburgs für g $9,8137 \text{ m/s}^2$ [PTB, 2007]. Allerdings ist die lokale Beschleunigung zusätzlich abhängig von der Höhe. Der Einfluss der Höhe kann nach Sternberg [2000] abgeschätzt werden. Die aus Sternberg [2000] angepasste Formel (26) liefert den Höheneinfluss in $[m/s^2]$ für Hamburg (ellipsoidische Höhe $h = 50 \text{ m}$ und geographische Breite $\varphi = 53,5404$). Dabei wird bezogen zur Erdschwere am Äquator die Abplattung des Erdkörpers in Abhängigkeit der geographischen Breite berücksich-

3.3 Kalibrierung eines Beschleunigungsmessers

tigt. Der Einfluss durch die Höhe wird hier mit $0,0049 \text{ m/s}^2$ berechnet und kann daher allein aufgrund des hohen Rauschens der Low-Cost Sensorik vernachlässigt werden.

$$g_h = (0.308769 - 0.000440 * \sin^2(\varphi)) * h + 0.0000721 * h^2 / 100 \quad (26)$$

Für die Schätzung der Korrekturgrößen müssen der funktionale Zusammenhang in Formel (27) zwischen Referenz \bar{a}_x und Messwert a_x sowie die Korrekturgrößen Offset b_x und Maßstab m_x in die Formel (25) eingebunden werden.

$$\begin{pmatrix} \bar{a}_x \\ \bar{a}_y \\ \bar{a}_z \end{pmatrix} = \begin{pmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & m_z \end{pmatrix} * \begin{pmatrix} a_x & b_x \\ a_y & -b_y \\ a_z & b_z \end{pmatrix} \quad (27)$$

Für die Berechnung der Korrekturgrößen von Maßstab und Offset auf Basis der Referenzerdbeschleunigung wird das Kalman Filter (KF) nach Wendel [2007] favorisiert. Zur Kontrolle dient ein Ansatz nach der Methode der kleinsten Quadrate (MdkQ).

Das Bewegungsmodell A des KF wird als Einheitsmatrix definiert. Der Zustandsvektor x beinhaltet die Korrekturparameter Maßstab und Offset für die jeweiligen Achsen, die allerdings durch die Auflösung des funktionalen Zusammenhangs und die Trennung von Beobachtungen und zu schätzenden Parametern nicht direkt geschätzt werden können. Als Beobachtung y wird der quadrierte Referenzschwerewert $y = g^2$ eingesetzt. Der Zusammenhang zwischen den gemessenen Beschleunigungen und dem Referenzwert wird durch die Messmatrix H beschrieben (28).

$$y = H * x \quad (28)$$

Nach Einsetzen des Referenzschwerewerts und der Messwerte a_x , a_y und a_z ergibt der Zusammenhang zwischen Messwertvektor und Zustandsvektor ($y = H * x$) Formel (29):

$$g^2 = (a_x^2, -2 * a_x, a_y^2, -2 * a_y, a_z^2, -2 * a_z, 1) \cdot \begin{pmatrix} m_x^2 \\ m_x^2 * b_x \\ m_y^2 \\ m_y^2 * b_y \\ m_z^2 \\ m_z^2 * b_z \\ m_x^2 * b_x^2 + m_y^2 * b_y^2 + m_z^2 * b_z^2 \end{pmatrix} \quad (29)$$

Die Messmatrix H beinhaltet als Zeilenvektor nur die erzeugten Beobachtungen. Die Berechnung von Maßstäben und Offsets nach der Filterung kann mit den Formeln in (30) vorgenommen werden, dabei repräsentieren die Variablen x_1 bis x_6 die Parameter aus dem Zustandsvektor.

$$m_x = \sqrt{x_1} \ ; \ b_x = \frac{x_2}{x_1} \ ; \ m_y = \sqrt{y_1} \ ; \ b_y = \frac{x_4}{x_3} \ ; \ m_z = \sqrt{z_1} \ ; \ b_z = \frac{x_6}{x_5} \quad (30)$$

Die Qualität der geschätzten Korrekturparameter hängt maßgeblich von den verfügbaren Beobachtungsdaten ab. Für eine zuverlässige Schätzung der Korrekturparameter müssen sie daher den gesamten Messbereich jeder Achse abdecken. Dazu wird das Smartphone möglichst gleichmäßig in alle Achsrichtungen mehrfach auf die Erdbeschleunigung eingedreht. Das Ziel ist es, die einzelnen Achsen des Beschleunigungsmessers positiv und negativ auf die Erdbeschleunigung einzuspielen, ohne zusätzliche Beschleunigungen aufgrund der Drehbewegungen zu erzeugen. In Anlehnung daran wird der Bewegungsablauf für die folgenden Untersuchungen festgelegt, um zudem gleichbleibende Voraussetzungen für einen Vergleich zu schaffen. Abb. 3.17 zeigt den verwendeten Bewegungsablauf für die Erzeugung der Kalibrierung zu Grunde gelegten Daten.

In der Ausgangsposition für die Kalibrierung des Beschleunigungssensors weist das Display des Smartphones (v) nach oben. Die Drehungen müssen behutsam durch-

3.3 Kalibrierung eines Beschleunigungsmessers

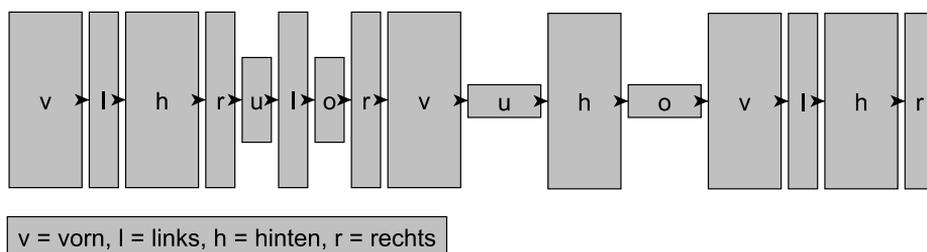


Abb. 3.17: Verwendeter Ablauf einer Kalibriermessung für den Magnetfeldsensor (räumlich) sowie für den Beschleunigungsmesser im Smartphone. Darstellung der Displayposition zum genäherten Horizont.

geführt werden, da der Sensor nicht im Drehzentrum des Smartphones verbaut und unvermittelte Beschleunigungen die Ergebnisse erheblich verfälschen. In Willemsen u. a. [2013] werden solche Einflüsse detailliert dargelegt und gezeigt, wie die Daten von Restbeschleunigungen bereinigt werden können. Allerdings räumen die Autoren ein, dass bei langsamer Drehung die zusätzlichen Beschleunigungen keine signifikanten Auswirkungen auf die Bestimmung der Korrekturparameter haben.

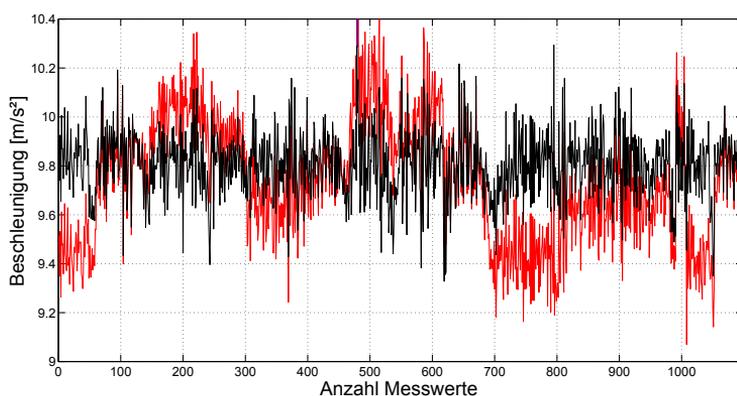


Abb. 3.18: Ergebnisse der Kalibrierung des Beschleunigungsmessers im Galaxy Nexus mit Kalman Filter. Vergleich der Magnituden (Zielgröße: $g = 9.81 \text{ m/s}^2$) vor und nach der Kalibrierung (rot: Magnitude fehlerbehaftet, schwarz: korrigierte Magnitude).

Für beide Versuchsgeräte werden Beschleunigungen nach dem beschriebenen Verfahren erzeugt. Die Ergebnisse des KF für beide Versuchsgeräte sind in Abb. 3.18

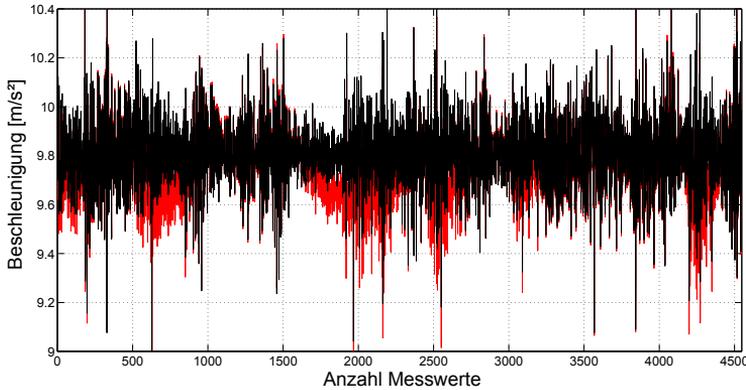


Abb. 3.19: Ergebnisse der Kalibrierung des Beschleunigungsmessers im Nexus 4 mit Kalman Filter. Vergleich der Magnituden (Zielgröße: $g = 9.81 \text{ m/s}^2$) vor und nach der Kalibrierung (rot: Magnitude fehlerbehaftet, schwarz: korrigierte Magnitude).

und Abb. 3.19 dargestellt. Dabei repräsentiert der rote Datensatz die Magnitude aller drei unkorrigierten Achsen der aufgezeichneten Daten. Nach Anbringung der ermittelten Korrekturparameter ergibt sich der hier schwarz wiedergegebene Datensatz.

Die Korrekturen beim Datensatz führen beim Galaxy Nexus zu deutlichen Verbesserungen. Das Nexus 4 scheint nur geringe Korrekturen zu benötigen. Die Daten zeigen hier gegenüber den Daten des Galaxy Nexus ein erhöhtes weißes Rauschen auf, was sich allerdings nicht in den Genauigkeiten der Korrekturparameter in Tab. 3.5 widerspiegelt und daher auf die erhöhte Aufzeichnungsrate zurückzuführen ist.

Tab. 3.5 stellt die Korrekturparameter für beide Testgeräte mit Genauigkeiten aus jeweils zwei Messungen dar. Die Werte für die Korrekturparameter sind mit einer Varianzfortpflanzung aus den funktionalen Zusammenhängen in Formel (30) und den auf Basis des KF berechneten Genauigkeiten bestimmt. Die geschätzten Maßstäbe unterscheiden sich zwischen beiden Geräten kaum in ihrer Größenordnung, allerdings weisen die Offsets deutliche Unterschiede auf. Die Systemkalibrierung für den MPU-6050 im Nexus 4 scheint besonders für die Offsets gute Korrekturen zu liefern.

3.3 Kalibrierung eines Beschleunigungsmessers

Tab. 3.5: Berechnete Korrekturparameter mit Standardabweichungen für die Kalibrierung des Beschleunigungsmessers beider Testgeräte.

Parameter	1. Messung		2. Messung	
	Maßstab / Offset [m/s ²]	StdAbw. (95%)	Maßstab / Offset [m/s ²]	StdAbw. (95%)
Galaxy Nexus				
m_x	0,9991	0,0027	0,9973	0,0032
b_x	-0,0226	0,0202	-0,0809	0,0262
m_y	0,9931	0,0031	0,9941	0,0039
b_y	-0,2442	0,0245	-0,2282	0,0333
m_z	1,0107	0,0032	1,0099	0,0035
b_z	-0,2988	0,0254	-0,2953	0,0283
Nexus 4				
m_x	0,9970	0,0022	0,9972	0,0021
b_x	0,0016	0,0134	0,0030	0,0121
m_y	1,0072	0,0027	1,0072	0,0022
b_y	0,0713	0,0199	0,0750	0,0138
m_z	1,0104	0,0024	1,0114	0,0024
b_z	0,0266	0,0162	0,0414	0,0154

Die Berechnung nach der MdkQ bestätigt im Rahmen der erreichten Genauigkeit die Ergebnisse der Auswertung auf Basis des KF. Bei beiden Testgeräten ist eine Bestimmung der Korrekturparameter einwandfrei möglich. Die Abweichungen zwischen Messung 1 und 2 demonstrieren allerdings auch eine minimale Instabilität der geschätzten Korrekturgrößen und könnte eine Sensorkalibrierung vor jedem Einsatz erforderlich machen. Beim Nexus 4 jedoch sind die Korrekturgrößen aufgrund der mutmaßlichen Systemkalibrierung gering und machen eine zusätzliche Kalibrierung überflüssig. In Willemsen u. a. [2013] wird am Beispiel des Galaxy Nexus aufgezeigt, dass eine fehlende Korrektur der Sensordaten bei der Berechnung der Verebnung einen Stehachsfehler von 1° erzeugen kann. Diese Fehlansrichtung führt bei einem Gyroskop beispielsweise dazu, dass sich die

Drehraten auf mehrere Achsen aufteilen und so die berechnete Orientierung in Abhängigkeit der Drehgeschwindigkeit abnimmt.

Hochwertige Inertialsysteme erlauben die Positionsberechnung aufgrund der zweifachen Integration des Beschleunigungssensors. Der Strapdown Algorithmus ist eine Möglichkeit der INS-basierten Positionsschätzung. Beim Strapdown Algorithmus dienen die Drehraten der Bestimmung der Lageänderung. Die Beschleunigungen werden zu Geschwindigkeiten integriert und anschließend zur Positionsberechnung erneut integriert. Die absolute Orientierung (Neigung, Nordsuche) kann aus einer Kombination von Beschleunigungsmesser und Magnetfeldsensor abgeleitet werden [Wendel, 2011]. Die Integration der Beschleunigungsdaten zu Geschwindigkeiten \vec{v}_{x_i} und Positionen \vec{x}_i erfolgt mit den Formeln (31, 32):

$$\vec{x}_i = \vec{x}_{i-1} + \vec{v}_{x_i} * dt + \vec{a}_i * dt * dt * 0.5 \quad (31)$$

$$\vec{v}_{x_i} = \vec{v}_{x_{i-1}} + \vec{a}_i * dt \quad (32)$$

Die Einsatzfähigkeit eines Strapdown Algorithmus zur MEMS-basierten Positionsschätzung, hängt maßgeblich von der Qualität der verfügbaren Sensordaten ab. Das Beispiel in Abb. 3.20 zeigt eine einfache und eine zweifache Integration der Beschleunigungen einer Messung mit dem Nexus 4 in Ruhelage. Die Messunsicherheiten, die sich über einen Zeitraum von 80 Sekunden ergeben, machen klar, dass der Einsatz eines Strapdown Algorithmus für MEMS-INS nicht sinnvoll ist. Das farbige Rauschen des MEMS-basierten Beschleunigungssensors erzeugt eine Drift bei den berechneten Geschwindigkeiten, die sich bei nochmaliger Integration verstärkt auf die Positionsberechnung auswirkt. Dieser gravierenden Abweichungen sprechen gegen die Nutzung des Strapdown Algorithmus beim Beschleunigungssensor.

Bei der Indoor-Navigation dienen MEMS-Beschleunigungssensoren daher als Schrittzähler. Bei dem Pedestrian Dead Reckoning (PDR) wird der Schrittzähler üblicherweise mit dem Gyroskop kombiniert und macht mit den integrierten Drehraten die Berechnung einer relativen Position möglich. Der für die vorliegenden Ausführ-

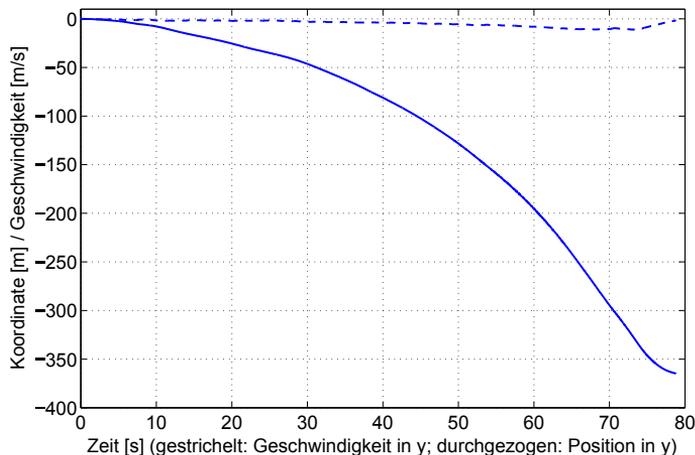


Abb. 3.20: Beschleunigungen einer Achse aus dem Testgerät Nexus 4 einer beliebigen Einzelmessung in Ruhelage zu Geschwindigkeiten (gestrichelte Linie) und zu Distanzen (blau) integriert.

rungen ausgewählte Schrittzähler wird im Kapitelabschnitt *4 Komponenten einer Smartphone gestützten hybriden Positionsschätzung* vorgestellt.

3.4 Kalibrierung eines Magnetfeldsensors

Ein Magnetfeldsensor wird in Smartphones häufig für Kompass-Applikationen eingesetzt. Die magnetisch bestimmte Nordrichtung ist auch als Bezugsrichtung für die Navigation von Interesse. Gegenüber dem Gyroskop unterliegt die Richtungsberechnung einer vergleichsweise geringen baubedingten Drift. Im Inneren von Gebäuden beeinträchtigen allerdings häufig elektromagnetische und ferromagnetische Einflüsse die Qualität der Sensordaten erheblich. Die Qualität der aus Magnetfeldsensordaten berechneten Größen ist daher immer abhängig von der jeweiligen Beschaffenheit der Gebäude.

Ebenso wie beim Beschleunigungsmesser basiert die Kalibrierung des Magnetfeldsensors auf einem erdbezogenen Referenzwert – in diesem Fall auf den Wert des lokalen Erdmagnetfelds. Der funktionale Zusammenhang für die gleichzeitige Kalibrierung der drei Achsen des Magnetfeldsensors kann daher direkt von der

Kalibrierung des Beschleunigungsmessers, wie in Abschnitt 3.3 *Kalibrierung eines Beschleunigungsmessers* dargelegt, abgeleitet werden. Der KF-Ansatz nach Wendel [2007] benötigt dafür lediglich eine Anpassung des Referenzwerts.

Die Referenzwerte der lokalen Magnetfelder stellt für Deutschland das Geoforschungszentrum (GFZ) in Potsdam zur Verfügung. Über dessen Webseite [GFZ, 2012] erhält man den Wert für das Erdmagnetfeld am angegebenen Ort sowie die Magnitude in der Horizontalen für magnetisch Nord.

Im Folgenden werden der dreidimensionale Ansatz nach Wendel [2007] und ein zweidimensionaler Ansatz mit dem funktionalen Zusammenhang auf Basis des Pythagoras vorgestellt. Für die Kalibrierung des Magnetfeldsensors muss sichergestellt sein, dass der Einfluss künstlicher Magnetfelder ausgeschlossen ist. Daher wurden die Messdaten zur Kalibrierung auf freiem Felde aufgezeichnet.

Kalibrierung des Magnetfeldsensors mittels dreidimensionalem Ansatz

Das Fehlermodell, und daher der funktionale Zusammenhang der Kalibrierung, entsprechen dem dreidimensionalen Ansatz der Beschleunigungsmesserkalibrierung. Die Kalibrierdaten lassen sich äquivalent zum Beschleunigungsmessersansatz (siehe Abb. 3.17) generieren, sodass eine Kalibrierung beider Sensoren auch synchron möglich ist.

Solche Kalibriermessungen wurden für beide Versuchsgeräte an gleicher Position durchgeführt und ausgewertet. In Abb. 3.21 und Abb. 3.22 sind die Ergebnisse für die Magnituden des jeweiligen dreiachsigen Sensors dargestellt. Der Referenzwert für den dreidimensionalen Ansatz beträgt $49,627 \mu T$ (Hamburg). Die blaue Linie in der Grafik repräsentiert die aus Rohdaten berechnete Magnitude und die schwarze Linie repräsentiert die mit Maßstäben und Offsets korrigierte Berechnung.

Bei beiden Geräten zeigen sich signifikante Verbesserungen der Datensätze, wobei die Ausgangswerte beim Nexus 4 stärker streuen. Die Ergebnisse sind zum Zweck einer besseren Darstellung geglättet. Wie zu sehen ist, liefern beide Geräte nach Anbringung der Korrekturparameter um den Referenzwert angenäherte Ergebnisse, allerdings verändert sich die Form der Abweichungen kaum. Das lässt vermuten, dass äußere Einflüsse auf die Sensoren wirken, die nicht durch den

3.4 Kalibrierung eines Magnetfeldsensors

funktionalen Zusammenhang abgebildet werden. Zudem können die zusätzlich im Smartphone verbauten Sensoren die Magnetfeldmessung beeinträchtigen.

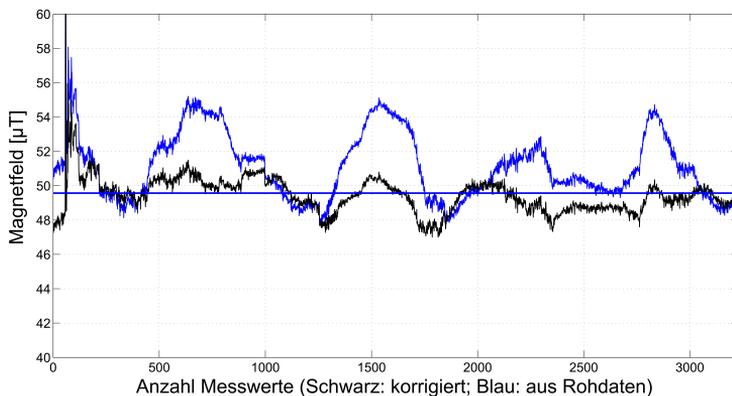


Abb. 3.21: Vergleich der Magnitude vor der Kalibrierung und nach der Korrektur aus den drei Achsen des Magnetfeldsensors im Galaxy Nexus (blau: aus Rohdaten; schwarz: korrigiert; Referenzwert für Hamburg: $49,627 \mu T$).

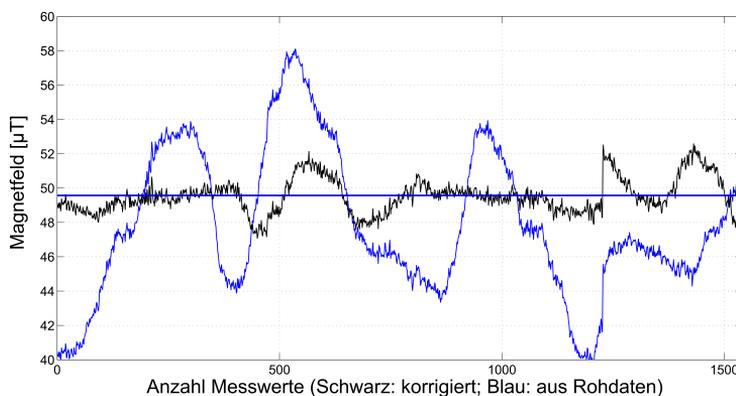


Abb. 3.22: Vergleich der Magnitude vor der Kalibrierung und nach der Korrektur aus den drei Achsen des Magnetfeldsensors im Nexus 4 (blau: aus Rohdaten; schwarz: korrigiert; Referenzwert für Hamburg: $49,627 \mu T$).

Die Korrekturparameter in Tab. 3.6 beider Geräte variieren relativ stark zwischen Messung 1 und 2 und lassen äußere Einflüsse vermuten. Die ermittelte hohe Genauigkeit der jeweiligen KF-Auswertung ist aufgrund der großen Anzahl an eingehenden Messwerten optimistisch geschätzt. Beide Messungen fanden am sel-

ben Ort zu nahezu gleicher Zeit statt. Die Auswertungen zeigen, wie instabil die Kalibrierung der Magnetfeldsensoren ist. Besonders die Offsets sind im Vergleich der jeweiligen berechneten Korrekturen nicht eindeutig bestimmbar.

Tab. 3.6: Berechnete Korrekturparameter aus zwei Kalibriermessungen jeweils für beide Magnetfeldsensoren in den Testgeräten auf Basis des räumlichen funktionalen Zusammenhangs.

Parameter	1. Messung		2. Messung	
	Maßstab / Offset [μT]	StdAbw. (95%)	Maßstab / Offset [μT]	StdAbw. (95%)
Galaxy Nexus				
s_x	0,9311	0,0001	0,9595	0,0001
b_x	-1,1254	0,0147	2,4030	0,0150
s_y	0,9275	0,0001	0,9638	0,0001
b_y	1,7280	0,0201	0,9616	0,0214
s_z	0,9083	0,0001	0,9443	0,0001
b_z	-1,0842	0,0171	-0,0313	0,0168
Nexus 4				
s_x	1,0914	0,0001	1,0741	0,0001
b_x	-3,4536	0,0229	-0,1300	0,0185
s_y	1,0044	0,0001	1,0094	0,0001
b_y	5,2751	0,0278	1,0567	0,0236
s_z	1,0374	0,0001	1,0310	0,0001
b_z	5,2912	0,0247	0,9215	0,0223

Kalibrierung des Magnetfeldsensors mittels zweidimensionalem Ansatz

Zur Bestimmung von Magnetisch Nord mit Magnetfeldsensordaten muss das Sensorkoordinatensystem in eine horizontale Ebene überführt werden. Anschließend lässt sich aus Messungen der x- und y-Achsen der Winkel zu Magnetisch Nord berechnen. Wenn der Magnetfeldsensor nur für diese Zwecke verwendet wird, ist die Kalibrierung des Sensors in der x-y-Ebene ausreichend.

Zur Horizontierung des Magnetfeldsensor-Koordinatensystems wird der kalibrierte Beschleunigungsmesser verwendet. Für diese Untersuchung wird vorausgesetzt,

3.4 Kalibrierung eines Magnetfeldsensors

dass die jeweiligen Sensorachsen zwischen den Sensoren parallel zueinander verbaut sind.

Auf Basis des Pythagoras und des Referenzwertes für Magnetisch Nord (Hamburg: $18,205 \mu T$ [GFZ, 2012]) ist der endgültige funktionale Zusammenhang unter Berücksichtigung der Korrekturparameter für die x- und y-Achse definiert durch (33):

$$M_{2D}^2 = s_x^2 * (m_x - b_x)^2 + s_y^2 * (m_y - b_y)^2 \quad (33)$$

M_{2D} bezeichnet in der Formel den Referenzwert für Magnetisch Nord. m_x und m_y stehen für die Sensorrohdaten. Die Maßstäbe werden mit s_x und s_y bezeichnet und b_x und b_y repräsentieren die Offsets. Zur Erzeugung der Messdaten zur Kalibrierung wird das Smartphone manuell möglichst auf der Horizontalebene mehrfach gedreht, um die Auswirkungen der Horizontierung durch die Beschleunigungen minimal zu halten. Die Messdaten des Magnetfeldsensors werden anschließend durch die Neigungen aus Beschleunigungen in den Horizont transformiert und dann der Kalibrierung zugeführt.

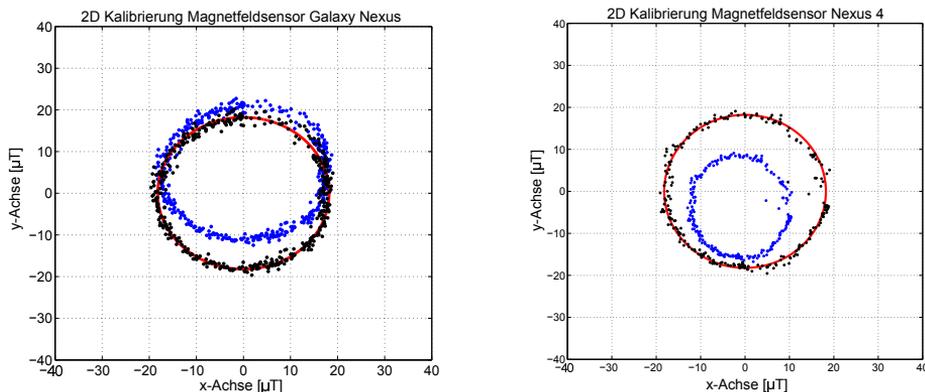


Abb. 3.23: Horizontierte Magnetfelddaten der x- und y-Achsen für das Galaxy Nexus (links) und das Nexus 4 (rechts) – vor und nach Berücksichtigung von Korrekturen einer Kalibrierung auf Basis des 2D-Pythagoras (blau: aus Rohdaten; schwarz: korrigiert; rot: Sollkreis).

Die Ergebnisse für beide Testgeräte sind in Abb. 3.23 zusammengefasst. Darge-

stellt sind die detektierten Magnetfelddaten entlang der x- und y-Achse. Durch Drehung des Smartphones und durch den Richtungsbezug nach Magnetisch Nord entsteht ein Kreis. Die blauen Punkte repräsentieren die Rohdaten. Durch Anbringung der Korrekturen aus der Ausgleichung erhält man die schwarzen Koordinaten. Anhand des roten Kreises, der mit einem Radius von $18.205 \mu T$ die Referenz anzeigt, ist die durch die Kalibrierung erfolgte Verbesserung deutlich sichtbar.

Tab. 3.7: Berechnete Korrekturparameter nach dem 2D-Ansatz zweier Kalibriermessungen für die Magnetfeldsensoren beider Testgeräte.

Parameter	1. Messung		2. Messung	
	Maßstab / Offset [μT]	StdAbw. (95%)	Maßstab / Offset [μT]	StdAbw. (95%)
Galaxy Nexus				
s_x	1,0324	0,0004	1,0244	0,0003
b_x	0,3603	0,0051	-0,1254	0,0039
s_y	1,1302	0,0004	1,1223	0,0003
b_y	4,9088	0,0048	4,8089	0,0036
Nexus 4				
s_x	1,6282	0,0010	1,1572	0,0006
b_x	-1,1962	0,0056	-1,7776	0,0065
s_y	1,4948	0,0009	1,1084	0,0005
b_y	-3,6648	0,0061	-3,2010	0,0066

In Tab. 3.7 sind die Korrekturen zweier aufeinander folgender Messungen aufgeführt. Wie bei der 3D-Auswertung ist auch hier eine Variabilität der Korrekturgrößen zwischen den unterschiedlichen Messungen zu erkennen. Zudem variieren die Ergebnisse im Vergleich zur 3D-Auswertung stark. Hier wäre zu erwarten gewesen, dass die geschätzten Parameter für die x- und y-Achse aus beiden Ansätzen, 2D und 3D, annähernd gleich sind. Diese Ergebnisse zeigen auf, dass aufgrund äußerer Einflüsse auf die Magnetfeldmessung eine Kalibrierung eines Magnetfeldsensors nur für den Moment stabil zu betrachten und daher nur bedingt nutzbar ist.

Prinzipiell ist ein Magnetfeldsensor für die Gebäude-Navigation aber hochinter-

3.4 Kalibrierung eines Magnetfeldsensors

essant, da seine mit der Nordsuche gegebene Absolut-Orientierung theoretisch die Drift der Orientierung des Gyroskops kompensieren kann. Zur Überprüfung der Praxistauglichkeit des Sensors wird nun exemplarisch eine Route im vierten Obergeschoss des HCU-Gebäudes hinsichtlich der Stabilität der berechneten Nordrichtung untersucht.

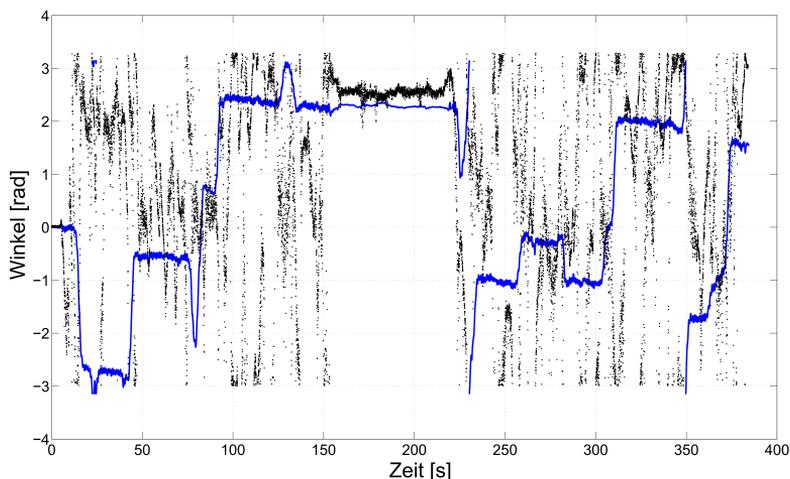


Abb. 3.24: Vergleich der Richtungen aus magnetisch Nord, berechnet durch die verebneten Magnetfeldsensordaten, und der Drehwinkel, berechnet aus verebneten Gyroskopdaten (blau: Gyroskop; schwarz: Magnetfeldsensor).

Abb. 3.24 zeigt die Nordrichtung, berechnet aus den horizontierten Daten des Magnetfeldsensors (schwarz), im Unterschied zu den aus den horizontierten Drehraten des Gyroskops berechneten Drehwinkeln der z-Achse (blau). Da die Drehwinkel aus Gyroskopdaten nur einen relativen Winkelbezug haben, wird zur Vergleichbarkeit der Nordrichtung des Magnetfeldsensors die erste Berechnung der Nordrichtung als Initialisierungsrichtung addiert.

Aus Gründen der Vergleichbarkeit werden die Drehwinkel um $2 * \pi$ bzw. $-2 * \pi$ korrigiert, sobald sie die Grenzen π und $-\pi$ übersteigen. Einzig der Bereich zwischen 150–220 Sekunden zeigt eine vergleichbar geringe Differenz bei vergleichbar geringen Datenrauschen zwischen beiden berechneten Winkeln. In diesem Zeitraum befand sich das Testgerät in Ruhelage, was die Berechnung von Magnetisch Nord begünstigen könnte, da keine Schrittmuster die Richtungsbestimmung wei-

ter beeinflussen. Insgesamt bedeutet das, dass sich der Magnetfeldsensor zur Richtungskorrektur im Gebäude nicht ohne ortsbezogene bekannte Informationen zum Einfluss des Gebäudes auf die Bestimmung der Nordrichtung eignet. Der Vorteil einer Kombination stochastisch unterschiedlicher Sensordaten zur Minimierung der Drift auf Kosten eines höheren Rauschens kann hier nicht gelten.

3.5 Einfluss der Nicht-Orthogonalität

Die Nicht-Orthogonalität zwischen den jeweiligen Achsen der dreiachsigen Sensoren – wie Beschleunigungsmesser, Gyroskop und Magnetfeldsensor – führt zu einer fehlerhaften Zuordnung der detektierten Messelemente. Die Abweichungen der Orthogonalität zwischen den Achsen sind fertigungsbedingt und können in einer Kalibrierung mit dem funktionalen Zusammenhang als Rotationsmatrix berechnet werden.

Weiterhin ist die Parallelität der Sensorachsen Voraussetzung für die Horizontierung von Gyroskop und Magnetfeldsensor durch den Beschleunigungssensor. Im Folgenden wird ein Ansatz zur Berechnung der Nicht-Orthogonalität der drei Achsen eines Beschleunigungsmessers beschrieben. Anschließend wird dieser Ansatz erweitert, um die Qualität der Parallelität zwischen den zwei Sensorkoordinatensystemen einschätzen zu können.

Orthogonalität der jeweiligen Sensorachsen

Zur Überprüfung der Orthogonalität der drei Achsen eines Sensors eignet sich die Formel (27), die schon zur Kalibrierung von Maßstab und Offset verwendet wurde. Um die Orthogonalitätsparameter erweitert ergibt sich die folgende Formel (34). Auf den Nebendiagonalen der 3x3 Matrix werden nun Drehparameter r_{ij} eingetragen, die die Nicht-Orthogonalität beschreiben:

$$\begin{pmatrix} \bar{a}_x \\ \bar{a}_y \\ \bar{a}_z \end{pmatrix} = \begin{pmatrix} m_x & r_{xy} & r_{xz} \\ r_{yx} & m_y & r_{yz} \\ r_{zx} & r_{zy} & m_z \end{pmatrix} * \begin{pmatrix} a_x & b_x \\ a_y & -b_y \\ a_z & b_z \end{pmatrix} \quad (34)$$

von Rosenberg [2006] verfolgt in seinem Ansatz die horizontale Ausrichtung des

MEMS-Sensors mithilfe eines Referenzwertes. Entsprechend soll nun der Sensor am Beispiel des Beschleunigungsmessers bestmöglich zur Erdbeschleunigung ausgerichtet werden. Für einen um Offset und Maßstab bereits korrigierten Datensatz zeigt die Formel (35) den funktionalen Zusammenhang für eine vertikale Sensorausrichtung der x-Achse des Beschleunigungsmessers auf.

$$\begin{pmatrix} 9,8137m/s^2 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & r_{xy} & r_{xz} \\ r_{yx} & 1 & r_{yz} \\ r_{zx} & r_{zy} & 1 \end{pmatrix} * \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \quad (35)$$

Der funktionale Zusammenhang in der Formel (35) enthält mindestens drei unbekannte Größen – unter der Annahme, dass die entgegengesetzten Drehungen wie r_{xy} und r_{yx} unter wechselndem Vorzeichen gleichgesetzt werden können. Zur Bestimmung der Korrekturgrößen muss jede Achse mindestens einmal zur Erdbeschleunigung ausgerichtet sein. Dazu wird jede Sensorachse vertikal zum Horizont ausgerichtet. Zusätzlich werden durch Drehung des Sensors negative Beschleunigungen aufgezeichnet für die der Referenzwert $-g$ eingeführt wird. So werden insgesamt sechs Beobachtungen erzeugt, die jeweils aus dem Koordinatentripel bestehen.

Für die Untersuchung am Gyroskop muss eine Referenzdrehung um die jeweilige Achse erzeugt werden. Die Ausrichtung ist bei einem Smartphone relativ schwierig, da das Gehäuse vieler Geräte, wie bereits erwähnt, uneben ist oder der Sensor nicht parallel zum Gehäuse verbaut sein kann. Daher empfiehlt sich dieser Ansatz nicht.

Um zusätzlichen Aufwand durch eine getrennte Kalibrierung von Maßstab, Offset und Orthogonalität zu vermeiden, wird in dieser Arbeit eine gemeinsame Ausgleichung der Korrekturparameter nach Formel (34) favorisiert. Von Vorteil dabei ist zudem die Minimierung fehlerhafter Zuordnungen durch die gemeinsame Auswertung getrennter Messepochen, da sich Unsicherheiten beispielsweise aufgrund von Orthogonalität in der Offset-Kalibrierung niederschlagen.

Für die gemeinsame Auswertung findet eine Ausgleichung nach der MdkQ mit dem leicht modifizierten Ansatz in Formel (34) statt. Die Kalibrierdaten werden nach

dem im Kapitelabschnitt 3.3 *Kalibrierung eines Beschleunigungsmessers* bereits beschriebenen Verfahren generiert.

Tab. 3.8 zeigt die Ergebnisse der um Orthogonalitätsparameter erweiterten Kalibrierung für die zweite Messung der Beschleunigungsmesser der Testgeräte. Gegenüber den Ergebnissen aus Tab. 3.5 ergeben sich leichte Veränderungen im Maßstab und im Offset. Dies zeigt die mögliche Überlagerung von Orthogonalitäten auf andere, zuvor geschätzte Korrekturen.

Tab. 3.8: Ergebnisse für Offset, Maßstab und Orthogonalität des zweiten Testdatensatzes zur Kalibrierung der Beschleunigungen unter Hinzunahme der Orthogonalitätsparameter für die Testgeräte Galaxy Nexus und Nexus 4.

Parameter	Galaxy Nexus		Nexus 4	
	Maßstab / Offset [m/s^2]	StdAbw. (95%)	Maßstab / Offset [m/s^2]	StdAbw. (95%)
a_x	-0,0428	0,0002	0,0011	0,0001
a_y	-0,2388	0,0002	0,0722	0,0001
a_z	-0,2945	0,0002	0,0421	0,0001
m_x	1,0041	0,000018	0,9986	0,000008
r_{xy}	-0,0018	0,0012	-0,0008	0,0005
r_{xz}	0,0071	0,0012	0,0025	0,0005
r_{yx}	-0,0018	0,0012	-0,0008	0,0005
m_y	0,9962	0,000024	1,0096	0,000010
r_{yz}	0,0019	0,0012	-0,0017	0,0005
r_{zx}	0,0071	0,0012	0,0025	0,0005
r_{zy}	0,0019	0,0012	-0,0017	0,0005
m_z	1,0120	0,000024	1,0134	0,000011

Der Einfluss der ermittelten Korrekturen aufgrund von Nicht-Orthogonalität auf reale Messdaten wird durch einen Vergleich einer typischen dreiachsigen Beschleunigung mit und ohne Berücksichtigung der Korrekturgrößen festgestellt. Tab. 3.9 listet beide Varianten und die Differenzen auf. Dabei zeigt sich, dass der Einfluss der Nicht-Orthogonalität vernachlässigt werden kann, da die errechneten Differenzen bei den Testgeräten in der Größenordnung des Sensorrauschens liegen.

3.5 Einfluss der Nicht-Orthogonalität

Tab. 3.9: Einflussgröße der Korrektur durch Orthogonalität am Beispiel des Beschleunigungsmessers der Testgeräte Galaxy Nexus und Nexus 4.

Vergleichsdatensatz	Mit Ortho.	Ohne Ortho.	Einfluss Ortho.
$[m/s^2]$	$[m/s^2]$	$[m/s^2]$	$[m/s^2]$
Galaxy Nexus			
x = 0,20	0,29	0,24	0,05
y = 5,20	5,43	5,42	0,01
z = 8,30	8,71	8,70	0,01
Nexus 4			
x = 0,20	0,22	0,20	0,02
y = 5,20	5,16	5,18	0,02
z = 8,30	8,36	8,37	-0,01

Parallelität der Sensorachsen zwischen Gyroskop und Beschleunigungssensor

Zur Überprüfung der Parallelität der Sensorachsen zwischen Gyroskop und Beschleunigungsmesser ist ein Versuchsaufbau nötig, der beide Sensoren in einem Referenzrahmen vereint. Für die Überprüfung des Gyroskops sind Referenzdrehtaten erforderlich. Daher muss hier eine Totalstation oder ein Drehtisch verwendet werden. Für diese Untersuchung wird dazu ein horizontal ausgerichteter Drehtisch der Firma *INA Drives & Mechatronics* mit der Bezeichnung *RDDS1-230x115-S-F-PD-WL-24000* verwendet. Auf ihm ist eine Halterungsvorrichtung für das Smartphone befestigt (Abb. 3.25).

In der Versuchsdurchführung wird jede Sensorachse des Smartphones auf dem Drehtisch einmal negativ und einmal positiv orthogonal zur Drehrichtung und damit parallel zur Erdbeschleunigung ausgerichtet. Der Drehtisch wird zuvor mit einer Röhrenlibelle horizontalisiert. Fehlerhafte Ausrichtungen aufgrund des Smartphonegehäuses wirken sich in dieser Versuchsanordnung weniger stark aus, da sie für beide Sensoren gleichermaßen gelten und nur die relativen Korrekturen das Maß für die Parallelität darstellen.

Wird nun der Drehtisch rotiert, registrieren die nicht ausgerichteten Achsen beider Sensoren nur noch Messdaten aufgrund von Abweichungen zur Orthogonalität

der eigenen Sensorachsen sowie aufgrund von Fehlausrichtung, sofern Offset und Maßstab zuvor korrigiert wurden. Für die Messung wird eine Drehrate von 30° und -30° gewählt, die ein Rauschen von $\pm 0,008^\circ$ aufweist. Für jede Ausrichtung einer Smartphoneachse werden zehn Sekunden lang Messdaten in Ruhelage und zehn Sekunden lang Daten während der Drehung aufgezeichnet. Aus jedem zehn Sekunden Datensatz wird für die weitere Auswertung der Mittelwert für jede Achse berechnet.

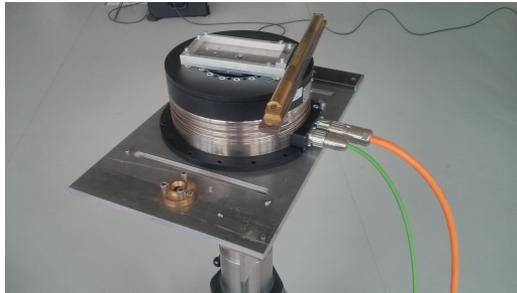


Abb. 3.25: Versuchsaufbau am Drehtisch zur Überprüfung der Parallelität der Sensorachsen von Beschleunigungsmesser und Gyroskop im Testgerät Nexus 4.

Pro Sensor werden 6 Sensor-Datensätze ausgewertet: positive und negative Ausrichtungen zum Referenzwert jeder Achse. Aus diesen Messungen wird mit Formel (35) für jeden Sensor aus dem funktionalen Zusammenhang eine Drehmatrix berechnet. Die Abweichungen in der Parallelität der Sensorachsen ergeben sich aus der Differenz beider Drehmatrizen.

In Tab. 3.10 sind die Testergebnisse für das Nexus 4 aufgelistet. Berechnet sind hier die Drehmatrizen nach der Formel (35), nachdem Offset und Maßstab gesondert korrigiert wurden. Dabei zeigt sich, dass sich die Parameter der Drehmatrix beider Sensoren ähneln und bis auf r_{xz} die gleichen Vorzeichen aufweisen. Außerdem fällt bei der Drehung r_{zy} die vergleichsweise große Drehkorrektur bei beiden Sensoren auf. Das kann ein Hinweis auf die vermutlich fehlerhafte Ausrichtung des Sensors zum Horizont sein. Die Differenzen der Orthogonalitätsparameter sind im Betrag kleiner 0,0032 und liegen im Vergleich zur gemeinsamen Auswertung von Orthogonalität, Offset und Maßstab (Tab. 3.8) in einer Größenordnung. Die Abweichungen gehen wie schon bei der Auswertung nach Tab. 3.9 im Rauschen unter.

3.6 Qualität des Zeitstempels im Nexus 4

Tab. 3.10: Orthogonalitätsparameter als Kriterium für die Parallelität der Sensorachsen des Gyroskops zum Beschleunigungsmesser des Nexus 4.

Drehung	Orthogonalität Gyroskop	Orthogonalität Beschleunigungsmesser	Differenz Orthogonalität
r_{xy}	-0,0055	-0,0030	0,0025
r_{xz}	0,0004	-0,0024	-0,0028
r_{yx}	0,0049	0,0037	-0,0012
r_{yz}	0,0046	0,0014	-0,0032
r_{zx}	0,0044	0,0032	-0,0012
r_{zy}	-0,0112	-0,0138	-0,0026

Insgesamt lässt sich festhalten, dass der Einfluss der Orthogonalitätsparameter auf die Messelemente vergleichsweise gering ist gegenüber den Korrekturen Maßstab und Offset. Die geschätzten Korrekturen der Testgeräte, besonders beim Nexus 4, sind derart gering, dass in den folgenden Versuchen eine Korrektur aufgrund von Nicht-Orthogonalität vernachlässigt wird und die jeweiligen Sensorachsen als parallel angenommen werden.

3.6 Qualität des Zeitstempels im Nexus 4

Während der Versuche werden die Rohmessdaten mit Hilfe einer eigenen App im Smartphone aufgezeichnet. Zum Einsatz kommt dabei ein Zeitstempel dessen Qualität die echtzeitfähige Entwicklung im Post-Processing beeinflusst. Damit die auf Basis der aufgezeichneten Sensordaten entwickelten Algorithmen auch unter den realen Bedingungen gelten, muss die Qualität des Zeitstempels überprüft werden. Eine Zeitdrift könnte beispielsweise unmittelbare die Berechnung der Orientierung aus den Gyroskopdaten verfälschen.

Zur Überprüfung des Zeitstempels der Smartphoneapplikation wird das Inertialmesssystem RQH1003 der Firma iMAR als Referenzzeitgeber verwendet. Das System kann mit einer GNSS-Antenne ausgestattet und mit der GPS-Zeit synchronisiert werden. Die Genauigkeit des Zeitstempels beträgt $10 \mu s$, sofern kein GNSS

empfangen wird [iMAR, 2010]. Für die Überprüfung der Drift des Zeitstempels im Smartphone wird diese Genauigkeit als ausreichend angenommen. Auf den Einsatz eines GNSS-Empfängers im Außenbereich kann daher verzichtet werden. Im Versuchsaufbau wird das Nexus 4 auf das RQH1003 System im Koordinatenmittelpunkt aufgesetzt und befestigt. Dabei wird auf eine möglichst parallele Achsausrichtung zwischen den Sensoren geachtet (Abb. 3.26). Um in beiden Zeitreihen diskrete Positionen für eine Ermittlung der Zeitunterschiede zu erhalten, wird der Aufbau in regelmäßigen Abständen angehoben.

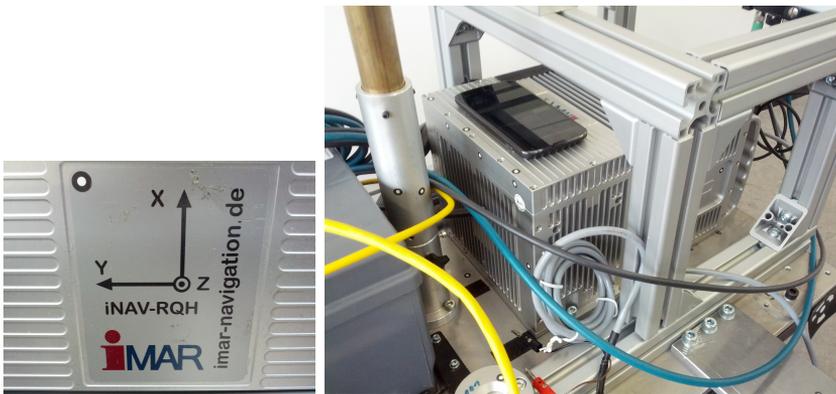


Abb. 3.26: Versuchsaufbau zur Überprüfung der Uhrendrift im Smartphone (links: Koordinatensystem INS, auf dem das Smartphone aufgesetzt wird, rechts: Versuchsaufbau).

Zur Bestimmung der Zeitdrift dienen die Beschleunigungen der jeweiligen z-Achse, da die z-Achse durch das Anheben des Versuchsaufbaus die größten Ausschläge verzeichnet und sich so diskrete Positionen sicher ableiten lassen. Im Smartphone wurden Daten mit einer Messrate von 200 Hz und im Referenzgerät mit 300 Hz gespeichert. Die Messdauer für den Versuch beträgt 15 Minuten. Die Datensätze werden mittels einer Matlab-Routine miteinander verglichen.

Zur Bestimmung des Startoffsets werden die Daten des Smartphones zur iMAR-Datenrate interpoliert um anschließend eine Kreuzkorrelation zu berechnen. Die Kreuzkorrelation liefert einen Korrelationskoeffizienten von 0,9954. Das belegt die große Übereinstimmung bei der Bewegungsdetektion. Der Offset von 6,4186 Sekunden wurde an den Datensatz des Smartphones angebracht. Abb 3.27 zeigt die

um das Offset korrigierten Rohdaten beider Systeme. Die hohen Beschleunigungen durch das Anheben der Inertialsensoren sind klar erkennbar.

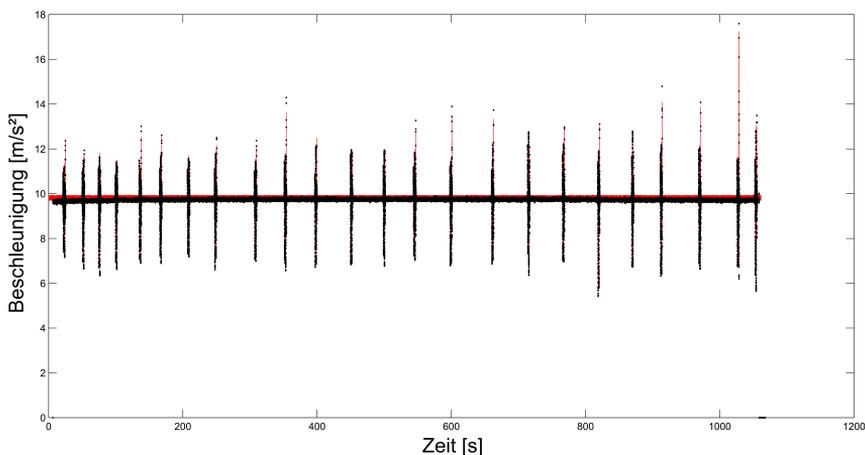


Abb. 3.27: Rohdaten beider z-Achsen der Beschleunigungssensoren aus der Versuchsdurchführung zur Uhrendrift. Die Referenzdaten (rot) und die Daten aus dem Smartphone (schwarz) nach Korrektur des Offsets.

Für die Bestimmung der Zeitdrift werden anschließend die minimalen und maximalen Werte der z-Achsen beider Messreihen gesucht. Die Gegenüberstellung der jeweiligen Zeitpunkte offenbart, ob eine Drift vorliegt. Dazu wird aus den Daten eine ausgleichende Gerade bestimmt. Mit ihr konnte eine signifikante Drift von $-57 \mu\text{s}/\text{s}$ ($\pm 3 \mu\text{s}/\text{s}$) ermittelt werden (Abb. 3.28). Wiederholungen des Versuchs belegten ein Rauschen um die ermittelte Drift von $\pm 20 \mu\text{s}/\text{s}$.

Für die Bestimmung der Auswirkungen dieser Drift auf die weitere Entwicklung wird eine typische maximale Navigationsdauer von 20 Minuten angenommen. Die oben ermittelte Zeitdrift verursacht nach 20 Minuten einen maximalen Zeitfehler von $< 0,1$ Sekunden. Da der Beschleunigungsmesser als Schrittzähler dient, ist der Einfluss in diesem Zusammenhang unbedeutend. Prinzipiell ist der Einfluss von Gyroskopdaten auf die Drehwinkelbestimmung nur schwer abzuschätzen, da positive und negative Drehraten möglich sind. Dadurch vermindert sich der Winkelfehler, wenn die Drehungen in gleicher Größenordnung in beide Richtungen erfolgen. Finden nur wenige Drehungen bei vergleichsweise niedriger Geschwin-

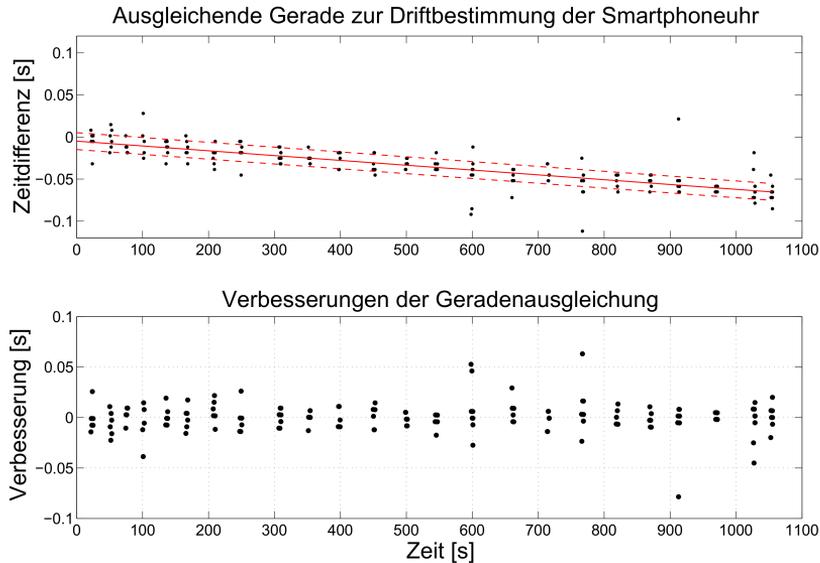


Abb. 3.28: Berechnung der Drift mittels ausgleichender Gerade (oben: Rohdaten mit ausgleichender Gerade (rot); unten: Verbesserungen).

digkeit statt, so ist der Fehlereinfluss der Zeitdrift zudem sehr gering. Aufgrund der hier angestrebten Genauigkeit von 1-5 m und die vergleichsweise geringen Auswirkung der Zeitdrift für den angegebenen Zeitraum sind Korrekturen der Zeitdrift zu vernachlässigen.

Zusammenfassung zur Untersuchung der Testgeräte im Hinblick auf die weitere Verwendung:

Die Untersuchung der Testgeräte hat aufgezeigt, dass die Sensordaten aktueller Geräte Rohdaten liefern, die mit weniger Unsicherheiten behaftet sind als Rohdaten von Vorgängermodellen. Auf eine Kalibrierung von Beschleunigungsmesser und Gyroskop kann daher verzichtet werden. Lediglich die ZUPT-Korrektur an den Gyroskopdaten wird in den nachfolgenden Untersuchungen verwendet. Die Rohdaten des Magnetfeldsensors unterliegen starken ferromagnetischen Einflüssen im Gebäude. Die Nutzung der Richtungsbestimmung aus Magnetfeldsensordaten ist daher ohne weitere Maßnahmen nicht sinnvoll. Die Luftdruckmessung zur Höhenbestimmung aus dem integrierten Barometer wird zwar durch Temperatur-

3.6 Qualität des Zeitstempels im Nexus 4

veränderung und Wetterdrift verfälscht, erlaubt aber trotzdem eine Stockwerksunterscheidung und wird daher in den folgenden Untersuchungen dazu verwendet.

4 Komponenten einer Smartphone gestützten hybriden Positionsschätzung

Die Möglichkeiten der Positionsschätzung in GNSS-abgeschatteten Bereichen sind vielfältig. Es lassen sich infrastrukturbasierte, hybride und bildbasierte Verfahren unterscheiden. Die hybriden Verfahren vereinen Inertialsensoren (INS) und infrastrukturbasierte Verfahren. In diesem Kapitel werden potentielle Komponenten der hybriden Positionsschätzung vorgestellt und diskutiert mit dem Ziel einer Einschätzung zur Anwendbarkeit im Hinblick der Minimierung von Aufwand und Kosten bei der Implementierung und Wartung. Zu Beginn wird das Pedestrian Dead Reckoning (PDR) vorgestellt, das die Grundlage für hybride bzw. autonome Anwendungen darstellt. Da das Smartphone als Navigationshardware im Fokus dieser Arbeit steht, kommen nur solche infrastrukturbasierten Technologien in Betracht, die mit einem modernen Smartphone verwendet werden können. Rein bildbasierte Ansätze werden nicht weiter untersucht, da sie eine kontrollierbare, gut ausgeleuchtete Umgebung für einen robusten Einsatz in der Positionsschätzung voraussetzen und meist auf Basis einer Feature-Erkennung arbeiten, die in monotonen Raumstrukturen nur unzureichend funktioniert.

4.1 Trajektorie aus Smartphone-Sensoren

Die Koppelnavigation beschreibt die Ortsbestimmung in Abhängigkeit von einer bekannten Startposition mithilfe von Richtung und Geschwindigkeit – eine Herangehensweise, die sich im geodätischen Sinne auch als *toter Polygonzug* bezeichnen lässt. Bei der kinematischen Vermessung mit INS liefert der Beschleunigungsmesser, der je nach Anwendung von einem Odometer unterstützt wird, den zurückgelegten Weg und das Gyroskop stellt die Drehwinkel zur Bestimmung der Bewegungsrichtung bereit. Im Folgenden wird das Pedestrian Dead Reckoning als Ansatz einer Koppelnavigation für Fußgänger auf Basis von MEMS-INS vorgestellt.

4.1.1 Pedestrian Dead Reckoning

Voraussetzung jeder Positionsschätzung per MEMS-INS ist die korrekte Übertragung der menschlichen Körperbewegungen auf das Smartphone, das jederzeit die Drehungen und Geschwindigkeiten registrieren können muss. Dies setzt eine stabile Ausrichtung des Geräts während der Navigation voraus. Wird dessen Lage zu sehr verändert, bleiben negative Effekte für die Positionsschätzung nicht aus.

In dieser Arbeit wird aus Gründen der Praxistauglichkeit von einer Ausrichtung ausgegangen, bei der der Nutzer das Smartphone vor dem Körper in der Hand hält. Fixieren die Augen während der Bewegung regelmäßig das Display, so wird die Bewegung einer Person weitgehend auf das Smartphone übertragen. Anders könnte der Nutzer das Display nicht lesen. Gegenüber sonstigen infrastrukturabhängigen oder foot-mounted-Methoden erzeugt diese Smartphone-Haltung, je nach Nutzerverhalten unterschiedlich stark, Unsicherheiten aufgrund von Veränderungen der Handhaltung oder Bewegungsabfolge. Dem entgegen steht die hohe Anwenderfreundlichkeit dieser Handhaltung sowie der geringe Aufwand für den Nutzer für den Übergang zu einer GNSS-gestützten Outdoor-Navigation.

Wie im Kapitelabschnitt *3.3 Kalibrierung eines Beschleunigungsmessers* dargestellt, eignet sich der Strapdown Algorithmus nicht zur Positionsschätzung mit MEMS-Sensoren. Stattdessen wird der MEMS-Beschleunigungsmesser zur Schritterkennung genutzt. Je nach dem Laufverhalten und der Art und Weise, wie das Smartphone dabei gehalten wird, unterscheiden sich die Bewegungsmuster, die der Beschleunigungssensor registriert. Der Schrittzähler muss möglichst so allgemeingültig entwickelt werden, dass er unterschiedliche Bewegungsmuster erfasst und nicht durch eine ungenaue Schritterkennung die Qualität der Positionsschätzung beeinträchtigt.

Beschleunigungsmesser als Schrittzähler

Vom Beschleunigungsmesser aufgezeichnete Schritte zeigen Beschleunigungen auf allen drei Achsen. Bei einem horizontalen Smartphone-Koordinatensystem ist besonders auf der z-Achse ein annähernd sinusförmiges Laufmuster vorhanden.

Für einen Schrittzähler auf Basis eines Beschleunigungssensors gibt es eine Reihe von Ansätzen, die unter anderem in [Hansson u. Tufvesson, 2011],[Renaudin u. a.,

2013],[Qian u. a., 2013] und [Gupta u. a., 2014] beschrieben sind. Sie unterscheiden sich in der Herangehensweise und dem Implementierungsaufwand teilweise erheblich voneinander. Zudem leiten einige Ansätze die Schrittlängen aus Beschleunigungen ab. In Gupta u. a. [2014] wird die Fourier-Transformation verwendet, um aus der maximalen Amplitude im Frequenzspektrum die Schrittfrequenz abzuleiten. Für eine echtzeitfähige Umsetzung wird dabei ein zurückliegendes Zeitfenster von einer Sekunde für die Fourier-Analyse favorisiert. Die Schrittzähler, wie sie Hansson u. Tufvesson [2011] und Qian u. a. [2013] vorstellen, funktionieren auf der Basis zuvor definierter Bedingungen, die für eine Schritterkennung nach einer bestimmten Reihenfolge erfüllt sein müssen. So müssen beispielsweise für die Beschleunigungswerte bestimmte Grenzwerte überschritten werden. Ein weiterer Ansatz findet sich bei Kupke u. a. [2016]. Er untersucht die Schritterkennung auf Grundlage künstlicher neuronaler Netze (KNN).

All diese Beispiele führen zu Ergebnissen unterschiedlicher Qualität. Der Ansatz der Mustererkennung auf Basis von KNN etwa ist mit über 80 Datensätzen von 40 Personen trainiert worden und kommt zu Abweichungen von lediglich 0.1 % der Gesamtschrittzahl. Jedoch gestaltet sich die Implementierung auf einem Smartphone vergleichsweise schwierig. Zudem ist die erforderliche Prozessorleistung – je nach Anzahl der verwendeten Knoten im künstlichen neuronalen Netz – hoch. Eine Fourier-Analyse benötigt nicht nur relativ viel Rechenzeit, ihre Qualität ist auch von dem gewählten Zeitfenster für die Fourier-Transformation abhängig. Besonders die Übergänge zwischen Stillstand und Gehen sind auf diese Weise schlecht zu detektieren, da der Untersuchungsbereich nur zum Teil einer Sinusschwingung entspricht. Eine Implementierung nach dem von Hansson u. Tufvesson [2011] und Qian u. a. [2013] vorgeschlagenen Verfahren ist demgegenüber zwar leicht zu implementieren und mit wenigen Parametern einfach anzupassen, erreicht vom Ergebnis her aber nicht die Qualität des KNN-Ansatzes.

Aufgrund der einfachen Implementierbarkeit wird ein Schrittzähler in Anlehnung an Hansson u. Tufvesson [2011] und Qian u. a. [2013] favorisiert, obwohl eine geringfügig schlechtere Qualität gegenüber dem KNN-Ansatz zu erwarten ist. Für eine Schritterkennung sind zwei Zustände an den Daten der horizontalen z-Achse des Beschleunigungsmessers zu erfüllen. Damit die Schrittdetektion er-

4.1 Trajektorie aus Smartphone-Sensoren

folgen kann, muss der Schrittzähler möglichst variabel an das Nutzerverhalten angepasst werden. Abb. 4.1 zeigt ein mit dem Nexus 4 aufgezeichnetes Schrittmuster. Das Testgerät wurde dabei, wie oben dargelegt, in der Hand gehalten und das Smartphone-Koordinatensystem in den Horizont transformiert berechnet. Der folgend verwendete Schrittzähler ist bereits in Willemsen u. a. [2015b] beschrieben.

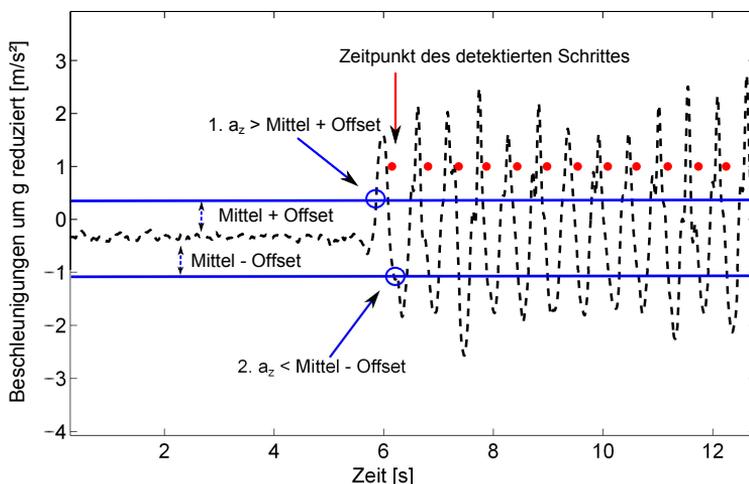


Abb. 4.1: Prinzip der Schritterkennung aus Beschleunigungen der z-Achse (schwarz) nach Verebnung des Smartphone-Koordinatensystems und Reduzierung um g (rot: Schritt erkannt; blau: Offset abhängig von in Ruhelage berechnetem Mittel der Beschleunigungen) [Willemsen u. a., 2015b].

Die schwarz gestrichelte Linie in Abb. 4.1 repräsentiert die um g reduzierte und horizontierte z-Achse des Beschleunigungssensors im Nexus 4. Die roten Punkte markieren die Zeitpunkte einer Schritterkennung. Die Schritterkennung erfolgt nach positiver Rückmeldung zweier aufeinanderfolgender Bedingungen: Die erste ist nach dem Überschreiten eines positiven Grenzwerts erfüllt, anschließend liefert die zweite eine Rückmeldung, sobald der Beschleunigungswert einen Grenzwert unterschreitet. Diese beiden Grenzwerte sind in Abb. 4.1 durch blaue Linien dargestellt.

Damit die Schritterkennung bei unterschiedlichen Laufmustern problemlos funktioniert, müssen die Grenzwerte in Abhängigkeit der Sensordaten bestimmt wer-

den. In der Initialisierungsphase ergeben sie sich aus den Messdaten während des Ruhezustands. Dazu wird ein Mittelwert aus den ersten gemessenen Sekunden, in denen das Smartphone in Ruhelage ist, berechnet. Für die jeweiligen Laufmuster wird dann ein Offset addiert und subtrahiert, um den maximalen und minimalen Grenzwert zu ermitteln. Dessen Höhe muss zu Beginn definiert werden und einen Wert erhalten, der höher als das berechnete Sensorrauschen der ersten Sekunden ist, um ein fehlerhaftes Detektieren von Schritten zu verhindern. Danach werden die Grenzwerte in Abhängigkeit von den minimalen und maximalen Werten der z-Achse regelmäßig an das jeweilige Schrittmuster angepasst. Die Spanne zwischen minimalen und maximalem Wert dient dabei als Skalierungsfaktor. Dieser Schrittzähler erlaubt eine Anpassungen an das Laufverhalten des Nutzers sowie an die Qualität des Beschleunigungsmessers.

Damit beim Übergang zwischen Stillstand und Bewegung nicht fälschlicherweise Schritte registriert werden, ist es notwendig, die oben genannte erste Bedingung nach einem bestimmten Zeitraum zurückzusetzen, sofern währenddessen die zweite Bedingung unerfüllt bleibt. Als Zeitraum wird eine maximale Schrittdauer von 1 s angenommen.

Schrittlängenschätzung

Für das PDR ist neben der Schritterkennung und der Richtungsbestimmung auch die Schrittlänge wichtig. Sie bestimmt die Skalierung der zu berechnenden Trajektorie. Die Schrittlänge ist dem Nutzer in aller Regel nicht bekannt, muss aber vorab festgelegt oder während der Positionsschätzung geschätzt werden. Dazu bietet sich die Schätzung mithilfe des MEMS-Beschleunigungsmessers an, der auch zur Detektion der Schritte dient.

Nach den bislang zum Thema veröffentlichten Ansätzen sind Schrittlängenschätzungen auf Basis von Beschleunigungsmessern nicht immer sehr präzise. Sie erreichen lediglich Genauigkeiten von ca. 7,0 %, wie sie für unterschiedliche Verfahren beispielsweise in Pratama u. a. [2012] erreicht werden. Zudem muss einschränkend erwähnt werden, dass diese Angaben nicht wirklich repräsentativ sind, weil sie oft nur konstruierten, nicht wirklich realitätsgemäßen Testbedingungen unterliegen. Die Annahme einer Schrittlängengenauigkeit von 10 % für die individuelle Schrittlängenschätzung erscheint daher plausibler.

Schrittlängenschätzungen beruhen oftmals auf der Annahme von Korrelationen der Schrittlänge zur Amplitude und Frequenz der Schrittdetektion sowie zu persönlichen Merkmalen wie Größe und Geschlecht des Nutzers. Dafür werden folgende Annahmen getroffen:

- Hohe Schrittfrequenz = Große Schrittlänge
- Hohe Amplitude = Große Schrittlänge
- Große Person = Große Schrittlänge
- Schrittlänge Frau \neq Schrittlänge Mann

Beispiele für die funktionalen Zusammenhänge zur Schrittlängenschätzung finden sich bei Bylemans u. a. [2009] und Pratama u. a. [2012]. Pratama u. a. [2012] vergleicht vier Ansätze miteinander. Auffällig bei nahezu allen dieser Ansätze ist die Schätzung eines k – *Faktors*, der einen Skalierungsparameter zwischen funktionalen Modell und Schrittlänge darstellt. Dieser Faktor ist aus einer entsprechend großen Stichprobe von Nutzern zu ermitteln und in Abhängigkeit zu den oben aufgeführten Annahmen für unterschiedliche Nutzergruppen berechenbar. Die Qualität der Schrittlängenschätzung ist daher einerseits vom funktionalen Modell und andererseits von den Trainingsdaten zur Bestimmung des k – *Faktors* abhängig.

Der erste von Pratama u. a. [2012] aufgeführte Ansatz entspricht einer statischen Methode. Sie beruht auf der Annahme einer größenabhängigen festen Schrittlänge l_{step} , die durch Multiplikation mit einem zuvor geschätzten k – *Faktor* errechnet wird (36).

$$l_{step} = h_{body} * k \tag{36}$$

Die weiteren funktionalen Zusammenhänge der Schrittlängenberechnung zeigen die Gleichungen nach Weinberg (37), nach Scarlett (38) und nach Kim (39). Die Beschleunigungen sind mit a_i , der maximale bzw. der minimale Beschleunigungswert eines Schritts mit a_{max} und a_{min} bezeichnet.

Die Schätzung des k – *Faktors* erfolgt mithilfe dieser Gleichungen und Referenzdaten aus entsprechenden Stichproben. Zur Bestimmung der für die Nutzergrup-

pen spezifizierten k – *Faktoren* werden die Referenzdaten zuvor entsprechend aufgeteilt. In Pratama u. a. [2012] wird für die Formel (36) der k – *Faktor* gesondert nach Geschlecht mit 0,415 für Männer und 0,413 für Frauen angegeben. Weinbergs Ansatz zufolge (37) wird der maximale Messausschlag des Beschleunigungsmessers im Verhältnis zur Schrittlänge verwendet und der k – *Faktor* mit 0,41 geschlechterunabhängig angegeben.

$$l_{step} = k * \sqrt[4]{a_{max} - a_{min}} \quad (37)$$

Für Scarletts Ansatz (38) wird der kleinste Beschleunigungswert eines Schrittes vom absoluten arithmetischen Mittels subtrahiert und das Ergebnis mit dem maximalen Messausschlag dividiert. Der k – *Faktor* wird hier geschlechterunabhängig mit 0,81 angegeben Pratama u. a. [2012].

$$l_{step} = k * \frac{\sum_{i=1}^N |a_i| - a_{min}}{a_{max} - a_{min}} \quad (38)$$

Im Kims Ansatz (39) wird die dritte Wurzel des arithmetischen Mittels der absoluten Beschleunigung eines Schritts mit dem k – *Faktor* multipliziert. Der k – *Faktor* beträgt nach Pratama u. a. [2012] 0,55.

$$l_{step} = k * \sqrt[3]{\frac{\sum_{i=1}^N |a_i|}{N}} \quad (39)$$

Den funktionalen Zusammenhang für die Schrittlängenschätzung nach Bylemans u. a. [2009] zeigt die Gleichung (40). Das absolute arithmetische Mittel, der k – *Faktor*, der maximale Messausschlag des Beschleunigungsmessers sowie die Schrittdauer Δt werden in diesem Ansatz verwendet.

$$l_{step} = k * \sqrt[2.7]{\frac{\sum_{i=1}^N |a_i|}{N}} * \sqrt{\frac{k}{\Delta t * (a_{max} - a_{min})}} * 0,1 \quad (40)$$

Die Nutzbarkeit einer Schrittlängenschätzung auf Basis dieser Gleichungen mit MEMS-Beschleunigungen hängt von der erzielbaren Genauigkeit ab. In den oben

angeführten Veröffentlichungen finden sich aber keine stichhaltigen Angaben dazu, da sie auf nicht repräsentativen Stichproben beruhen. Für eine Qualitätseinschätzung auf Basis einer größeren Stichprobe wird in Kupke u. a. [2016] der Ansatz nach Bylemans u. a. [2009] favorisiert.

Die große Variation der vorgestellten funktionalen Zusammenhänge zur Schätzung der Schrittlänge mittels Beschleunigungen zeigt zudem die hohe Unsicherheit für die Festlegung eines Zusammenhangs zwischen Schrittlänge und MEMS-Beschleunigungen. Darum wird ein Ansatz mit einem künstlichen neuronalen Netz realisiert, der eine Schrittlängenschätzung ohne großes Vorwissen über den funktionalen Zusammenhang ermöglicht. Dazu wird eine Matlab-Toolbox verwendet, die auf Basis der Mustererkennung ein feed-forward Netzwerk entwickeln lässt. In das KNN gehen die Körpergröße und die Daten des Beschleunigungsmessers ein [Kupke u. a., 2016]. Das KNN wird daher mit sieben input layer, dem k – Faktor als output layer und zwei zusätzlichen hidden layer aufgebaut. Für die Untersuchung stehen Referenzdaten von 40 Personen, für die Referenz-Schrittlängen Zentimeter-genau bekannt sind, mit insgesamt 80 Datensätzen zur Verfügung.

Die Ergebnisse für den KNN-Ansatz sowie für den Ansatz nach Bylemans u. a. [2009] zeigen Unsicherheiten (95%) in der Schrittlängenschätzung von 9 – 10 % der Schrittlänge auf. Bei einer mittleren Schrittlänge von 0,75 m bedeutet das eine Unsicherheit von $\pm 0,075$ m. Die geschätzte Schrittlänge liegt daher in einem Unsicherheitsbereich zwischen 0,68 m und 0,83 m. Das bedeutet, dass auch eine aufwendiger durchgeführte Schätzung nicht zu deutlich besseren Resultaten – zumindest für die Beschleunigungen – führt.

Da sich die Schrittlängen je nach Tagesform des Nutzers oder der Nutzerin ändern können und auch abhängig von der getragenen Kleidung sind, wird für diese Arbeit eine mittlere Schrittlänge vorausgesetzt. Die Positionsschätzung muss dann allerdings die Schrittlänge zusätzlich schätzen oder der Algorithmus muss die Unsicherheiten aufgrund fehlerhafter Schrittlänge kompensieren können.

Drehwinkel aus Gyroskopdaten

Für das Pedestrian Dead Reckoning liefert das Gyroskop Drehraten \vec{r} , die zu Drehwinkeln \vec{r} integriert werden (41):

$$\vec{r}_i = \vec{r}_{i-1} + \vec{r} * \Delta t \quad (41)$$

Die Drehraten des Gyroskops werden für jede Messung mit der Rotationsmatrix R_{xy} (43), die die Neigungen aus den Beschleunigungsmessungen enthält, in ein horizontales Koordinatensystem rotiert. Anschließend ist nur noch die Drehung um die z-Achse für das Pedestrian Dead Reckoning notwendig. Dies erlaubt dem Nutzer eine freie Neigungshaltung seines Smartphones. Die Drift, hervorgerufen durch das farbige Rauschen, wird zu Beginn einer Aufzeichnung per ZUPT-Befehl minimiert.

Zustandserkennung während der Positionsschätzung

Die Detektion von Zuständen während der Navigation erlaubt einen Eingriff in die Positionsschätzung, etwa die Aktivierung des ZUPT-Befehls in Ruheposition. Folgende Zustände sind für diese Arbeit von Interesse:

- Gehen
- Stehen
- Treppensteigen
- Fahrstuhlfahren

Das *Gehen* wird vom Schrittzähler erkannt und bedeutet eine Positionsänderung in Abhängigkeit von der Orientierung in der Horizontalen. *Stehen* definiert einen Positionstillstand, der durch den Schrittzähler aufgrund unwesentlicher Veränderungen der Beschleunigungen erkannt wird. In diesem Zustand kann beispielsweise ein ZUPT-Befehl zur Korrektur des farbigen Rauschens im Gyroskop angewendet werden.

Die Höhenänderung in einem Gebäude erfolgt entweder über eine *Treppe* oder einen *Fahrschein*. Zur ihrer Detektion dient das Barometer. In Kombination mit dem Schrittzähler ist die Unterscheidung zwischen einer Treppe und einem Fahrschein möglich, da auch beim Treppensteigen Schrittmuster aufgezeichnet werden. Eine Korrektur von Richtung und Position der MEMS-INS-basierten Positionsschätzung bietet sich besonders beim Wechsel der Stockwerke an, da die Bewegungsrichtung üblicherweise klar vorgegeben ist. Besonders beim *Treppensteigen*

ist die frühzeitige Erkennung der Treppe notwendig, um die Schrittlänge der Stufenlänge anzupassen, da die gelaufene Trajektorie ansonsten zu lang geschätzt wird.

Pedestrian Dead Reckoning

Für die Berechnung werden Schritterkennung, Schrittlänge und Orientierung einer relativen Position vereint. Damit der Nutzer die Neigung des Smartphones individuell einstellen und sich das Bezugskordinatensystem parallel zu den Stockwerken ausrichten kann, wird das Smartphone-Koordinatensystem horizontalisiert. Die Stockwerksunterscheidung selbst basiert, wie bereits dargelegt, auf Luftdruckmessungen. Auf diese Weise ist das Pedestrian Dead Reckoning auf unterschiedlichen Stockwerksebenen möglich.

Die Beschleunigungen dienen der Berechnung der Neigung des Smartphones zum Horizont. Mit der Formel (42) werden die Neigungswinkel r_x und r_y in Abhängigkeit aller drei Achsen des Beschleunigungsmessers a_x, a_y, a_z berechnet.

$$\begin{aligned} r_x &= \arctan\left(\frac{a_y}{a_z}\right) \\ r_y &= \arctan\left(\frac{a_x}{a_z}\right) \end{aligned} \tag{42}$$

Dann werden die Neigungswinkel r_x und r_y in die Rotationsmatrix R_{xy} eingefügt, damit die Sensordaten des Gyroskops und des Beschleunigungsmessers durch Multiplikation mit der Matrix in Formel (43) [Alyafawi u. a., 2015] in das verebnete Koordinatensystem rotiert werden können. Für das PDR wird jetzt nur noch die z-Achse (r_z) des Gyroskops und die z-Achse des Beschleunigungssensors die Schrittdetektion verwendet.

$$R_{xy} = \begin{pmatrix} 1 & \sin(r_x) * \tan(r_y) & \cos(r_x) * \tan(r_y) \\ 0 & \cos(r_x) & -\sin(r_x) \\ 0 & \sin(r_x)/\cos(r_y) & \cos(r_x)/\cos(r_y) \end{pmatrix} \tag{43}$$

Die Berechnung der relativen 2D-Koordinaten erfolgt nach jeder Schrittdetektion mit Formel (44).

$$\vec{x}_i = \vec{x}_{(i-1)} + R_z * t \quad (44)$$

Die Rotationsmatrix R_z (45) enthält die Drehwinkel um die z-Achse, und t (46) ist der Vektor für die Schrittlänge. Je nach Handhaltung des Smartphones zum Nutzer kann in t die Position von l_{step} jeweils für die Bewegungsrichtung in x- bzw. y-Richtung ausgetauscht werden.

$$R_z = \begin{pmatrix} \cos(r_z) & \sin(r_z) \\ -\sin(r_z) & \cos(r_z) \end{pmatrix} \quad (45)$$

$$t = \begin{pmatrix} l_{step} \\ 0 \end{pmatrix} \quad (46)$$

Das PDR wird bei jeder Schritterkennung aktiv und verwendet den aktuellen Winkel r_z , die Schrittlänge l_{step} sowie die vorherige Position $\vec{x}_{(i-1)}$, um neue Koordinaten \vec{x}_i zu berechnen. Systematische Unsicherheiten in den einzelnen Komponenten, beispielsweise die Drift aus Gyroskop-Daten, werden so mitgeführt und summieren sich auf.

Zur Stockwerksunterscheidung dienen die Höheninformationen aus den Luftdruckmessungen des Barometers. Dazu wird die barometrische Höhenformel nach h umgestellt (23). In (23) wird die internationale Standardatmosphäre (Temperatur 15 °C = 288,15 K, Luftdruck 1013,25 hPa, Temperaturgradient 0,65 K pro 100 m Höhenunterschied) als mittlerer Zustand eingeführt. Relative Höhen werden auf Basis eines Anfangsmittels, das zu Beginn einer Positionsschätzung bestimmt wird, berechnet.

4.1.2 Berechnung einer Trajektorie (2D + 1D)

Der im vorherigem Kapitelabschnitt 4.1.1 *Pedestrian Dead Reckoning* beschriebene Aufbau zum PDR wird nun für einen Datensatz im Gebäude D des ehemaligen HCU-Gebäudes umgesetzt. Mit dem Testgerät Galaxy Nexus wurde im zweiten Obergeschoss eine Strecke abgelaufen. Start und Ende der Trajektorie befinden

sich an einer Position. Während des Rundgangs befand sich das Smartphone in leichter Neigung zum Nutzer, damit der Nutzer das Display komfortabel betrachten konnte. Zu Beginn wurde das Smartphone ca. 5 Sekunden ruhig in der Hand gehalten und das ZUPT berechnet. Anschließend wurde der Rundgang im Uhrzeigersinn abgelaufen.

Die zurückgelegte Strecke beträgt 80 m und die Laufzeit 60 Sekunden. Die berechnete Trajektorie ist in Abb. 4.2 dargestellt. Sie wurde an der bekannten Startposition initialisiert. Die Schrittlänge wird für dieses Beispiel als bekannt angenommen. In der Abbildung steht die blaue Trajektorie für das PDR vor der ZUPT-Korrektur, die Grüne stellt die um den ZUPT korrigierten Ergebnisse dar.

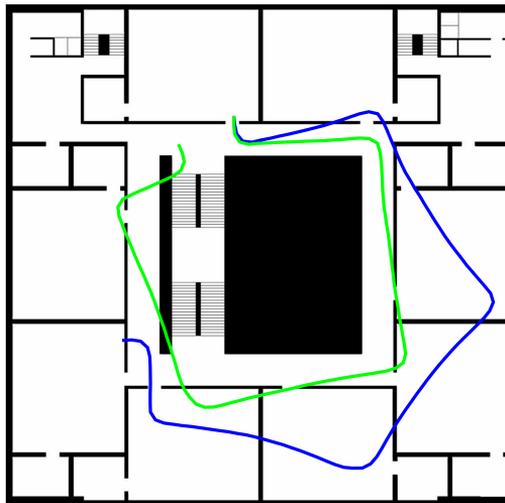


Abb. 4.2: Pedestrian Dead Reckoning mit Sensordaten des Galaxy Nexus im zweiten Obergeschoss des ehemaligen HCU Gebäudes (grün: mit ZUPT-Korrektur; blau: ohne ZUPT-Korrektur) [Willemsen u. a., 2015a].

Die Abbildung belegt eine eindeutige Verbesserung durch den ZUPT. Zum Ende der Trajektorie ergibt sich dennoch nach nur 60 Sekunden ein Positionsfehler von ca. 4 m. Das zeigt, dass neben der Unsicherheiten der Schrittlängenschätzung die Unsicherheit der Richtungsbestimmung ein weiterer großer Einfluss der MEMS-basierten Positionsschätzung ist.

In Abb. 4.3 ist der Drehwinkel r_z für die gesamte Trajektorie sichtbar. Blau ist der noch mit dem Drehratenoffset behaftete Drehwinkel, schwarz der korrigierte

Datensatz dargestellt. Linksseitige Drehungen erzeugen in diesem Beispiel positive Winkel. Daran sind die vier rechtsseitigen und die zwei linksseitigen Drehungen des Testlaufs zu erkennen. Das minimale Rauschen, während keine Richtungsänderung stattfindet, wird durch die Schrittbewegung hervorgerufen.

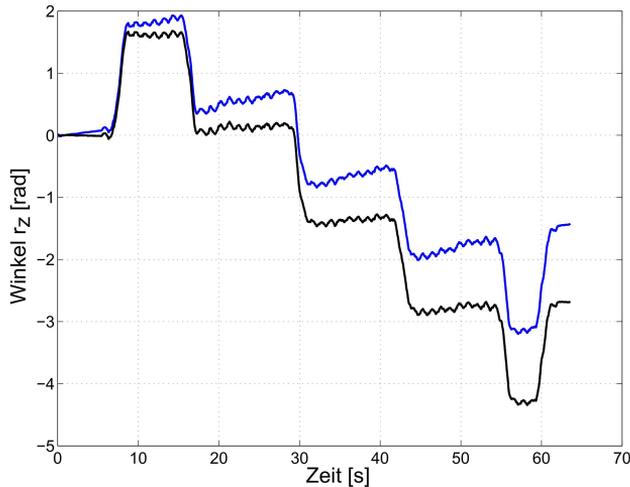


Abb. 4.3: Verwendeter Drehwinkel r_z aus den Testdaten für das Dead Reckoning in Abb. 4.2 (blau: ohne Korrektur des Drehratenoffsets; schwarz: mit Korrektur).

Die Verbesserung der Drehwinkel per ZUPT beträgt am Ende der 60 Sekunden langen Trajektorie ca. 1,2 rad (69°). Unter der Annahme, dass das Gebäude rechtwinklig ist, beträgt der Drehwinkelfehler in diesem Beispiel trotz ZUPT nach nur 60 Sekunden immer noch 0,46 rad (26°). Diese hohe Differenz nach so kurzer Zeit macht deutlich, wie wichtig zusätzliche Stützinformationen für eine MEMS-INS-basierte Fußgängernavigation sind.

Für eine Einschätzung der Einflussfaktoren auf die Berechnung der Trajektorie wird folgend eine Varianz-Kovarianzfortpflanzung (VKF) mit den bekannten Unsicherheiten am Beispiel der vorliegenden Daten umgesetzt. Die Unsicherheit der Orientierung aus Gyroskopdaten wird aus einem zufälligen und einem maßstäblichen Anteil berechnet, zudem erfolgt die Berücksichtigung der Schrittlängenunsicherheit.

Der funktionale Zusammenhang ist das PDR mit der zuvor beschriebenen For-

mel (44). In die VKF gehen die Positionsgenauigkeiten der Ausgangsposition S_x und S_y ein. Die Unsicherheit der Orientierung wird mit S_r , sowie die Schrittlängenunsicherheit mit S_l bezeichnet. In Matrixschreibweise wird die VKF nach folgender Formel berechnet (47):

$$Q_x = F * Q_F * F^T \quad (47)$$

Die partiellen Ableitungen des funktionalen Modells nach den eingehenden Beobachtungen werden in der Funktionalmatrix F wie folgt eingebunden (48):

$$F = \begin{pmatrix} 1 & 0 & -\sin(r_z) * l_{step} & \cos(r_z) \\ 0 & 1 & \cos(r_z) * l_{step} & \sin(r_z) \end{pmatrix} \quad (48)$$

Die Varianzen der geschätzten Unsicherheiten werden in der Kovarianzmatrix Q_F eingeführt (49):

$$Q_F = \begin{pmatrix} S_x^2 & 0 & 0 & 0 \\ 0 & S_y^2 & 0 & 0 \\ 0 & 0 & S_r^2 & 0 \\ 0 & 0 & 0 & S_l^2 \end{pmatrix} \quad (49)$$

Als Kenngröße der Genauigkeit wird der Helmertsche Punktfehler mit einem Konfidenzniveau von 95% verwendet, da sich die Unsicherheiten der Koordinatenanteile in Abhängigkeit der Orientierung verändern. In Abb. 4.4 wird er für jede Schritterkennung der Teststrecke aufgezeichnet. Die schrittbezogenen Helmertschen Punktfehler in rot repräsentieren die Unsicherheiten bei zusätzlich fehlender Korrektur des ZUPT. Hierzu wurde der maßstäbliche Anteil der Richtungsgenauigkeit in der VKF durch die reale ZUPT-Korrektur ersetzt, sodass sich die Richtungsgenauigkeit in gleicher Größe fortpflanzt. In blau und grün sind die Unsicherheiten nun nach Richtungs- und Schrittanteil aufgeteilt dargestellt. Die in cyan dargestellten Ergebnisse repräsentieren eine Trajektorie, die um den ZUPT korrigiert wurde und daher der maßstäbliche Anteil geringer ausfällt. Die eingestellten

Ausgangsgenauigkeiten sind in Anlehnung der berechneten Trajektorie im Ruhezustand abgeschätzt, um die Relationen herzustellen. Für die Schrittlänge wird eine Unsicherheit von 10% angesetzt. Der zufällige Anteil der Richtungsgenauigkeit beträgt $0,3^\circ$ und der maßstäbliche Anteil wird aus der ZUPT-Korrektur mit $0,6^\circ$ pro Schritt berechnet. Bei erfolgter ZUPT-Korrektur wird eine Rest-Unsicherheit von $0,1^\circ$ pro Schritt angenommen.

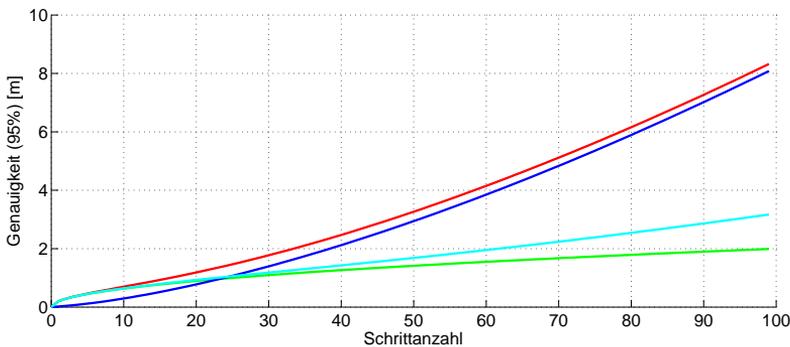


Abb. 4.4: Genauigkeiten (Helmertscher Punktfehler mit 95 % Konfidenzniveau) der Trajektorie aus Varianz-Kovarianzfortpflanzung (rot: gesamt; grün: nur Anteil Schrittlänge; blau: nur Anteil Gyroskop-Orientierung, cyan: mit ZUPT Korrektur).

Für die beiden berechneten Trajektorien (mit und ohne ZUPT) sind die berechneten Genauigkeiten im Vergleich zu den Ergebnissen des PDR zu optimistisch geschätzt. Die Abweichung des Endpunktes der Trajektorie mit ZUPT-Korrektur beträgt ca. 4 m und liegt damit deutlich über der VKF-Schätzung (3m, cyan). Für die Genauigkeitsabschätzung der Trajektorie ohne ZUPT-Korrektur werden zum Abschluss 8 m (blau) geschätzt, aber in den realen Daten liegt die Abweichung abschließend bei ca. 14 m. Es ist daher von weiteren Einflüssen auf die Sensorgenauigkeiten auszugehen. Aufgrund des deutlich sichtbaren Schrittmusters in den Drehwinkeln ist anzunehmen, dass diese Einflüsse weitestgehend von der Trageweise der Testgeräte in Verbindung mit dem Nutzerverhalten entstehen. Die Schrittbewegungen erzeugen ein erhöhtes Rauschen in den ermittelten Drehraten.

Durch die Trennung von Richtungs- und Schrittlängenunsicherheit lässt sich besonders zu Beginn die Auswirkung der Schrittlängenunsicherheit auf die Positionsgenauigkeit erkennen, während nach ca. 25 Schritten der Einfluss der Orien-

tierungsunsicherheit überwiegt. Es bleibt daher festzuhalten, dass das PDR weitestgehend durch die Drift des Gyroskops beeinflusst ist und auch ein ZUPT zwar Verbesserung bringt, aber die bestehenden äußeren Einflüsse die Positionsberechnung weiter verschlechtern.

Die relativen Höhen aus Luftdruckmessungen gibt Abb. 4.5 wieder. Die Werte wurden mit der barometrischen Höhenformel (23) berechnet und mit einem gleitenden Mittel zu einer Sekunde geglättet. Eine hohe Abweichung des Initialisierungswertes vom Mittelwert erzeugt daher, wie hier zu sehen, ein Ansteigen bzw. Absinken der Höhen zu Beginn des gleitenden Mittels. Bei genauer Betrachtung deutet das anschließend um ca. 0,5 m rauschende Signal einen positiven Trend an, der allerdings aufgrund der kurzen Messdauer nicht sicher belegt werden kann. Die Daten zeigen, dass das Stockwerk nicht gewechselt wurde. Bei dem Testgelände handelte es sich um einen offenen Flur, bei dem optimale Bedingungen für die relative Ermittlung des Luftdrucks bestanden.

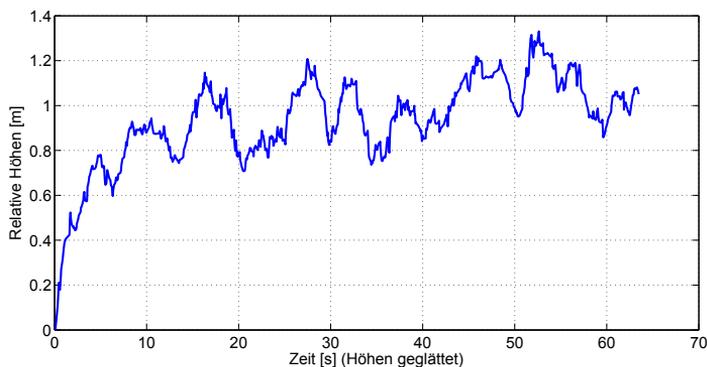


Abb. 4.5: Höhen aus relativen Luftdruckmessungen abgeleitet während eines Testlaufs im zweiten Obergeschoss des ehemaligen HCU-Gebäudes.

Die Ergebnisse des PDR mit Höhenberechnung zeigen, dass eine Positionsschätzung mittels den MEMS-INS zwar möglich ist, dass aber Messunsicherheiten die Positionsgenauigkeit mit andauernder Schätzung erheblich verschlechtern. Eine regelmäßige Korrektur der Positionsschätzung durch die Kombination der internen, stochastisch unterschiedlichen Sensoren oder durch externe Informationen auf Basis der vorhandenen Infrastruktur ist daher nötig.

4.2 Infrastrukturabhängige Positionsschätzung

Eine hybride Positionsschätzung bezeichnet die Kombination einer MEMS-INS-basierten Positionsschätzung, die mit durch infrastrukturbasierte Verfahren gestützt wird. Als infrastrukturabhängig werden alle Verfahren zur Positionsschätzung bezeichnet, die auf Basis von im Gebäude installierter Hardware umgesetzt werden können. In dieser Arbeit werden aktuelle Smartphones als Navigationshardware zugrunde gelegt, so dass sich die möglichen Verfahren, die in dieser Arbeit für die MEMS-INS-basierte Positionsschätzung zur Stützung herangezogen werden, auf die in ihnen verbauten Sensoren eingrenzen lassen.

4.2.1 Unterscheidung und Überblick der bekanntesten Verfahren

Der überwiegende Teil der infrastrukturbasierten Verfahren setzt Funksignale ein; aber auch Ultraschall, Infrarot und künstliche Magnetfelder sind mögliche Technologien. Im Folgenden werden die bekanntesten Berechnungsmethoden der infrastrukturabhängigen Ansätze beschrieben. Anschließend werden die Technologien erläutert und den Methoden der Positionsschätzung zugeordnet. Zur Übersicht hier nun eine Auflistung der in diesem Kapitel beschriebenen Berechnungsmethoden bzw. Technologien:

Methoden:

- Cell of Origin
- Weighted Centroid Localization
- Bluetooth
- Trilateration
- Triangulation
- Polares Anhängen

Technologien:

- WLAN
- Fingerprinting
- Ultraschall
- Magnetfeld
- RFID
- Infrarot
- LTE
- Ultra Wide Band
- GPS Repeater

Annäherung nach dem Prinzip Cell of Origin

Cell of Origin (COO) nennt sich ein Verfahren der mobilen Zellortung, beispielsweise per GSM im Außenbereich. Die Positionsschätzung erfolgt über die Zuordnung der Koordinaten der empfangenen Sendestation [Blankenbach, 2007]. Das COO ist prinzipiell für Technologien, die auf direkten Funkkontakt oder einem Signalempfang begrenzter Reichweite basieren, einsetzbar. Funkbasierte Technologien setzen eine Überlagerung der kreisförmig ausgesandten Signale voraus, damit Funklöcher vermieden werden, die eine Positionsschätzung unmöglich machen (Abb. 4.6). Dies ist besonders innerhalb von Gebäuden wichtig, da es je nach Bauart und verarbeiteten Materialien zu unterschiedlich starken Dämpfungen der Signalausbreitung kommt. Beim Empfang mehrerer Signale stellt das Smartphone die Verbindung zu der Zelle mit der größten Signalstärke her. Für Anwendungen, die lediglich Informationen von raumgenauer Qualität benötigen, könnte solch ein Ansatz durchaus von Interesse sein.

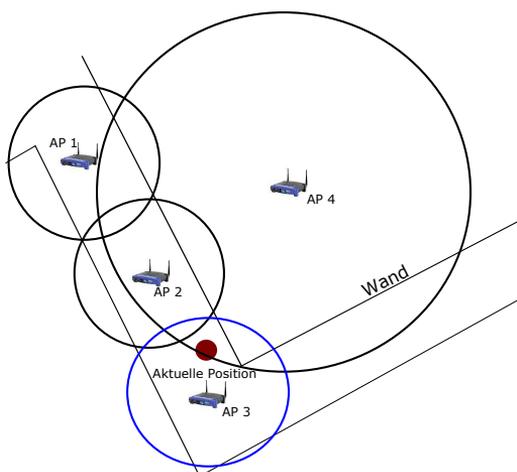


Abb. 4.6: Positionsbestimmung nach dem Prinzip der Annäherung und Cell of Origin für funkbasierte Verfahren (abgeleitet nach [Blankenbach, 2007]).

Weighted Centroid Localization (WCL)

Gegenüber dem COO, in dem der AP mit der stärksten Signalstärke als Position ausgewählt wird, erfolgt im *Weighted Centroid Localization* die Positionsschätzung durch eine gewichtete Mittelbildung aller verfügbaren AP. Die Berechnung der Gewichte erfolgt in Abhängigkeit einer Distanz, die beispielsweise aus Signal-

stärken abgeleitet werden kann und dadurch das Gewicht in Abhängigkeit der Signalstärkeausbreitung berechnet wird. AP nahe der zu schätzenden Position erhalten dadurch ein erhöhtes Gewicht in der Mittelbildung. Das Verfahren ist vergleichsweise leicht zu realisieren, da nur distanzabhängige Gewichte zu berechnen sind und anschließend nur eine Mittelbildung der AP-Koordinaten erfolgen muss [Blumenthal u. a., 2007].

Szenenanalyse mit Fingerprinting

Die Szenenanalyse als Methode zur Positionsbestimmung basiert in der Regel auf einer großen Anzahl empfangener Signale [Blankenbach, 2007], die zusammengenommen ein Signalmuster, einen einmaligen, unverwechselbaren *Fingerprint* repräsentieren. Das Fingerprinting-Verfahren wird häufig bei WLAN Netzwerken eingesetzt, da sich WLAN-Router heute in praktisch allen Gebäuden befinden und Signale senden. Es können aber auch Bluetooth oder andere Funksignale genutzt werden [Zekavat u. Buehrer, 2012]. Damit das Fingerprinting funktioniert, muss sich eine große Anzahl von Sendern, im Folgenden als Access Points (AP) bezeichnet, in dem Gebäude befinden. Die Positionsbestimmung erfolgt über eine Referenzdatenbank, die die Koordinaten bekannter Positionen und die dort empfangenen Signalstärken der AP enthält. Die Abschwächung der Signalstärke bei Funksignalen ist im freien Raum logarithmisch in Abhängigkeit zur Entfernung. Künstliche Hindernisse, wie Wände und Decken schwächen das Signal stark ab. Dies erzeugt bei einer Verwendung von genügend vielen AP ein einmaliges Signalstärkemuster für jede Position in einem Gebäude [Willemsen u. a., 2014b].

Zur Erzeugung der Referenzdatenbank müssen die zuvor festgelegten, koordinatenmäßig bekannten Wegpunkte abgelaufen werden (Abb. 4.7 rechts). An jedem Wegpunkt wird das Funknetzwerk abgehört und so die empfangenen AP mit der jeweiligen Signalstärke registriert. Während einer Positionsbestimmung werden die aktuell aufgezeichneten Daten (das Ist-Muster) mit der zuvor generierten Referenzdatenbank (Referenzmuster) nach deterministischen oder probabilistischen Ansätzen ausgewertet (Abb. 4.7 links). Interpolationen ermöglichen zudem eine Positionsbestimmung zwischen den Referenzkoordinaten. Die Qualität der Positionsschätzung ist von der Verteilung und Anzahl der AP im Gebäude und den jeweiligen Signaldämpfungseigenschaften der Infrastruktur abhängig. Aus diesem

4.2 Infrastrukturabhängige Positionsschätzung

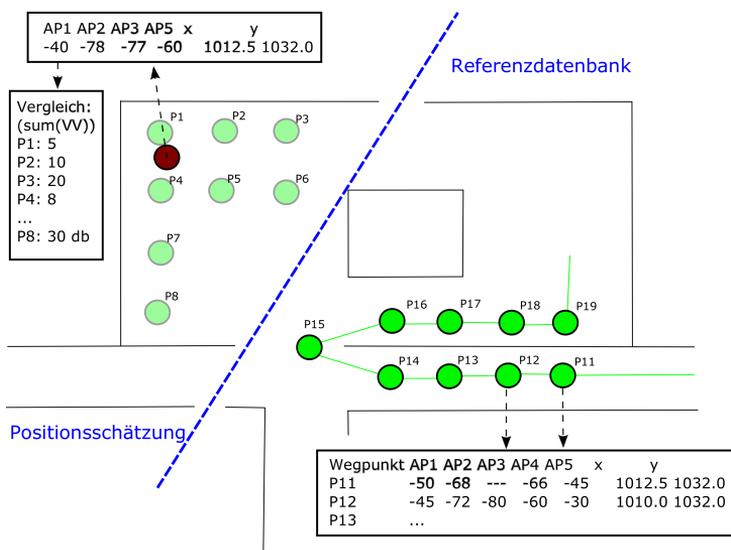


Abb. 4.7: Funktionsprinzip des Fingerprintings (rechts: Generierung einer Referenzdatenbank durch Beobachtungen der verfügbaren Signale an bekannten Positionen; links: Positionsschätzung auf Basis eines Vergleichs mit den zuvor gesammelten Referenzdaten) [Willemsen u. a., 2014b].

Grund ist der Implementierungsaufwand zwar abhängig von der Infrastruktur, aber doch grundsätzlich als hoch anzunehmen.

Trilateration

Die Trilateration beschreibt die Positionsschätzung eines Objekts durch Bestimmung der Distanzen zwischen den Objekten und Referenzpositionen [Blankenbach, 2007]. Für eine Schätzung im ebenen Koordinatensystem sind drei Distanzen erforderlich, für ein räumliches mindestens vier. Die Verarbeitung zusätzlicher Distanzen kann sinnvoll sein, um grobe Messunsicherheiten auszuschließen und die Zuverlässigkeit der Berechnung zu verbessern. Abb. 4.8 stellt das Prinzip mit drei Distanzen in einem ebenen Koordinatensystem dar. Ein robuster Schätzer wird in [Blankenbach, 2009] beschrieben. Der funktionale Zusammenhang für die Lage ist der Pythagoras mit (50):

$$S_{ij} = \sqrt{(x_{AP_i} - x_j)^2 + (y_{AP_i} - y_j)^2} \quad (50)$$

Der Pythagoras stellt den Zusammenhang zwischen den gemessenen Strecken S_{ij} , den bekannten AP-Koordinaten x_{AP_i} und y_{AP_i} sowie der gesuchten Position x_j , y_j her. Die Trilateration bietet sich für alle Verfahren an, bei denen es möglich ist, direkt oder indirekt Distanzen zwischen AP und navigierender Person zu bestimmen.

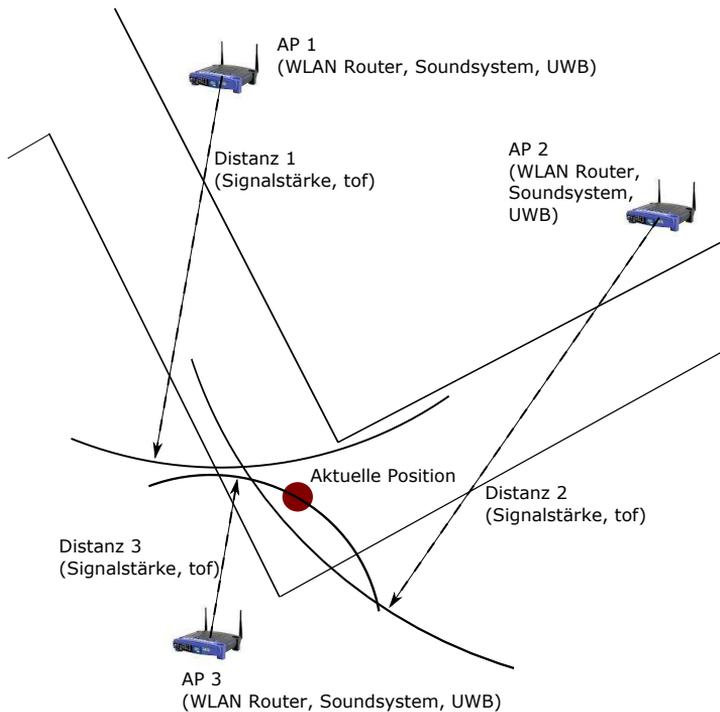


Abb. 4.8: Positionsschätzung nach dem Prinzip der Trilateration am Beispiel von Innenraumanwendungen. Ableitung der Distanzen zwischen Sender und Empfänger und anschließende Schätzung mit dem Pythagoras als funktionalem Zusammenhang [Willemsen u. a., 2014b].

Triangulation

Eine auf Triangulation basierte Positionsschätzung verwendet ausschließlich Richtungsmessungen zu bekannten Referenzpunkten, die im Bezugskoordinatensystem vorliegen. Die Positionsberechnung erfolgt nach einem Rückwärts- oder Vorwärtschnitt [Blankenbach, 2007]. Gegenüber der Trilateration ist die Triangulation auf Basis von zwei Beobachtungen eindeutig (Abb. 4.9). Je nach verwendeter Technologie zur Richtungsmessung ist eine Sichtverbindung notwendig (iGPS) oder vor-

teilhaft, um Messunsicherheiten zu minimieren. Den funktionalen Zusammenhang zwischen den AP-Koordinaten x_{AP_i} und y_{AP_i} sowie den Objektkoordinaten x_j und y_j für die Positionsberechnung liefert im lokalen System die Azimut-Berechnung nach Formel (51).

$$t_{ij} = \arctan \left(\frac{y_{AP_i} - y_j}{x_{AP_i} - x_j} \right) \quad (51)$$

Die netzwerkgestützte Triangulation basiert auf Antennen, die Richtungsinformationen liefern. Dies wird durch ein Antennen-Array möglich, das sich auf einer Platine im AP befindet. Das Prinzip der Triangulation in einer solchen Technologie wird auch als Angle of Arrival (AOA) bezeichnet. Die Qualität der Positionsschätzung ist von Mehrwegeeffekten im Gebäude und von der Aufnahmegeometrie abhängig [Blankenbach, 2007]. Zekavat u. Buehrer [2012] favorisieren den Einsatz der Triangulation mit Antennenarrays nur bei direkter Sichtverbindung zwischen Empfänger und Sender (Line of Sight; LOS), da Mehrwegeeffekte ansonsten die Richtung zu stark beeinflussen. Solche Mehrwegeeffekte ergeben sich beispielsweise aus Signalreflexionen, die unter einem anderen Winkel auf das Antennenarray treffen und so die reale Strahlenrichtung verändern.

Polares Anhängen

Das Polare Anhängen beschreibt in der Geodäsie die Berechnung einer neuen Position von einer bekannten Position auf Basis von Strecken- und Richtungsmessungen [Blankenbach, 2007] (Abb. 4.10). Der Vorteil gegenüber den bisher beschriebenen Verfahren liegt in der Möglichkeit, eine Position mit nur einem AP bestimmen zu können. Die Einflüsse auf die Strecken- und Richtungsbestimmung entsprechen denen der zuvor beschriebenen Triangulations- und Trilaterationsverfahren. Eine Kombination dieser Verfahren und die Berechnung nach dem polaren Anhängen erlaubt eine Steigerung der Beobachtungsanzahl ohne die Anzahl von AP im Gebäude zu erhöhen.

Technologien zur Positionsschätzung

WLAN

Das WLAN zur Netzwerkkommunikation ist mittlerweile in nahezu allen öffentlichen Gebäuden zu finden. Dies macht es als Werkzeug zur Innenraumnavigation

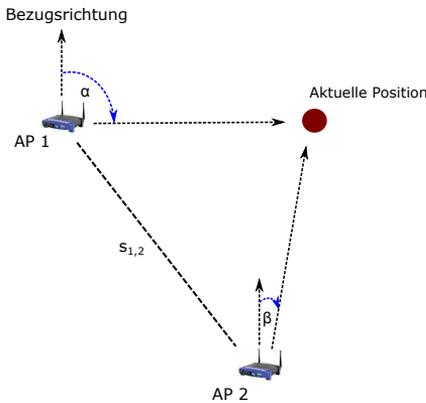


Abb. 4.9: Prinzip der Triangulation für die Lagebestimmung (abgeleitet nach [Blankenbach, 2007]).

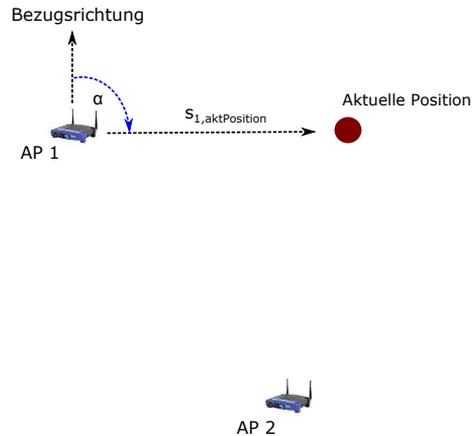


Abb. 4.10: Prinzip des Polaren Anhängens für die Lagebestimmung (abgeleitet nach [Blankenbach, 2007]).

besonders interessant. Die Funksignale von WLAN können auf unterschiedliche Weise zur Positionsschätzung genutzt werden. Am einfachsten ist das Prinzip der Annäherung (Cell of Origin). Wird ein AP von der Navigationseinheit empfangen, so befindet sich die navigierende Person innerhalb eines zuvor für den AP definierten Bereichs, beispielsweise in einem Raum. Die Position wird dann der Koordinate des AP gleichgesetzt. Eine Erweiterung wäre die Positionsschätzung nach dem WCL Verfahren, in dem mehrere AP zur Positionsschätzung beitragen. Ein Einsatz auf Basis der Trilateration ist zusätzlich möglich. Dazu müssen die Signalstärken [dB] in Distanzen [m] umgerechnet werden. Dies erfolgt auf Basis des Ausbreitungsmodells von Funksignalen. Die Signaldämpfung entspricht einer logarithmischen Funktion, daher nimmt mit steigender Entfernung zum AP die Unsicherheit der Distanz aus Signalstärken zu. Diese Methode ist vergleichsweise ungenau, da Wände und Decken in Gebäuden auf die Signale zusätzlich dämpfend wirken [Johnke, 2012].

Aufgrund der starken Signaldämpfung und der Möglichkeit der Umsetzung mit Standardhardware wird das Fingerprinting auf Basis von WLAN Signalstärken für die Positionsschätzung gerne genutzt. Das allerdings setzt eine große Dichte an

vorhandenen AP voraus, damit jede Position mit hoher Wahrscheinlichkeit aus den Referenzdaten ableitbar ist. Üblicherweise muss ein vorhandenes Netzwerk mit weiteren AP ausgestattet werden, damit das Fingerprinting ausreichend gut Positionen schätzen lässt. Nutzbar ist jeder AP, der Signale senden kann. Ein Internetzugang ist nicht notwendig. Ein weiterer Vorteil der Methode besteht darin, dass die Positionen der vorhandenen AP nicht ermittelt werden müssen und das WLAN weiterhin für das Internet eingesetzt werden kann.

Einsatzgebiete sind auch innerhalb von Stadtgebieten zu finden. Die Genauigkeiten der Positionsbestimmung betragen je nach Beschaffenheit des Areals zwischen 3-100 m. Zur schnellen Initialisierung setzt beispielsweise Google WLAN-AP ein, um die Position aus einer Kombination von GPS und WLAN schnell zu finden und gleichzeitig die Smartphone-Batterie zu schonen. Die für Google-Street-View erfassten Netzwerke dienen der Positionsbestimmung ohne GPS-Information [heise.de, 2016]. Darüber hinaus bietet Google mit seinen Indoor Maps eine eigene Lösung zur WLAN basierten Indoor-Navigation an [Johnke, 2012]. Sie ist aktuell allerdings nur bedingt nutzbar, denn die erreichbaren Genauigkeiten, aufgrund des fehlenden Eingriffs in die Infrastruktur, erheblich variieren.

Das Fraunhofer Institut für integrierte Schaltung (IIS) hat mit dem System *awiloc* einen auf Fingerprinting basierten Indoor-Positionierungs-Dienst umgesetzt, bei dem die Positionsschätzung ausschließlich per WLAN Fingerprinting erfolgt. Bisher konnten so 60 Projekte für unterschiedlichste Anwendungen umgesetzt werden [IIS, 2015]. Für ausreichend gute Positionsgenauigkeiten nutzt *awiloc* unter anderem Richtantennen, um ein für jede Position unterscheidbares Muster zu erzeugen. Entsprechend hoch ist der Implementierungsaufwand. Ein hybrider Ansatz mit INS könnte den Aufwand an Infrastrukturmaßnahmen minimieren und so eine flächendeckende Positionsbestimmung mittels Fingerprinting unnötig machen. Das bedeutet, dass das Fingerprinting sich zur Stützung einer IMU-basierten Positionsschätzung anbieten kann [Willemsen u. a., 2014b].

Bluetooth

Bluetooth sendet wie WLAN im Frequenzband von 2,4 GHz und kann zur Positionsschätzung nach den gleichen Prinzipien umgesetzt werden. Üblicherweise haben Gebäude keine Bluetooth-Infrastruktur, sie muss, im Gegensatz zum WLAN,

eigens geschaffen werden. Die erreichbare Genauigkeit ist mit einer WLAN basierten Positionsschätzung vergleichbar [Johnke, 2012]. Im Vergleich zu den Vorgängerstandards eignet sich der von Nokia entwickelte Standard Bluetooth 4.0 (BLE) am besten zur Positionsbestimmung. Durch Bluetooth 4.0 erfreuen sich beispielsweise BLE-Beacons großer Beliebtheit und werden als kabelloses Low Energy System beworben, das vergleichsweise einfach zu installieren ist [indoornavigation.de, 2014]. Voraussetzung ist, dass das Navigationsgerät, hier das Smartphone, eine BLE-Schnittstelle besitzt.

Ultraschall

Ultraschallverfahren nutzen die Ausbreitungsgeschwindigkeit von Schall 343 m/s im Frequenzbereich größer 16kHz . Dabei werden Ultraschall-Signale von der Navigationshardware oder vom AP ausgesendet und vom jeweils anderen Gerät empfangen. *ActiveBat* ist ein System, das in der Infrastruktur verbaut ist und Schall von der Navigationshardware empfängt, während *Cricket* den umgekehrten Weg geht und kontinuierlich Signale im Gebäude aussendet [Blankenbach, 2007]. Die Reichweite von Ultraschall ist begrenzt. Sie liegt nur bei wenigen Metern ($<5\text{ m}$). Darüber hinaus stören Sichtbehinderungen und Reflexionen die Distanzbestimmung. Aufgrund der geringen Reichweite müssen für Anwendungen, die nach dem Prinzip der Trilateration arbeiten, hohe Dichten an AP realisiert werden. Eine Verringerung des Implementierungsaufwands kann auf Kosten der Positionsgenauigkeit mit COO erreicht werden. Eine Umsetzung für eine smartphonebasierte Navigation scheint schwierig und sie benötigt zusätzliche Hardware, wenn das Mikrofon des Smartphones den Frequenzbereich nicht auflösen kann. Dem hohen Implementierungsaufwand steht die hohe erreichbare Positionsgenauigkeit im Bereich weniger Zentimeter gegenüber.

Künstliches Magnetfeld

Eine weitere Möglichkeit der Positionsschätzung ist beispielsweise in Hellmers u. a. [2013] beschrieben. Bisher wurde das Magnetfeld weitgehend mit Hilfe von Magnetometern zur Bestimmung der Nordrichtung verwendet. Dies ist aufgrund äußerer ferromagnetischer Einflüsse im Gebäude oft nicht sinnvoll. In Hellmers u. a. [2013] wird ein Ansatz präsentiert, der es erlaubt ein künstliches Magnetfeld für die Positionsbestimmung zu nutzen. Ein dort beschriebenes System wird mit *MILPS*

(*Magnetic Indoor Local Positioning System*) bezeichnet. Dazu werden mit Kupfer unwickelte Spulen als Elektromagnete verwendet, die jeweils ein künstliches Magnetfeld erzeugen. MILPS arbeitet nach dem Prinzip des Zeitmultiplexverfahrens (TDMA = Time Division Multiplex Access), wodurch sich unterschiedliche AP am empfangenen Gerät voneinander unterscheiden lassen. Die Berechnung der Position erfolgt nach der Trilateration aus mindestens drei empfangenen Spulen. Eine weitere Anwendungsmöglichkeit ist der Einsatz zur Stützung einer INS-basierten Navigation. Hierfür würde bereits eine Spule ausreichen [Hellmers u. a., 2013]. Das von [Hellmers u. a., 2013] vorgestellte System arbeitet mit einem üblichen MEMS-Magnetfeldsensor und kann daher in Kombination mit dem Smartphone umgesetzt werden.

Radio Frequency Identification

Das Prinzip der Radio Frequency Identification (RFID) dient zur Auszeichnung und Kennzeichnung von Waren. RFID-Tags erlauben eine Speicherung von Warenbezogenen Informationen, die sich von einer entsprechenden Empfangseinheit kabellos auslesen lassen. Es lassen sich aktive und passive Systeme unterscheiden. Während aktive Systeme eine Stromversorgung, beispielsweise eine Knopfzellenbatterie, benötigen, können passive mit Energie aus den empfangenen Funkwellen betrieben werden. Beide Systeme unterscheiden sich nicht nur in der Bauart, auch in der Leistung und Reichweite sind sie verschieden. Während batteriebetriebene RFID-Chips in hohen Frequenzen (2,4-2,5 GHz, 5,8 GHz) Reichweiten von bis zu 1 km erreichen, sind passive Systeme vermehrt für einen Einsatz von wenigen Zentimetern ausgelegt [Blankenbach, 2007]. Mit der COO-Methode können Genauigkeiten zwischen 0,5 - 20 m erreicht werden [Blankenbach, 2007]. Ein Einsatz als Fingerprinting, als WCL oder eine Distanzbestimmung für die Trilateration sind ebenfalls möglich.

Infrarot

Methoden, die Infrarotsignale verwenden, arbeiten nach dem Prinzip der Annäherung. Die Navigationshardware kann dabei als Sender oder Empfänger fungieren. Fungiert sie als Sender, muss eine serverbasierte Positionsschätzung stattfinden, da die Navigationshardware Informationen zur Erreichbarkeit im Netzwerk benötigt. Dazu können sogenannte *Active Badge* als Empfänger dienen. *IR-Baken* dage-

gen fungieren als Sender, wenn die Navigationshardware als Empfänger eingesetzt werden soll [Blankenbach, 2007]. Eine auf Infrarot basierte Positionsschätzung dient der Raumidentifikation [Blankenbach, 2007] und benötigt für eine Kommunikation die Sichtverbindung zwischen Sender und Empfänger.

Long Term Evolution

Long Term Evolution (LTE) ist ein aktueller Mobilfunkstandard, der auch mit lokalen AP zur Innenraumpositionsschätzung eingesetzt werden kann. In Alyafawi u. a. [2015] wird eine Umsetzung einer LTE-basierten Positionsschätzung vorgestellt. Dazu wurde ein Stockwerk mit vier LTE-fähigen AP ausgestattet, die sich für einen Einsatz im Funknetz als Femtozellen eignen. Zur Bestimmung der Position können Signalstärken verwendet werden. In Alyafawi u. a. [2015] werden Genauigkeiten von $2,7m$ und $3,3m$ erreicht. Der Vorteil von LTE liegt in der einfachen Kombination von Außenbereich und Innenbereich.

Ultra Wide Band

Als Ultra Wide Band (UWB) wird eine breitbandige Funktechnologie bezeichnet, deren Bandbreite bei >500 MHz liegt. UWB eignet sich besonders für die Übertragung großer Datenmengen im Nahbereich bis 10 m. Aufgrund strenger Leistungsbegrenzungen beträgt der maximale Einsatzbereich ca. 40-50 m. Die hohe Übertragungsrate im Nahbereich macht UWB zur Konkurrenz von kabelgebundenen Datentransfersystemen wie USB und FireWire. Ein Vorteil ist auch der gegenüber WLAN-Systemen geringe Energieverbrauch [Blankenbach, 2007]. Norrdine u. a. [2013] zeigen auf, dass mit der Verwendung des Indoor Local Positioning System (ILPS), das auf UWB basiert, Positionsupdates mit einer Messrate von drei Sekunden zu erzielen sind. Das genutzte Frequenzband liegt zwischen 3,2 und 6,3 GHz. UWB zur Positionsschätzung kann auf gleiche Weise angewendet werden wie WLAN oder Bluetooth. Allerdings sind wegen der hohen Bandbreite weniger Signalstörungen durch Wände zu erwarten. So können Genauigkeiten in der Positionsschätzung im Bereich von wenigen Zentimetern erreicht werden. Die Umsetzung ist allerdings zurzeit nur zu Versuchszwecken möglich, da es in Europa keine Standards zur UWB-Nutzung gibt [Blankenbach, 2007].

GPS-Repeater

GPS-Repeater dienen dazu, das empfangene GPS-Signal an GPS abgeschattete Bereiche weiterzuleiten. Im Außenbereich wird dazu eine GPS-Antenne befestigt, die den Kontakt zu den Satelliten herstellt. Im Innenraum werden GPS-Repeater installiert, die das Signal von dem außen liegenden Sensor empfangen und in den Innenraum weiterleiten [Johnke, 2012]. Der Vorteil liegt in der Verwendung von nur einem Koordinatensystem für die Verknüpfung von Outdoor- und Indoor-Navigation. Die praktische Umsetzung ist allerdings kostspielig und aufwendig. Auch ist zu beachten, dass die Repeater nur die Position der Empfangsantenne weiterleiten. Eigentliches Anwendungsgebiet ist die Signalversorgung in GPS-abgeschatteten Bereichen, beispielsweise für eine Entwicklungsabteilung, die mit GPS arbeitet. Die Kosten für einen Repeater bewegen sich im Bereich von 50 Euro bis 850 Euro [GPS-Repeater, 2016].

4.2.2 WLAN Fingerprinting am Beispiel des HCU-Gebäudes

Als praktisches Beispiel wird aus dem schon vorhandenen WLAN Netzwerk im HCU-Gebäude eine Positionsschätzung auf Basis des WLAN-Fingerprintings abgeleitet. Dabei findet keine Verbesserung der Positionsgenauigkeit durch eine Netzverdichtung mit zusätzlichen AP statt. Das Netzwerk wird von der hauseigenen IT-Abteilung gepflegt, so dass ein zusätzlicher Aufwand dafür entfällt. Ziel des Tests ist die Ermittlung der möglichen Positionsgenauigkeit sowie eine Einschätzung des Arbeitsaufwands und der Einsatzmöglichkeit einer hybriden Anwendung am Beispiel des HCU-Gebäudes. Die Zuordnung der Referenzpositionen wird dabei nach deterministischem und probabilistischem Ansatz vergleichend berechnet.

Positionsberechnung nach deterministischem und nach probabilistischem Ansatz

Die deterministischen Ansätze verwenden die Differenzen zwischen den aktuell gemessenen Signalstärken und den Daten aus der Referenzdatenbank. Ein Beispiel ist der Nearest Neighbour Algorithmus, der die euklidische Distanz nutzt.

Die euklidische Distanz d_E berechnet sich aus der Wurzel der Summe der quadrierten Signalstärkendifferenzen aller möglicher zu empfangener AP (52). Daher

ist die Differenz jeder empfangenen Signalstärke $P_{act_{AP}}$ zu jeder zugehörigen Signalstärke in jeder Referenzposition $P_{ref_{AP}}$ zu ermitteln. Empfangene AP, die nicht bei einer Position in der Referenzdatenbank verzeichnet sind, werden in der Berechnung durch eine vergleichbar hoch gewählte Differenz eingebunden.

$$d_E(ref_{AP}, act_{AP}) = \sqrt{\sum_{i=0}^n (P_{ref_{AP_i}} - P_{act_{AP_i}})^2} \quad (52)$$

In Abb. 4.11 ist die euklidische Distanz für jede Position in der HCU-Referenzdatenbank schwarz dargestellt. Rot markiert sind die größtmäßig sortierten Distanzen. Die minimale euklidische Distanz entspricht dabei der wahrscheinlichsten Position.

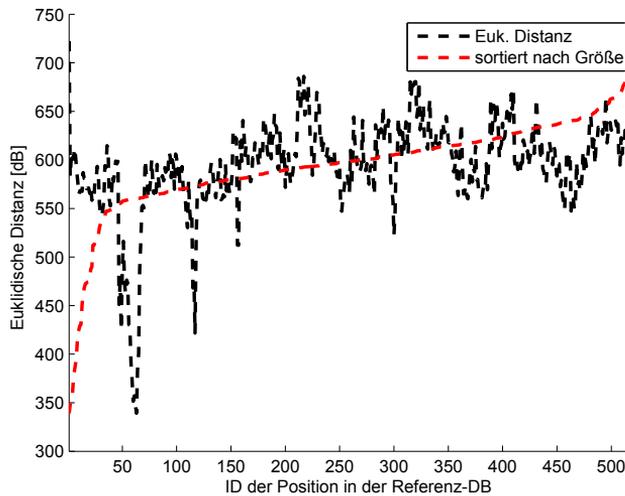


Abb. 4.11: Euklidische Distanz einer Positionsschätzung mittels Fingerprinting durch einen Vergleich zu jeder Position in der vorliegenden Datenbank. Die minimale euklidische Distanz ist die wahrscheinlichste aktuelle Position.

Ein probabilistischer Ansatz wird in Leitz u. a. [2015] favorisiert. Nach dem Occupancy Grid Ansatz werden in Abhängigkeit der berechneten Differenzen zwischen aktueller empfangener Signalstärke und den Referenzdaten den Positionen Wahrscheinlichkeitsfaktoren zugeordnet.

Zu Beginn hat jede Position in der Referenzdatenbank den gleichen Wahrscheinlichkeitswert. Jede Differenz zu einem AP bringt einen neuen Wahrscheinlichkeitswert ein. Zum Abschluss werden alle Wahrscheinlichkeiten miteinander multipliziert und über den gesamten Datensatz normiert. In Abb. 4.12 werden die Wahrscheinlichkeiten der möglichen Referenzpositionen für eine Positionsberechnung aufgezeigt. Wie schon bei dem Ansatz der euklidischen Distanz lässt sich hier eine Position eindeutig feststellen. Es zeigen sich zudem erhöhte Wahrscheinlichkeiten der umliegenden Positionen, da die ID-Bezeichnungen auch den Zusammenhang in der Örtlichkeit darstellen.

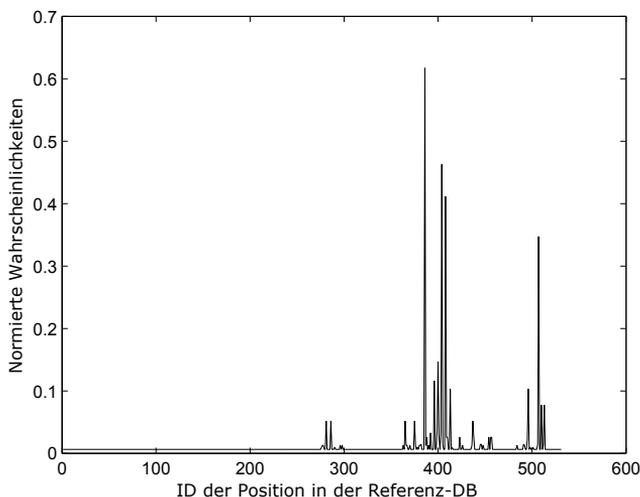


Abb. 4.12: Berechnete Wahrscheinlichkeiten einer Fingerprinting Positionsschätzung auf Basis eines probabilistischen Ansatzes. Der maximale Wert repräsentiert die wahrscheinlichste Position.

Umsetzung für das HCU-Gebäude

Das Gebäude der HCU besteht aus sechs Stockwerken mit Seminarräumen, PC-Pools und Büros. Die Räumlichkeiten und die Nutzungsweisen sind höchst unterschiedlich. Zurzeit arbeiten und studieren insgesamt an die 2.000 Personen im Gebäude.

Um die Referenzdatenbank für das gesamte Gebäude zu erstellen, wurden die öffentlichen Bereiche mit einem ungefähren Raster von 2,5 m abgelaufen. Für die

Stockwerke vom Erdgeschoss bis zum 5. Obergeschoss wurden so 520 Referenzpositionen registriert. Zudem wurden ca. 619 Mac-Adressen von AP erfasst. Dies entspricht nur indirekt der real installierten Hardware im Gebäude, da jeder AP bis zu vier unterschiedliche Netzwerke aussendet. An jeder Referenzposition wurden Daten für die Dauer von 10 Sekunden aufgezeichnet. Das Mittel repräsentiert die Signalstärke des jeweiligen Routers an der jeweiligen Position in der Referenzdatenbank. Um die einseitige Signalabschwächung durch den Nutzer zu reduzieren wurde das Smartphone an einem Stab auf Kopfhöhe des Nutzers fixiert.

Die Referenzdatenbank umfasst 520 Zeilen und 623 Spalten und benötigt als reine Textdatei einen Speicher von 1,2 MB. Jede Zeile enthält eine ID, die jeden Datensatz eindeutig identifiziert, eine Position aus den xyz-Koordinaten und 619 Spalten mit den Signalstärken für jeden AP. Wird ein AP nicht empfangen, enthält das Feld einen Nullwert für die Signalstärke.

Zur Bestimmung der Position werden der Nearest Neighbour Algorithmus und der Occupancy Grid Ansatz genutzt. Dann werden beide Ansätze auf Basis zweier statischer Messungen und zweier gelaufener Trajektorien verglichen. Die dazu verwendeten Rohdaten sind zum Zwecke einer prinzipiellen Eignung des Fingerprintings im HCU-Gebäude ausgewählt worden und entsprechen nicht unbedingt repräsentativen allgemeinen Testdaten, da schon die Auswahl der verwendeten Hardware dort eine Beschränkung bedeutet. Die Umsetzung durch die euklidische Distanz erfolgt nach Formel (52). Dabei werden niedrige Signalstärken ($< -90\text{dB}$) als nicht empfangene AP gewertet, da – je nach Hardware und Position des Nutzers – ihr Empfang gestört sein kann.

Die Wahrscheinlichkeiten für das Occupancy Grid werden in Abhängigkeit der Höhe der Differenzen verteilt. Wird eine Differenz der Signalstärken von 10 dB unterschritten, wird ein Faktor von 0,5 zugeordnet. Ist die Differenz größer als 10 dB, werden 0,4 zugrunde gelegt. Hat die empfangene Signalstärke einen Wert größer als 90 dB oder ist der AP bei den Referenzdaten nicht empfangen worden, dann wird ein Faktor von 0,3 festgelegt. Bei der Initialisierung erhalten alle Positionen das Gewicht $1/n$. Dieses Gewicht wird nun mit den neuen zugeordneten Wahrscheinlichkeiten multipliziert und normiert. Die Faktoren sind willkürlich wählbar und es könnten beispielsweise weitere Zwischenstufen implementiert wer-

4.2 Infrastrukturabhängige Positionsschätzung

den. Diese hier gewählten Faktoren liefern die bestmöglichen Ergebnisse für die HCU-Daten.

Tab. 4.1: Vergleich der Positionsschätzung im Fingerprinting nach dem deterministischem und dem probabilistischen Ansatz gegen Referenzpositionen. Der Wert 0.0 repräsentiert eine richtige Zuordnung. Die Abweichungen sind der räumliche Abstand zwischen Ist- und Soll-Koordinaten.

Kontrollpunkt	Euk. Distanz [m]	Occupancy Grid [m]	Kontrollpunkt	Euk. Distanz [m]	Occupancy Grid [m]
Datensatz OG4			Datensatz OG1		
1	0,0	0,0	1	28,9	42,5
2	3,2	2,9	2	26,5	0,0
3	10,2	0,0	3	3,1	2,7
4	0.0	0,0	4	0.0	0,0
5	3,0	3,0	5	7,9	3,8
6	0.0	0,0	6	24.9	5,0
7	0,0	0,0	7	28,0	8,1
8	2,4	3,3	8	10,8	10,8
9	2,7	3,3	9	3,1	3,1
10	0.0	0,0	10	6.7	0,0

Für den statischen Vergleich wurden Einzelpunkte in verschiedenen Stockwerken mit dem Textgerät Nexus 4 beobachtet. Die Daten wurden im Sekundentakt mit der Aufzeichnungs-App empfangen und gespeichert. Der Fingerprint wurde aus fünf Messungen gemittelt. Eine repräsentative Auswahl der Ergebnisse des statischen Vergleichs ist in Tab. 4.1 dargestellt. Für zehn Kontrollpunkte des ersten und vierten Obergeschosses sind Differenzen zwischen berechneter wahrscheinlichster Position sowie Referenzposition bestimmt worden. Aus den Daten lässt sich ablesen, dass die probabilistische Methode bessere Positionsberechnungen ermöglicht. Deutlich wird, dass sich Positionen im OG4, in dem sich viele kleine Büroräume befinden, präziser als im OG1 bestimmen lassen, das aus großen offenen Bereichen besteht. Dies zeigt die Abhängigkeit des Fingerprintings an der Infrastruktur sowie an der Art der Signalausbreitung für eine präzise Positions-

schätzung. Da die Signaldämpfung durch eine große Anzahl an Wänden im OG1 nicht gegeben ist, wäre ein Eingriff in die Infrastruktur erforderlich, um im OG1 gleichwertige Ergebnisse bezüglich der erreichbaren Genauigkeit zu gewährleisten. Eine andere Möglichkeit wäre die zusätzliche Installation von AP, beispielsweise als Richtantennen.

Die Positionsschätzung dient primär der Fußgängernavigation. Der Abgleich mit einer gelaufenen Trajektorie ist für die praktische Umsetzung daher von Bedeutung. Dazu wurden zwei Routen generiert. Die erste verläuft ausschließlich innerhalb des fünften Obergeschosses, die zweite erstreckt sich vom fünften in das vierte Obergeschoss.

Für beide Trajektorien wird eine typische Laufgeschwindigkeit von 6 km/h angenommen. Die letzten zwei empfangenen Signalstärke-Datensätze werden für jede Position gemittelt und zur Positionsberechnung verwendet. Die so im Sekundenrhythmus aufgezeichneten Daten entsprechen bei einer angenommenen Schrittgeschwindigkeit von $1,5\text{ m/s}$ einem Weg von ca. 3 m . Abb. 4.13 zeigt grafisch die Referenz (rot) und die Positionsschätzung nach beiden Algorithmen (grün: euk. Distanz, blau: Occupancy Grid). Die obere Grafik zeigt die erste Trajektorie für den Weg innerhalb des OG5. Der Testlauf beginnt rechts und verläuft über die Flure bis zum Fahrstuhl. In der unteren Skizze ist die zweite Trajektorie vom OG5 (schwarz) zum OG4 (grau) dargestellt. Der Testlauf beginnt rechts und verläuft über die Treppe in das OG4 bis zum linken Ende des Gebäudes.

Grafisch zeigen sich zum Teil größere Abweichungen der beiden Berechnungsmethoden. Für einen besseren Überblick sind in Tab. 4.2 die Differenzen einer Auswahl der berechneten Positionen zu den Referenzpositionen aufgeführt. Die Auswertung der Trajektorie im OG5 zeigt für das Occupancy Grid gegenüber der Euklidischen Distanz der Referenz besser angepasste Positionen. Dies wird bei der zweiten Trajektorie noch deutlicher. Die euklidische Distanz kann mit zusätzlichen Bedingungen weiter verbessert werden, beispielsweise durch Berücksichtigung naheliegender AP mit erhöhtem Gewicht. Zusätzlich kann die typische Schrittgeschwindigkeit für beide Ansätze herangezogen werden, um den Suchbereich nach möglichen Referenzpositionen einzugrenzen. Dies verringert zwar die Rechenzeit, aber erzeugt bei fehlerhaften vorherigen Schätzungen weitere Positi-

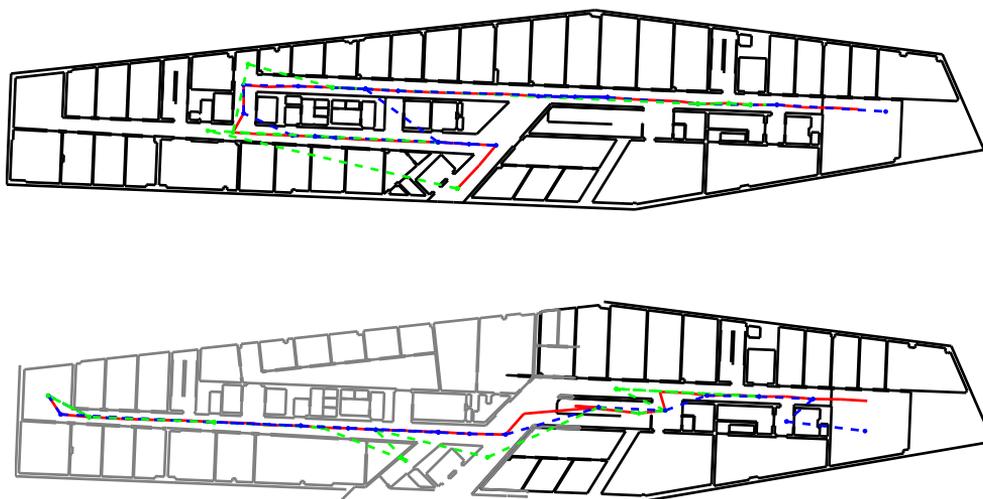


Abb. 4.13: Trajektorien im OG5 (schwarz) und OG4 (grau) berechnet nach der Euklidischen Distanz (grün) und dem Occupancy Grid (blau; Referenztrajektorie in rot).

onsfehler und minimiert den Vorteil gegenüber MEMS-INS-Verfahren, die zudem sich sukzessiv vergrößernder Positionsunsicherheiten unterliegen.

An dieser Stelle kann festgehalten werden kann, dass eine Positionsschätzung per WLAN-Fingerprinting, wie sie exemplarisch im HCU-Gebäude erprobt wurde, weitgehend möglich ist. Nicht nur die vorhandene Netzwerkstruktur, sondern auch die Architektur des Gebäudes wirken sich dabei positiv auf die Qualität der Schätzung aus. Dennoch ist sie in den großen und offenen Bereichen ungenau und benötigt zusätzliche Infrastrukturmaßnahmen, beispielsweise den Einsatz von Richtantennen. Im HCU-Gebäude erwies sich das Occupancy Grid als zuverlässigere Methode zur Positionsschätzung, die der euklidischen Distanz vorzuziehen ist.

Zur Erstellung der Referenzdatenbank für ein der HCU vergleichbares Gebäude benötigt eine Person bis zu zwei Tage Zeit. Mit angepassten Tools und Vorabinformationen zu den Referenzpositionen lässt sich der Aufwand verringern.

Der Nachteil der Nutzung vorhandener Netzwerkinfrastruktur ist die primäre Nutzung als Internetzugang. Das HCU-Netzwerk könnte reagieren, indem es die Signalstärke bei erhöhter Auslastung automatisiert anpasst, was wiederum eine Be-

Tab. 4.2: Vergleich der einzelnen Positionen für die Berechnung der Trajektorie nach dem probabilistischen und dem deterministischen Ansatz. Der Wert 0.0 repräsentiert eine richtige Zuordnung. Die Abweichungen sind der räumliche Abstand zwischen Ist- und Sollkoordinaten.

Kontrollpunkt	Euk. Distanz [m]	Occupancy Grid [m]	Kontrollpunkt	Euk. Distanz [m]	Occupancy Grid [m]
Datensatz OG5			Datensatz OG5 nach OG4		
1	8,9	3,0	1	27,5	3,3
2	6,0	5,9	2	24,6	4,4
3	3,0	3,0	3	21,6	3,8
4	8,6	0,0	4	18,6	3,0
5	0,0	0,0	5	15,7	0,0
6	2,4	2,4	6	2,4	2,4
7	5,7	0,0	7	9,9	0,0
8	7,8	2,0	8	7,9	3,2
9	10,9	4,8	9	2,1	2,1
10	10,8	6,0	10	2,9	7,3

einträchtigung der Positionsschätzung zur Folge hätte. Zudem macht jede Veränderung der Netzwerkinfrastruktur, der Gebäudeinfrastruktur sowie der Gebäudeeinrichtung eine erneute Erzeugung der Referenzdatenbank erforderlich.

4.3 Möglichkeiten der Stützung einer INS-basierten Positionsschätzung

Im Kapitelabschnitt *4.1.1 Pedestrian Dead Reckoning* wurde gezeigt, dass eine Positionsschätzung lediglich aus MEMS-Beschleunigungsmesser und MEMS-Gyroskop auf Dauer keine zufriedenstellenden Ergebnisse liefert. Die Positionsschätzung benötigt daher regelmäßige Korrekturen, um eine ausreichende Genauigkeit der Positionsschätzung für eine Anwendung zur Fußgängernavigation bereitzustellen. Dazu können unterschiedlichste Informationen genutzt werden, wie Richtungen und Distanzen, Koordinaten oder topologische Zustandsabfragen.

Der vorherige Kapitelabschnitt *4.2 Infrastrukturabhängige Positionsschätzung* gibt einen Überblick über mögliche infrastrukturbasierte Verfahren, die zusätzliche Informationen zur Positionsschätzung beitragen können. Im Folgenden findet eine Eingrenzung der favorisierten Komponenten statt.

4.3.1 Weitere Eingrenzung von Stützungsmöglichkeiten

Damit Implementierung und Wartung nicht zu aufwendig werden, soll sich die Positionsschätzung auf Basis von MEMS-INS-Sensoren so autonom wie möglich realisieren lassen, aber trotzdem die für die Fußgängernavigation erforderliche Genauigkeit einhalten. Abb. 4.14 listet im linken Kasten die Sensoren in aktuellen Smartphones auf. Der Kasten rechts liefert die Informationen, die sie bereitstellen, teilweise mit Bezug zur Infrastruktur, die im unteren Abschnitt des Überblicks aufgelistet ist.

Als primäre Sensoren dienen in dieser Arbeit die Inertialsensoren, der Magnetfeldsensor und der Luftdrucksensor. Der Magnetfeldsensor ist aufgrund des nicht berechenbaren Einflusses durch die Gebäudeinfrastruktur für die Orientierungsbestimmung nicht einsetzbar und vermag die Drift der Orientierungsberechnung aus Gyroskop-Daten nicht zu kompensieren.

Prinzipiell bietet sich die Kamera des Smartphones für einen autonomen Ansatz an, allerdings hängt die Qualität der Bildauswertung von der Umgebungsbeleuchtung und der Infrastruktur ab. Die Kamera lässt sich auf unterschiedliche Weise einsetzen, etwa zur Richtungsbestimmung mit einer Kantenerkennung bzw. Feature-Erkennung oder mit einer Target-Erkennung im Objektraum auf Basis eines Rückwärtseinschneidens. Die so ermittelten Daten können zur absoluten Positionsschätzung oder nur zur Korrektur einer Positionsschätzung dienen.

Aufgrund der Abhängigkeit von Infrastruktur und Lichtverhältnissen wird die Kamerafunktion auf eine Auswertung von QR-Codes reduziert. Dies kann zur Initialisierung der relativen Positionsschätzung (PDR) genutzt werden, um dem Anwender die Mühe zusätzlicher Eingaben zu ersparen. Für die benötigten Informationen wie Koordinate und Orientierung reicht ein QR-Code der Größe von $20 \times 20 \text{ mm}^2$ aus, der mit in Türschilder aufgenommen werden kann.

4.3 Möglichkeiten der Stützung einer INS-basierten Positionsschätzung

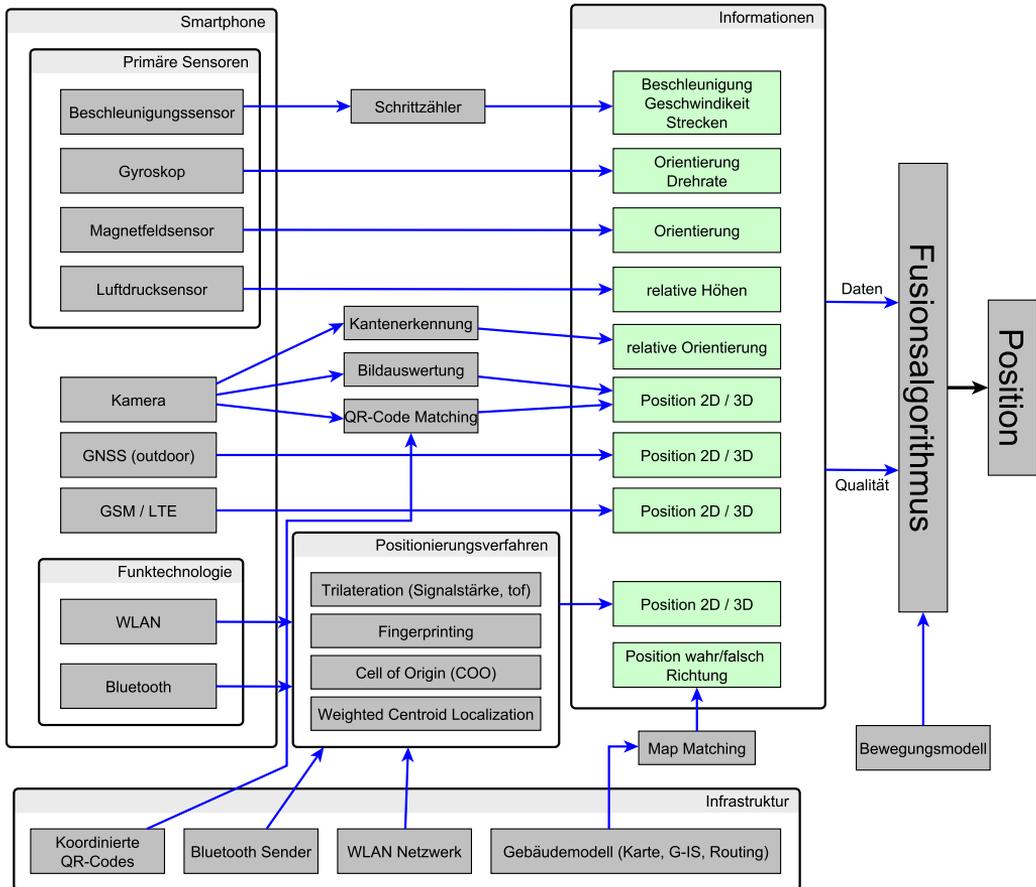


Abb. 4.14: Schematischer Aufbau der verfügbaren Komponenten für eine Indoor-Navigation mittels Smartphonesensoren.

GNSS- oder GSM/LTE-Daten liefern grobe Informationen, sofern eine Position im Außenbereich gesucht wird. Bei größeren Gebäudekomplexen kann mit ihrer Hilfe die Verbindung zwischen verschiedenen Gebäuden eines Komplexes oder der Übergang zwischen Außen- und Innenbereich überbrückt werden.

Als infrastrukturabhängige Positionierungsverfahren können Funktechnologien wie WLAN und Bluetooth zur Positionsschätzung eingesetzt werden. Allerdings kommen diese Verfahren in dieser Arbeit nur dann infrage, wenn bereits eine Netzwerkinfrastruktur vorhanden ist und der Aufwand an Implementierung und Wartung minimal ist.

Für eine Navigation sind Karten- und Routingdaten unabhängig von der Art der verwendeten Positionsschätzung notwendig, da sonst keine Routenberechnung zum Ziel möglich ist. Sie können zudem als Map Matching die Genauigkeit der Positionsschätzung steigern. Bei Karten- und Routingdaten handelt es sich um infrastrukturabhängige Informationen, die bei jeder Veränderung der Infrastruktur anzupassen sind. Da diese Informationen vor einer Navigation auf dem Smartphone verfügbar sein können und daher ohne Netzwerkanbindung eine Navigation möglich ist, lässt sich eine MEMS-INS-basierte Positionsschätzung, die sich nur auf Karten- und Routingdaten stützt, weiter als autonom charakterisieren.

In dieser Arbeit wird eine Fusion von den primären Sensoren mit Karten- und Routingdaten angestrebt. Im Folgenden werden dazu die Möglichkeiten der Verarbeitung der Daten für die Positionsschätzung beschrieben.

Primäre Sensoren

Die vorhandenen MEMS-Sensoren – Beschleunigungsmesser, Gyroskop, Magnetfeldsensor sowie Barometer – können sich auf unterschiedliche Weise gegenseitig stützen, um die Positionsschätzung zu verbessern. Der Beschleunigungsmesser wird genutzt, um das Smartphone-Koordinatensystem zu horizontieren. Dies ermöglicht es, dass nur die z-Achse des horizontierten Gyroskops in der Positionsschätzung verwendet werden muss. Mit dem Beschleunigungsmesser ist ferner die Initialisierung der Lagewinkel möglich, um anschließend eine rauscharme Horizontierung mit der Kombination aus den Daten von Beschleunigungsmesser und Gyroskop zu berechnen. Die Drift des Gyroskops steht dann dem erhöhten Messrauschen des Beschleunigungsmessers gegenüber. Wobei allerdings sich das Schrittmuster weiterhin auf die Neigungsberechnung auswirkt.

Das Barometer wird zur Stockwerksunterscheidung eingesetzt. Es kann unabhängig von den anderen Sensoren die relative Höhe bestimmen und erkennen, wenn das Stockwerk gewechselt wird. Fahrstuhl oder Treppe lassen dabei sich durch eine Kombination von Luftdruckänderung und Schrittzähler identifizieren.

Die höchste Positionsunsicherheit auf die Trajektorie im PDR-Verfahren wird durch die Orientierung um die Horizontalachse (r_z) erzeugt, die durch die Drift der integrierten Gyroskop-Daten hervorgerufen wird. Dieser Einfluss lässt sich per ZUPT-Befehle nur minimieren, sorgt aber weiterhin für Unsicherheiten bei der

Richtungsbestimmung. In einer magnetisch störungsfreien Umgebung kann der Magnetfeldsensor diese driftbehaftete Richtung aus dem Gyroskop mit Hilfe der Nordrichtung abmildern. Dazu muss zuvor die Richtungsangabe des Gyroskops mit der magnetischen Nordrichtung initialisiert werden. Bei der Testmessung im HCU-Gebäude waren aber die Resultate der Nordsuche (Abb. 3.24) nicht ausreichend für eine Richtungsstützung des MEMS-Magnetfeldsensors.

Karte

Grundlage jeder Navigation bilden die Karten (Abb. 4.15, schwarze Linien) zur Visualisierung. Zudem dienen sie als Grundlage für die Erstellung eines Routing-Graphs (blaue Linien). Das Datenformat des Kartenmaterials ist, wie bereits ausgeführt, ausschlaggebend für die Art und Weise der Verarbeitung in einer Applikation. In dieser Arbeit wird das Vektorformat bevorzugt. Es ist skalierbar und lässt einfache geometrische Berechnungen für einen Einsatz zur Stützung in der Positionsschätzung zu, ohne zu viel Rechenzeit und -leistung zu beanspruchen. Die Wände in einem Gebäudeplan dienen dabei der Abfrage topologischer Zusammenhänge und können zur Richtungsstützung genutzt werden.

Die Verwendung der Topologie ist in Abb. 4.16 dargestellt. Die aktuelle Position (blauer Punkt) sowie die Orientierung (r_z) bilden die Gerade, die mit den umliegenden Wänden geschnitten wird. Dazu werden Wände in Laufrichtung und 90° quer zur Laufrichtung geschnitten, damit der Übergang einer Wand nicht durch schleifende Schnitte hervorgerufen wird. Eine derartige Durchquerung der Wand wird dann möglich, wenn sich die navigierende Person zumeist parallel zu den Wänden fortbewegt. Die in der Abbildung mit roten Kreisflächen gekennzeichneten Schnittpunkte dienen daher als Bezugspunkte für die Bestimmung des Abstands zwischen Wand und Nutzerposition. Wird ein Grenzwert unterschritten, kann eine fortlaufende Bewegung unterbunden werden oder die verwendete Richtung der Wand ein erhöhtes Gewicht für eine Korrektur in der Positionsschätzung erhalten.

Die Berechnung der Schnittpunkte SP zu den umliegenden Wänden wird in Abb. 4.17 (rechts) gezeigt. Die Koordinaten des Schnittpunktes Y_{SP} und X_{SP} berechnen sich nach dem Geradenschnitt [Gruber u. Joeckel, 2004] durch Gleichungen (53), (54) und (55). Die Richtung der Bezugswand im Gebäudekoordinatensystem wird

4.3 Möglichkeiten der Stützung einer INS-basierten Positionsschätzung

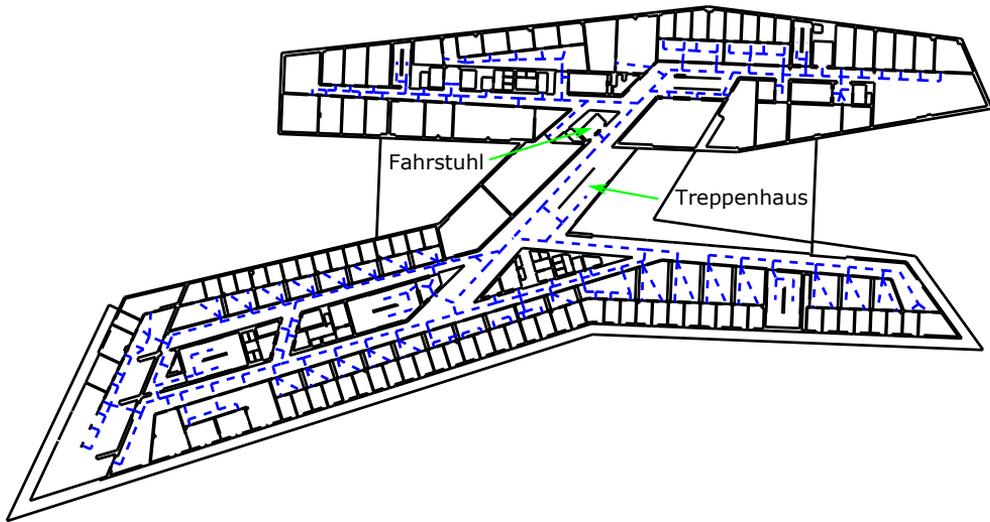


Abb. 4.15: Gebäudeplan des vierten Obergeschosses im HCU-Gebäude mit manuell erzeugtem Routing-Graph (blau).

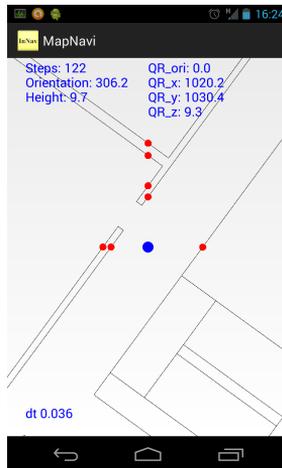


Abb. 4.16: Beispiel einer Kollisionsberechnung mittels Geradenschnitt aus aktueller Position und Richtung auf Basis einer Vektorkarte [Willemsen u. a., 2014a].

mit $t_{WP1,WP2}$ bezeichnet. Eine Rechenkontrolle erfolgt durch den Vergleich der Ergebnisse für Y_{SP} aus Formeln (54) und (55). Ein Schnittpunkt mit der Wand liegt vor, wenn der Schnittpunkt zwischen beiden Wandpunkten $WP1$ und $WP2$ liegt.

$$X_{SP} = X_{WP1} + \frac{(Y_{WP1} - Y_P) - (X_{WP1} - X_P) * \tan(r_z)}{\tan(r_z) - \tan(t_{WP1,WP2})} \quad (53)$$

$$Y_{SP} = Y_{WP1} + (X_{SP} - X_{WP1}) * \tan(t_{WP1,WP2}) \quad (54)$$

$$Y_S = Y_P + (X_{SP} - X_P) * \tan(r_z) \quad (55)$$

Nach diesem Ansatz können die Schnittpunkte für alle umliegenden Wände in Bewegungsrichtung oder quer dazu berechnet werden. Anschließend kann die Auswahl der Stützwand nach minimalem Abstand und minimaler Richtungsdifferenz erfolgen. Eine solche Korrektur kann besonders in Kurven zu ungewollten Verfälschungen der geschätzten Richtung führen. Eine zusätzliche Bedingung, die voraussetzt, dass der Fußpunkt der aktuellen Position auf der Wand liegt, kann einen negativen Einfluss der Richtungskorrektur in Kurven reduzieren.

Routing-Graph

Der Routing-Graph ist infrastrukturabhängig, denn das Netz wird aus den Gebäudeplänen abgeleitet und jede Veränderung am Gebäude ist daher auch beim Routing-Netz zu berücksichtigen. Der aus dem Kartenmaterial der HCU manuell abgeleitete Routing-Graph ist in Abb. 4.15 für das vierte Obergeschoss blau gestrichelt dargestellt.

Da das Routing-Netz die meist genutzten und möglichen Wege widerspiegelt, eignet sich das Kanten- und Knotenmodell gut zur Stützung der MEMS-INS-basierten Positionsschätzung. Die Auswahl von Routing-Kanten zur Stützung in der Positionsschätzung ist vergleichsweise einfach, da die Kante direkt über den orthogonalen Abstand o_P ausgewählt werden kann. Abb. 4.17 zeigt auf der linken Seite grafisch den Zusammenhang für die Berechnung von Ordinate o_P und Abszisse a_P der aktuellen Position P zur Routing-Kante auf. Die Länge der Or-

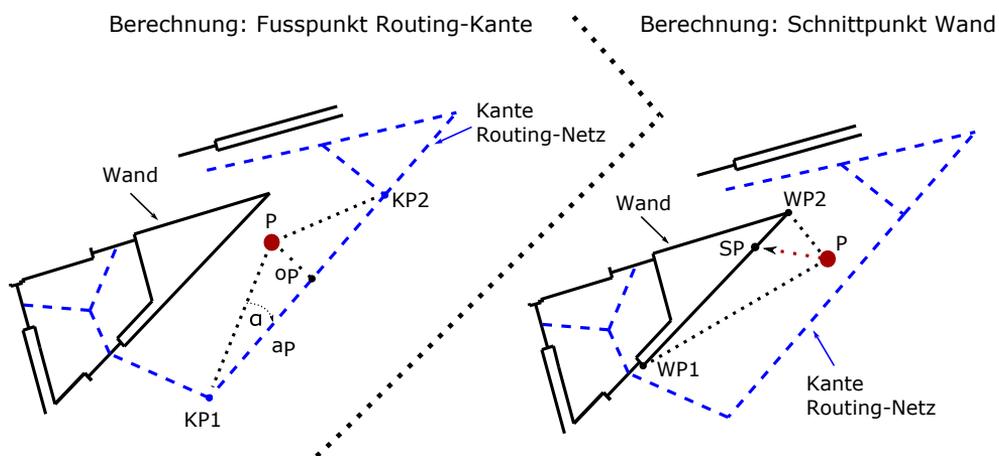


Abb. 4.17: Geometrischer Zusammenhang für die Berechnung von Stützinformati-
onen aus Karten- und Routingdaten (rot: aktuelle Position; blau: Routing-Kanten;
schwarz: Wände).

dinate dient als Auswahlkriterium für den Einsatz zur Positionsschätzung, sofern der Fußpunkt der aktuellen Position innerhalb des Bereichs der Routing-Kante liegt. Dies kann durch einen Vergleich der Länge der Abszisse mit der Kantenlänge abgefragt werden.

Die Gleichungen (56), (57), (58), (59) und (60) [Gruber u. Joeckel, 2004] zeigen den Ablauf zur Berechnung des Fußpunktes der aktuellen Position zu einer Routing-Kante. Der im aufgespannten Dreieck innen liegende Winkel α kann mittels Kosinussatz berechnet werden (56). Der orthogonale Abstand op und die Abszisse a_p sind anschließend bezogen zum Knotenpunkt $KP1$ mit Sinus- und Kosinusfunktion zu bestimmen (57, 58). Abschließend können die Fußpunktkoordinaten $Y_{P_{Fußpunkt}}$ und $X_{P_{Fußpunkt}}$ nach dem Polarpunktverfahren berechnet werden, wobei mit $t_{KP1,KP2}$ die Richtung der Routing-Kante im Gebäudekoordinatensystem bezeichnet wird.

$$\alpha = \arccos\left(\frac{S_{KP1,KP2}^2 + S_{P,KP1}^2 - S_{P,KP2}^2}{2 * S_{KP1,KP2} * S_{P,KP1}}\right) \quad (56)$$

$$op = \sin(\alpha) * S_{P,KP1} \quad (57)$$

$$a_P = \cos(\alpha) * S_{P, KP1} \quad (58)$$

$$Y_{P_{Fu\sspunkt}} = Y_{KP1} + \sin(t_{KP1, KP2}) * a_P \quad (59)$$

$$X_{P_{Fu\sspunkt}} = X_{KP1} + \cos(t_{KP1, KP2}) * a_P \quad (60)$$

Für eine Positionskorrektur kann der Fußpunkt direkt eingesetzt werden. Er korrigiert so die Positionsschätzung hin zur Routing-Kante, die meist mittig in einem Gang verläuft. Ist nur eine Stützung der Richtung gewünscht, ist die Richtung der Routing-Kante aus den zugehörigen Knoten zu berechnen. Die Auswahl der favorisierten Kante erfolgt dann zusätzlich über die minimale Richtungsdivergenz umliegender Routing-Kanten zur aktuell geschätzten Richtung.

Ein weiterer Einsatz des Routing-Netzes kann in einer direkten Positionsschätzung im Knoten- und Kantenmodell nach dem Prinzip des Map Matchings bestehen. Eine aufwendige Bestimmung weiterer Korrekturen, beispielsweise aus Karten, kann hier entfallen. Zudem könnte eine aufwendige Fusion zumindest vereinfacht werden, da nur die grobe Bewegungsrichtung bekannt sein muss. Ein erstmaliger Ansatz wird im folgendem Kapitel *5.3 Topologischer Ansatz auf einem Routing-Graphen* vorgestellt.

4.3.2 Favorisierte Komponenten

Die Möglichkeiten der Positionsschätzung in Innenbereichen sind vielfältig, aber in Anlehnung der in dieser Arbeit gesetzten Ansprüche an einer Positionsschätzung ist nicht jede Methode sinnvoll einsetzbar. Für eine möglichst autonome Positionsschätzung zur Minimierung von Aufwand in der Implementierung und Wartung sowie der einfachen Übertragung auf andere Gebäude werden in Abb. 4.18 die favorisierten Komponenten zusammengestellt.

Auf der linken Seite der Abbildung sind die verfügbaren Technologien und Informationen, unterteilt nach ihren Abhängigkeiten während einer Positionsschätzung,

4.3 Möglichkeiten der Stützung einer INS-basierten Positionsschätzung

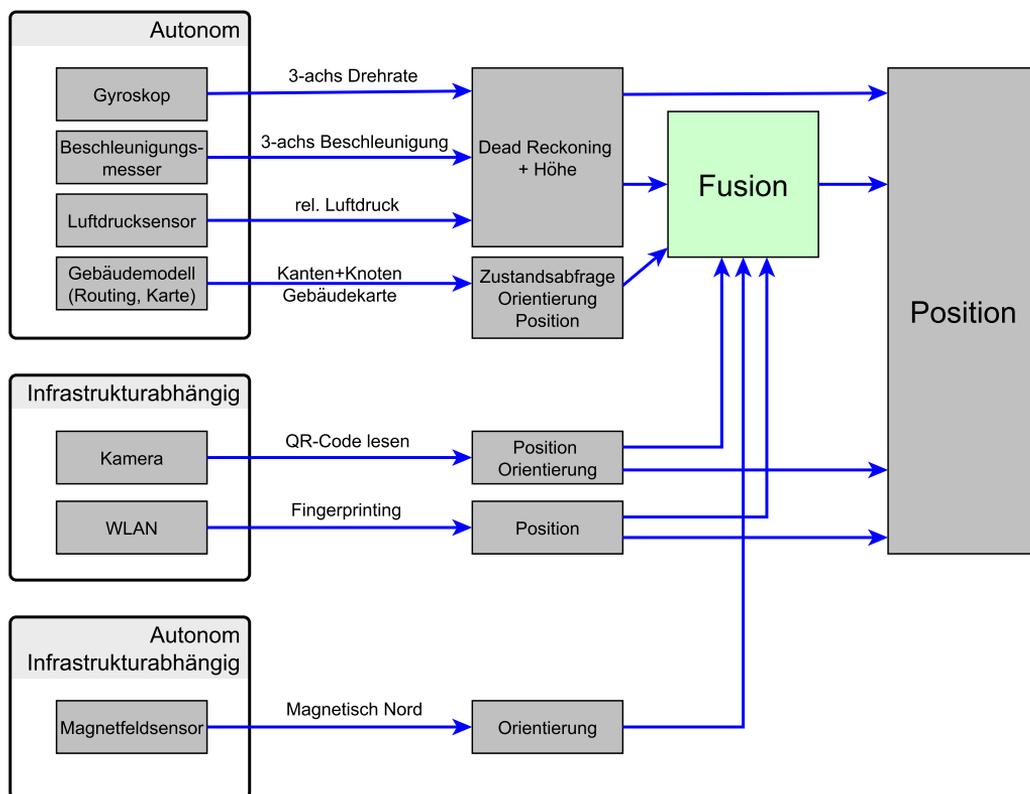


Abb. 4.18: Favorisierte Sensoren und deren Kombinationsmöglichkeit für die Positionsschätzung mittels Smartphone (Spalte 1: Sensoren, Spalte 2: Teilergebnis für die Fusion).

zung, aufgelistet. Die unter *Autonom* eingegliederten Komponenten lassen sich bei der Wegführung ohne äußere Interaktion verarbeiten. Zudem ist eine relative Positionsberechnung in Lage und Höhe durch den Beschleunigungsmesser, Gyroskop und Barometer möglich. Die Routing- und Kartendaten werden während der Navigation nicht verändert und können in der Fusion zusätzliche Stützinformationen liefern.

Die Komponenten des rein infrastrukturabhängigen Blocks benötigen eine Interaktion mit der Umgebung. Das Fingerprinting, basierend auf vorhandenem WLAN, benötigt Empfang von AP und die Kamera benötigt QR-Codes zur Initialisierung des PDR. Ist es nicht möglich, QR-Codes im Gebäude anzubringen, kann eine

Initialisierung über die Eingabe von Koordinaten oder Raumnummern erfolgen. Der Kamerasensor wäre dann unnötig, allerdings wird für einen QR-Code-Scan das Smartphone für einen kurzen Moment in einer bekannten Haltung gehalten, was die Unsicherheiten in der Initialisierung für Orientierung und Position maßgeblich verringert.

Ein WLAN Netzwerk wie im HCU-Gebäude kann zur Positionsschätzung zwar eingesetzt werden, da aber Infrastrukturmaßnahmen vermieden werden sollen und keine weiteren AP installiert werden, ist die infrastrukturbasierte Positionsschätzung nur bedingt einsetzbar. Das hier favorisierte Fingerprinting kann dennoch an einigen Positionen im Gebäude eine ungenaue MEMS-INS-basierte Positionsschätzung durch absolute Koordinaten verbessern. Der Vorteil ist die Unabhängigkeit beider Verfahren voneinander. Prinzipiell sind auch weitere infrastrukturbasierte Verfahren sinnvoll einsetzbar, vorausgesetzt sie können mit vergleichsweise wenig Aufwand zusätzliche Informationen in den Bereichen bereitstellen, in denen eine MEMS-INS-basierte Positionsschätzung, korrigiert aus Karten- und Routingdaten, versagt. Ein Beispiel wäre der Einsatz eines *MILPS*-AP, das auf künstlichen Magnetfeldern basiert und schon durch nur eine Distanzinformation die Positionsschätzung im Filter verbessern kann [Hellmers u. a., 2013].

Der Magnetfeldsensor wird in dieser Arbeit ansonsten gesondert betrachtet, da eine Orientierung zu magnetisch Nord zwar autonom möglich ist, aber ferro- und elektromagnetische Einflüsse im Gebäude unterschiedlich stark den Winkel zu magnetisch Nord verfälschen können.

Wie Abb. 4.18 aufzeigt, werden die Sensordaten bzw. die Informationen vorverarbeitet und können anschließend in der Fusion miteinander für eine robuste Positionsschätzung kombiniert werden. Die Möglichkeiten der Fusion werden im folgenden Kapitel 5 *Gestützte MEMS-basierte Positionsschätzung* am Beispiel von drei Möglichkeiten näher untersucht.

5 Gestützte MEMS-basierte Positionsschätzung

Auf Basis von Kalman Filter und Partikelfilter werden in diesem Kapitel zwei weitgehend autonom arbeitende Ansätze vorgestellt. Anschließend wird ein Verfahren diskutiert, das die Positionsschätzung ausschließlich per Routing-Graph erlaubt. Alle drei Ansätze unterscheiden sich in der Verarbeitung von Stützinformationen und haben unterschiedliche Stärken und Schwächen. Zum Abschluss wird ein kombinierter Ansatz vorgestellt, der sich aus dem Partikelfilter und dem kantenbasierten Ansatz des Routing-Graphen herleitet.

5.1 Kalman Filter

Das Pedestrian Dead Reckoning bildet die Basis einer MEMS-INS-basierten Positionsschätzung. Im Folgenden wird diskutiert, wie sich Stützinformationen im Kalman Filter (KF) einbinden lassen, um die Unsicherheiten der Positionsschätzung zu minimieren. Für die Grundlagen des KF sei auf das Kapitel 2.2.1 *Kalman Filter* verwiesen.

5.1.1 Aufbau des Filters

Ansatz mit Korrektur durch magnetisch Nord:

Ein erster eigener Ansatz einer KF-basierten Positionsschätzung wird in Willemssen u. a. [2013] beschrieben. Er zielt auf eine direkte Verarbeitung der Sensordaten im Filter auf Rohdatenebene ab. Als Systemzustand werden jeweils dreidimensionale Positionen, Geschwindigkeiten, Orientierungen und Drehgeschwindigkeiten in einem horizontalen Koordinatensystem definiert. Das Bewegungsmodell wird durch die Systemmatrix A beschrieben (61). Im Bewegungsmodell repräsentiert der obere Anteil die drei Orientierungen $r_{x_{eben}}$, $r_{y_{eben}}$ und $r_{z_{eben}}$ mit den jeweiligen verebneten Drehraten $\dot{r}_{x_{eben}}$, $\dot{r}_{y_{eben}}$ und $\dot{r}_{z_{eben}}$, wobei die Orientierungen aus den Drehraten aktualisiert werden. Der untere Teil von A , berücksichtigt die räumliche Position x , y und z , die durch die Geschwindigkeiten der jeweiligen Koordinatenrichtungen in der Lage \dot{x} und \dot{y} aktualisiert werden.

$$A = \begin{pmatrix} 1 & 0 & 0 & dt & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & dt & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & dt & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & dt & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & dt & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & dt \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (61)$$

Der Zustandsvektor x (62) beinhaltet somit die horizontierten Orientierungen und Dreh-raten sowie die 2D+1D-Position und deren Geschwindigkeiten. Da die Höheninformation allein mit dem Barometer berechnet wird und somit keine direkte Verbindung zur Lageinformation besteht, kann bei einem horizontierten Smartphonekoordinatensystem das 3D-Koordinatensystem in Lage und Höhensystem (2D+1D) aufgeteilt werden.

$$x = \begin{pmatrix} r_{x_{eben}} \\ r_{y_{eben}} \\ r_{z_{eben}} \\ \dot{r}_{x_{eben}} \\ \dot{r}_{y_{eben}} \\ \dot{r}_{z_{eben}} \\ x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} \quad y = \begin{pmatrix} \dot{r}_x \\ \dot{r}_y \\ \dot{r}_z \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ a_m \\ h_b \\ \dot{x} \\ \dot{y} \end{pmatrix} \quad (62)$$

Für die Korrektur in der Estimation stehen die gemessenen Drehraten und die Beschleunigungen sowie die aus den Sensorrohdaten abgeleiteten Größen – wie die Höheninformationen, die Richtungen zu magnetisch Nord und die Geschwindigkeiten im Messwertvektor y – zur Verfügung (62). Die Geschwindigkeiten \dot{x} und \dot{y} werden aus dem Schrittzähler und der Schrittlänge abgeleitet. Die Richtung

zu magnetisch Nord a_m wird aus den verebneten Magnetfeldsensordaten berechnet und die Höheninformation h_b lässt sich aus den Luftdruckunterschieden des Barometers bestimmen.

Die Messmatrix H bildet die Transformationsparameter für die Vergleichbarkeit von Zustandsvektor x und Messwertvektor y ab. Die Horizontierung der Drehraten erfolgt durch die Einbindung einer inversen Drehmatrix, die aus den aktuellen Beschleunigungen berechnet wird. Hierzu wird als Referenzwert die Schwerebeschleunigung $g = 9,81$ m verwendet. Die Korrekturen durch Schrittgeschwindigkeit, Höhen und magnetisch Nordrichtung machen keine aufwendigen Berechnungen zur Transformation erforderlich und werden in H zur Überführung von y nach x mit dem Wert 1 berücksichtigt.

In Kapitel *3.4 Kalibrierung eines Magnetfeldsensors* wird aufgezeigt, dass der Einsatz von Magnetfeldsensordaten zur Richtungsbestimmung in Gebäuden hohe Unsicherheiten hervorrufen kann. In diesem Ansatz wird daher der Magnetfeldsensor vor der Nutzung kalibriert, um während des Einsatzes die Magnitude der korrigierten Sensordaten mit dem örtlichen Referenzwert für das Erdmagnetfeld zu vergleichen. Ist die ermittelte Differenz gering, wird eine Korrektur im KF durch die Richtung zu magnetisch Nord zugelassen. Dies soll den Einfluss künstlicher magnetischer Effekte im Gebäude auf die Korrektur im KF verringern, da hohe Abweichungen des lokalen Magnetfeldes zur Referenzmagnitude auf zusätzliche äußere magnetische Einflüsse im Gebäude schließen lassen. Um magnetisch Nord und die Drehwinkel aus dem Gyroskop zu kombinieren und die Positionsschätzung zu initialisieren, wird zusätzlich die aktuelle Differenz zwischen Nordrichtung und Kartenausrichtung benötigt. Ansonsten sind die aktuelle Ausgangsposition und die Orientierung die Startwerte, von der die Navigation gestartet wird.

In Willemsen u. a. [2013] werden die Ergebnisse eines Versuchs im ehemaligen HCU-Gebäude beschrieben. Festzustellen ist, dass keine Verbesserung der Schätzung durch den Magnetfeldsensor im HCU-Gebäude erzielt werden konnte. Die Stützung durch die Nordrichtung in einer Beispieltrajektorie von ca. 300 m Länge erfolgte nur zweimal während des gesamten Testverlaufs. Zudem zeigt ein Richtungsvergleich der Orientierung des Gyroskops mit magnetisch Nord ein lokales Rauschen der Nordrichtung von bis zu 1 rad (Abb. 5.1). Daher wird in der nun

folgenden Variation des KF auf eine Einbindung des Magnetfeldsensors in dieser Form verzichtet. Zusätzlich wurde in Willemsen u. a. [2013] deutlich, dass die eingebundenen Rohdaten weitgehend nur eine Glättung in Abhängigkeit der Unsicherheiten erfahren und daher eine Einbindung im KF nicht notwendig ist.

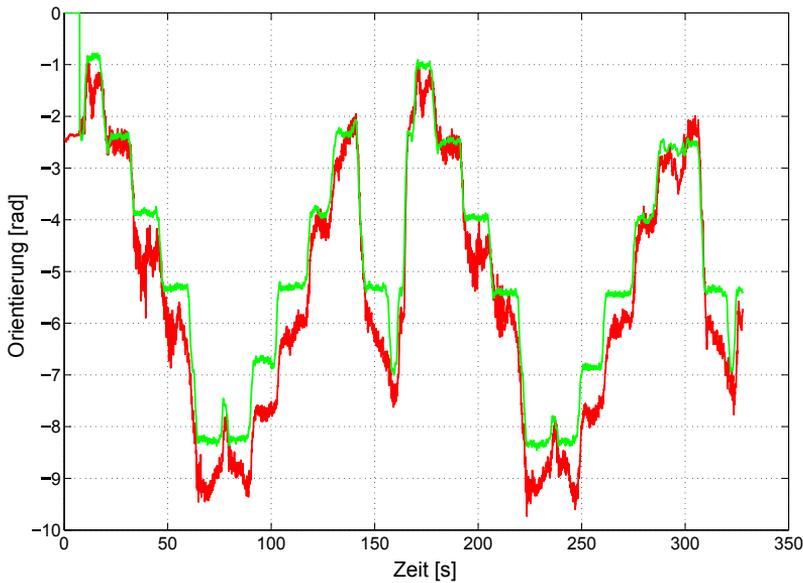


Abb. 5.1: Vergleich der Orientierung aus Magnetfeldsensor (rot) zur Orientierung des Gyroskops (grün) [Willemsen u. a., 2013].

Ansatz mit Korrektur durch das Routing-Netz

Der bereits beschriebene Ansatz aus Willemsen u. a. [2013] wird nun weiter vereinfacht. Dazu wird die Verebnung der Sensoren außerhalb des Filters durchgeführt. Dies erübrigt eine inverse Drehmatrix in der Messmatrix H zur Verebnung der Koordinatensysteme. Zudem werden der Zustandsvektor x und der Messwertvektor y auf die für die Positionsschätzung notwendigen Größen im 2D+1D Koordinatensystem reduziert. Zur zusätzlichen Korrektur wird der KF-Ansatz für den Einsatz des Routing-Graphen erweitert.

Eine Möglichkeit, die MEMS-basierte Positionsschätzung für längere Trajektorien zu verbessern, besteht in der Verwendung eines Routing-Netzes. Das Kanten-Knoten-Modell wird in der Navigationsapplikation für die Routenberechnung be-

nötigt und repräsentiert in Abhängigkeit von der Gebäudestruktur die typischen Laufwege einer Person. Abb. 5.2 zeigt das in dieser Arbeit favorisierte Prinzip der Stützung von Position und Richtung auf Basis der Routing-Kanten. Der blaue Kreis repräsentiert die vorherige Schätzung im KF. Die aktuelle Position (Prädiktion) wird durch den grauen Kreis dargestellt. Die zur Korrektur auszuwählende Routing-Kante wird durch den orthogonalen Abstand zwischen Routing-Kanten und prädizierter Position sowie der Differenz zwischen prädizierter Richtung und Kantenrichtung bestimmt. Die Gleichungen zur Berechnung der notwendigen Größen sind bereits in Kapitel 4.3.1 *Weitere Eingrenzung von Stützungsmöglichkeiten* beschrieben worden.

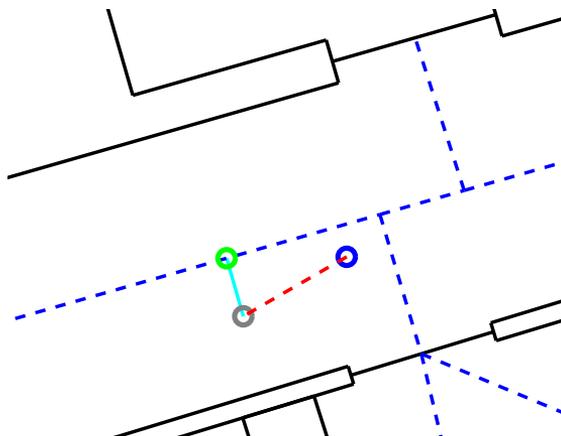


Abb. 5.2: Grafische Darstellung der Verwendung der Kanten des Routingnetzes zur Positionskorrektur im KF (blau: aktuelle Position; grau: neue Schätzung; grün: Fußpunkt auf favorisierter Routing-Kante) [Willemsen u. a., 2015b].

Die Korrektur erfolgt beim KF durch ein Koordinaten-Update mit Hilfe des berechneten Fußpunktes (grüner Kreis) der prädizierten Position auf der ausgewählten Routing-Kante. Aufgrund der starken und auch notwendigen Generalisierung des Routing-Graphen ist eine Richtungskorrektur auf Basis der Routing-Kante besonders in Kurven nicht sinnvoll. Daher wird die Richtungsstützung mit einer Bedingung gekoppelt, die vor einer möglichen Korrektur prüft, ob eine Routing-Kante schon mindestens dreimal hintereinander detektiert wurde. Dies dient dem Ausschluss kurzer Kanten, wie sie vermehrt in Kurven auftreten, um eine Richtungskorrektur während der Drehung zu vermeiden.

Der Zustandsvektor x (63) beinhaltet nun die räumliche Position mit x , y und z , die Lage-Geschwindigkeit mit \dot{x} und \dot{y} sowie die Orientierung $r_{z_{eben}}$ und die Drehrate $\dot{r}_{z_{eben}}$. Die Koordinaten x und y werden durch die Geschwindigkeit propagiert, während die Orientierung durch die Drehrate aktualisiert wird. Die Höhenschätzung z wird nur durch das Barometer gestützt. Sie ist, solange keine Korrektur erfolgt, allein durch die Varianzen-Fortpflanzung mit einer zunehmend höheren Unsicherheit behaftet. Eine Korrektur der Höhe kann allerdings auch als Koordinaten-Update bei der Stützung auf Basis des Routing-Graphen erfolgen, benötigt allerdings eine Erweiterung des Messwertvektors um $z_{routing}$.

$$x = \begin{pmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ r_{z_{eben}} \\ \dot{r}_{z_{eben}} \end{pmatrix} \quad y = \begin{pmatrix} \dot{r}_{z_{eben}} \\ \dot{x}_{step} \\ \dot{y}_{step} \\ x_{routing} \\ y_{routing} \\ h_b \\ r_{routing} \end{pmatrix} \quad (63)$$

Zur Stützung im Estimationsschritt beinhaltet der Messwertvektor y die verebnete Dreh-rate $\dot{r}_{z_{eben}}$ des Gyroskops, die Geschwindigkeiten \dot{x}_{step} und \dot{y}_{step} aus der Schrittdetektion sowie die metrisch berechnete Höhe h_b aus den Barometerdaten. Zudem werden Fußpunktkoordinaten $x_{routing}$ und $y_{routing}$ sowie die Richtung der favorisierten Routing-Kante $r_{routing}$ für die Korrektur durch den Routing-Graphen eingebunden.

Für die Steuerung der Einflussstärke der eingehenden Messungen sind die Vektoren für das Systemrauschen w und für das Messwertrauschen v aufzustellen mit denen sich dann die zugehörigen Kovarianzmatrizen Q_w und R_v aufstellen lassen. Dabei ist die Höhe der Einzelwerte in den Vektoren der Wiederholungsrate des KF anzupassen. In den folgenden Beispielen in dieser Arbeit haben sich die Einzelwerte aus den Formeln in (64) bewährt. Die Wiederholungsrate der Propagation beträgt 0,001 Sekunden und für die der Estimation wird mit 0,01 Sekunden abgefragt, wobei die Schritterkennung nur ca. in 0,5 Sekunden Abständen eingebunden wird.

$$w = \begin{pmatrix} 0,002 \\ 0,002 \\ 0,002 \\ 0,002 \\ 0,002 \\ 0,0005 \\ 0,001 \end{pmatrix} \quad v = \begin{pmatrix} 0,0001 \\ 0,001 \\ 0,001 \\ 0,01 \\ 0,01 \\ 0,01 \\ 0,005 \end{pmatrix} \quad (64)$$

Die A-Matrix (65) repräsentiert das Bewegungsmodell. Sie enthält eine Einheitsmatrix, die einen Zeitschritt dt nur an drei Positionen zur Prädiktion der Koordinaten x und y sowie der Orientierung $r_{z_{eben}}$ enthält. Dass zur Estimation direkt vorverarbeitete Stützdaten Verwendung finden, vereinfacht den Aufbau der Transitionsmatrix H (66). Sie enthält nur Einsen, um den Bezug zwischen Variablen des Zustandsvektors x und des Messwertvektors y herzustellen. Je nach Verfügbarkeit von Stützinformatonen, kann die Transitionsmatrix angepasst werden, um aktuell nicht verfügbare Stützinformatonen aus der Korrektur herauszunehmen. Dazu werden an den entsprechenden Positionen in der Matrix die Einsen durch Nullen ersetzt. Dies ermöglicht eine vergleichsweise einfache Integration der Stützinformatonen, die zu unterschiedlichen Zeitpunkten und in unterschiedlicher Wiederholungsrate zur Verfügung stehen.

$$A = \begin{pmatrix} 1 & 0 & 0 & dt & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & dt & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (65)$$

$$H = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (66)$$

Eine Vorverarbeitung der Sensordaten ist bei diesem KF-Ansatz unerlässlich. Hohe Messraten der MEMS-Sensoren in Smartphones, beispielsweise beim Versuchsgerät Nexus 4, das INS-Daten mit 200 Hz aufzeichnet, erlauben eine Glättung der Rohdaten zur Minimierung des Messrauschens. Die Bestimmung der Neigung aus den Beschleunigungen, die der Transformation der dreiachsigen Sensoren wie Gyroskop und Beschleunigungsmesser dient, wird so präziser. Die Höheninformation wird im KF durch die relativen Höhen aus geglätteten Luftdruckmessungen des Barometers korrigiert. Dazu wird der Luftdruck mit der barometrischen Höhenformel direkt in eine relative Höhe, bezogen auf die Startposition, umgerechnet.

Das KF wird mit der Start-Orientierung und der aktuellen 3D-Position (z.B. aus einem QR-Code) initialisiert. Die Schritterkennung des Beschleunigungsmessers, die in diesem Filter die aktuelle Laufgeschwindigkeit $\dot{x}_{step}, \dot{y}_{step}$ liefert, funktioniert mit Hilfe des in Kapitel 4.1.1 *Pedestrian Dead Reckoning* beschriebenen Schrittzählers (Abb. 4.1). Die Schrittlänge wird in diesem Ansatz als bekannt angenommen.

Topologische Abfragen auf Basis von Karteninformationen – wie etwa das Verbot der Navigation durch Wände hindurch – sind bei diesem KF-Ansatz nicht sinnvoll, da beispielweise Unsicherheiten bei der Positionsberechnung einen Durchbruch durch Wände theoretisch erforderlich machen könnten, damit das Filter nicht divergiert. Daher wird auf eine Korrektur auf Basis von Karteninformationen verzichtet und es kommen nur die Korrekturen mit Hilfe des Routing-Graphen in Betracht.

5.1.2 Ergebnisdiskussion zweier repräsentativer Routen

Datensatz 1 für 2D

Die prinzipielle Einsetzbarkeit des zuvor beschriebenen KF soll nun am Beispiel zweier repräsentativer Routen untersucht werden. Die Routen erstrecken sich für eine typische Navigationsdauer durch das HCU-Gebäude. Sie durchqueren alle vorhandenen Raumtypen: offene Bereiche ohne Routing-Graph, offene Bereiche mit Routing-Graph, schmale Flure, Treppenhaus und Fahrstuhl. Zur Vergleichbarkeit werden alle folgenden Algorithmen mit den hier vorgestellten Datensätzen

getestet. Die erste Route erstreckt sich über ca. 400 m im vierten Obergeschoss des HCU-Gebäudes. Start- und Endpunkt der Trajektorie sind identisch. An einer Position in der Trajektorie wurde von der Testperson eine Pause von mehreren Minuten eingelegt, sodass die reine Laufzeit ca. 8 Minuten beträgt. Währenddessen wurden im Smartphone die Sensordaten aufgezeichnet und anschließend mittels Mathworks Matlab ausgewertet. Die aufgezeichneten Daten wurden in den Algorithmen zeitabhängig verarbeitet, um der Verfügbarkeit von Sensordaten in Anlehnung an einer Applikation auf dem Smartphone zu entsprechen.

Abb. 5.3 zeigt die Positionsschätzung für die erste Route auf Basis des KF. In Schwarz ist der Grundriss des vierten Obergeschosses des im Jahr 2014 bezogenen HCU-Gebäudes zu erkennen. Die Start- und Endposition ist durch einen Kreis in Magenta markiert, die aus dem KF berechnete Trajektorie ist rot dargestellt. Die grünen Punkte auf den Routing-Kanten stehen für die zur Korrektur verwendeten Fußpunkte der Stützung durch den Routing-Graphen. Auf eine Darstellung der Referenzstrecke wird aus Gründen der Übersichtlichkeit verzichtet, da die geringen Abweichungen zur Referenzstrecke die real gelaufene Teststrecke erkennen lassen. Im Nordteil des vierten Obergeschoss befinden sich an den Enden der Flure Aufenthaltsbereiche, in denen keine Routing-Kanten den Laufweg stützen könnten.

In diesem Beispiel wird die Position durch eine Routing-Kante regelmäßig korrigiert, die Richtungskorrektur findet aber nur unregelmäßig bei langen Kanten statt. An den Positionen, an denen keine Routing-Kanten existieren, kann auch keine Korrektur stattfinden. Dies schlägt sich in der Qualität der Trajektorie nieder, die besonders in offenen Bereichen ungenau wird und im linken oberen Bereich der Abbildung einen Durchbruch durch die Außenwand erfährt. Mit der Höhe des orthogonalen Abstands zur Auswahl einer Routing-Kante lässt sich die Positionskorrektur mit Routing-Kante steuern. Wird der Suchradius erhöht, erlaubt dies der abweichenden Trajektorie – im linken oberen Bereich der Abbildung – schneller den richtigen Weg abzubilden. In einem Gebäude mit parallelen Gängen kann dies aber auch zu einer Fehlzuordnung führen. Grundsätzlich sollten die Zeiträume kurz gehalten werden, bei denen die Position keine Stützung im KF erhält, damit die Unsicherheiten der Sensoren nur eine geringe Abweichung von der tat-

sächlichen Route zur Folge haben. Dies hilft einer anschließenden Zuordnung der Routing-Kanten bei der erneuten Korrektur. Das Beispiel zeigt, dass eine auf dem KF basierende Positionsschätzung funktioniert, sich allerdings eine raumgenaue Positionsschätzung nicht gewährleisten lässt. Auf einen Koordinatenvergleich zur Referenztrajektorie wird an dieser Stelle verzichtet. Sie ist in Kapitel 5.4 *Vergleich und Diskussion der vorgestellten Ansätze* ausführlich dargestellt.

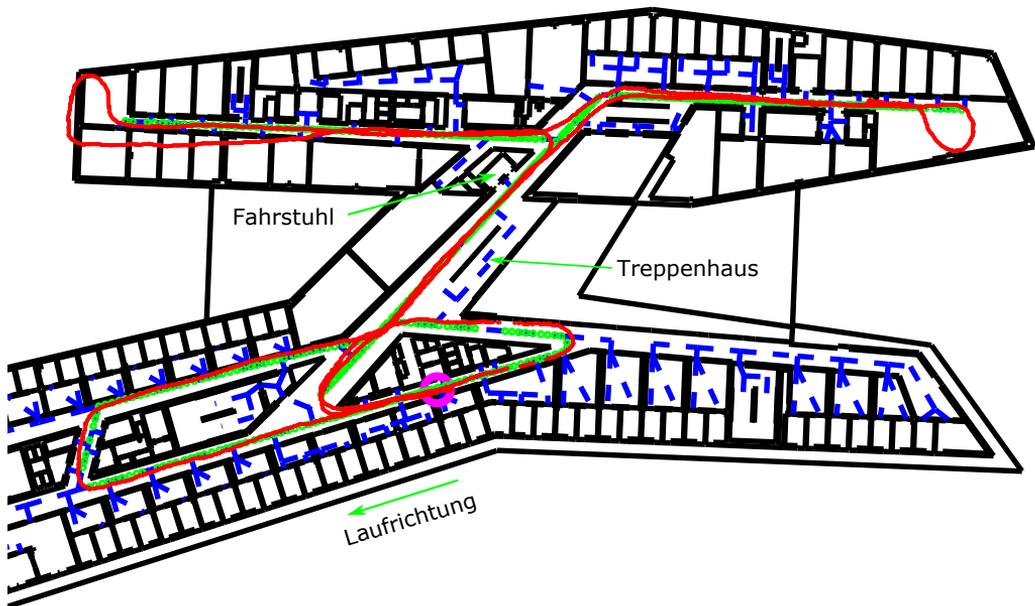


Abb. 5.3: Positionsschätzung mit Kalman Filter im 4. OG der HCU (rot: geschätzte Trajektorie; grün: Fußpunkte auf den Routing-Kanten, die zur Positionskorrektur im Filter eingesetzt wurden) [Willemsen u. a., 2015b].

Stockwerks- und Treppenerkennung

Eine Positionsschätzung im 2D-Raum lässt sich vergleichsweise einfach realisieren. Der Übergang zwischen den Stockwerken bildet den zusätzlichen 1D-Raum ab, der für einen 3D (2D+1D) Ansatz und damit für eine vollständige Innenraumnavigation notwendig ist. Die Höhe zur Detektion der Stockwerke wird aus relativen Luftdruckmessungen des integrierten Barometers berechnet. Diese Höhe lässt sich unabhängig von der Lage berechnen, deshalb kann sie zur Korrektur der 2D-Positionsschätzung dienen. Beispielsweise ist eine Höhenänderung hilfreich,

um Treppen oder Fahrstühle in der Umgebung zu identifizieren. Diese Möglichkeit wird in den folgenden Ansätzen verwendet, um Unsicherheiten der Positionsschätzung beim Wechsel der Stockwerke per Positionskorrektur (CUPT – Coordinate UPdaTe) zu minimieren.

Für die Korrektur der Position ist die Unterscheidung von Fahrstuhl und Treppe auch insofern entscheidend, als beide oftmals nur wenige Meter voneinander entfernt sind. Zu ihrer Unterscheidung findet der Schrittzähler Anwendung. Wird eine Höhenänderung aus relativen Luftdruckmessungen erkannt, kann durch den Schrittzähler die Trajektorie der Treppe oder dem Fahrstuhl zugeordnet werden. Die Korrektur der Position ist beim Fahrstuhl aufgrund der geringen Größe relativ einfach. Schwieriger ist die Anpassung der Positionsberechnung bei Treppen, da sich die Schrittlänge hier verändert. Die Schrittlänge muss entweder angepasst werden oder aber die Positionskorrektur muss beim Übergang der Stockwerke die Unsicherheiten durch fehlerhafte Schrittlängen abfangen. Grundsätzlich gilt, je früher eine Treppe erkannt wird, desto geringer sind die Unsicherheiten durch fehlerhafte Schrittlängen.

Aufgrund des Rauschens der relativen Höhenbestimmung mit dem Barometer von ca. 1 m ist eine Treppenerkennung frühestens nach ungefähr sieben Stufen möglich. Dies bedeutet bei einer durchschnittlichen Schrittlänge von 0,75 m und der durchschnittlichen Länge einer Treppenstufe von 0,25 m einen Positionsfehler von mindestens 3,15 m. Bei breiten, gegenläufigen Treppenhäusern minimieren sich die Fehler, sofern ein Richtungswechsel vorgenommen und eine gegensätzliche gleiche Anzahl an Treppen abgelaufen wird. Zudem dürfen auf den gemessenen Luftdruck für diesen Zeitraum keine zusätzlichen Einflüsse wirksam sein, damit die Übergänge in gleicher Zeitfolge erkannt werden.

Im Folgenden wird ein Ansatz auf Basis von Luftdruckmessungen untersucht, der ein möglichst frühzeitiges Erkennen von Treppenstufen erlaubt. Da das Schrittmuster des Beschleunigungsmessers aufgrund des hohen Messrauschens und der variierenden Schrittmuster keine signifikanten Unterschiede zwischen Flur und Treppe herstellen lässt, wird ein zusätzlicher Einsatz des Beschleunigungsmessers für diese Unterscheidung ausgeschlossen.

Zur frühzeitigen Erkennung von Höhenänderungen werden die stark rauschenden

Barometerdaten gemittelt und die Standardabweichung S_0 für eine Beobachtung berechnet. Bei einer Höhenänderung steigt die Unsicherheit des Mittels rapide an, da die Höhenänderung keine normalverteilte Mittelbildung repräsentiert. Überschreitet das S_0 einen Grenzwert, so wird das als Detektion einer Höhenänderung verstanden und die Schrittlänge von diesem Zeitpunkt an durch die Treppenstufenlänge ersetzt. Für das KF bedeutet das eine Anpassung der Laufgeschwindigkeit, vorausgesetzt es werden während der Höhenänderung gleichzeitig Schritte detektiert.

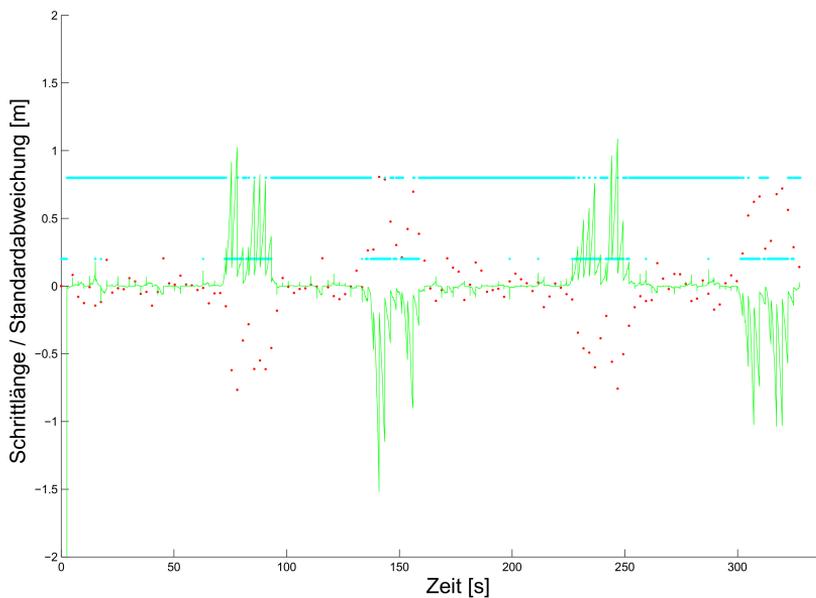


Abb. 5.4: Treppenerkennung mit Barometerdaten für einen Datensatz im Gebäude D des ehemaligen HCU-Gebäudes (cyan: Schrittlänge; grün: StdAbw. S_0 aus Daten über einen Zeitraum von 1 s).

In Abb. 5.4 ist ein Ergebnis zur frühzeitigen Erkennung der Treppenstufen abgebildet. Die cyan-farbigen Punkte repräsentieren die Schrittlänge, die grüne Linie ist das S_0 aus den relativen Höhen der vorangehenden Sekunde berechnet. Bei einer Höhenänderung steigt die Standardabweichung des Mittels rasant an. Dies führt trotz hohen Messrauschens zu einer vergleichsweise zügigen Treppenerkennung.

Datensatz 2 für 2D+1D

Der zweite Datensatz für die vergleichende Untersuchung zu den Algorithmen basiert auf einer Trajektorie von 600 m Länge und 10 Minuten Dauer. Start- und Endpunkt sind gleich gewählt und entsprechen der Position des ersten Testdatensatzes. Die Route beginnt im vierten Obergeschoss, dann werden die Stockwerke über das Haupttreppenhaus bis hin zum Erdgeschoss abgescritten, zum Schluss wird der Fahrstuhl verwendet, um in das vierte Obergeschoss zurückzukehren.

Abb. 5.5 zeigt die Höhen aus dem KF. Nach dem Initialisierungsschritt, bei dem Position und Richtung dem Filter übergeben werden, wird die aktuelle Stockwerkshöhe initialisiert. Nach ca. 100 Sekunden wird das vierte Obergeschoss in Richtung des dritten Obergeschosses verlassen. Die grünen horizontalen Linien markieren den realen Stockwerksübergang. Wie schon bei der Untersuchung des Barometers wurden nivellierte Stockwerkshöhen aus einer Höhenübertragung mittels Nivelliergerät und Stahlmessband zur Referenz herangezogen. Die Trajektorie beinhaltet einen längeren Stillstand der Testperson im Erdgeschoss, durch die sich die Aufenthaltsdauer dort gegenüber den anderen Stockwerken verlängert. Die Fahrstuhlfahrt endet im vierten Obergeschoss. In dem vorliegenden Datensatz findet während der gesamten Berechnung keine Reduzierung der Unsicherheiten aus den Luftdruckmessungen durch eine Re-Initialisierung in der Stockwerkshöhen statt. Bei diesem Test zeigt sich im Ergebnis, dass sich das Barometer gut zur Stockwerkserkennung eignet, dass allerdings nach längeren Messzeiträumen und durch unterschiedliche Stockwerksebenen Unsicherheiten in der Höhenbestimmung entstehen. Eine Korrektur der Höhenberechnung sollte daher bei länger andauernden Schätzungen durch Koordinaten-Updates erfolgen.

Zur Darstellung der Trajektorie werden drei repräsentative Stockwerke ausgewählt, in dem Fall des HCU-Gebäudes das vierte und zweite Obergeschoss sowie das Erdgeschoss. Die anderen Stockwerke können vernachlässigt werden, da sie keinen zusätzlichen Erkenntnisgewinn versprechen. In Abb. 5.6 wird die Trajektorie zu Beginn und am Ende der Aufzeichnung angezeigt. Der Start- und Endpunkt entspricht dem des ersten Testlaufs. Die Route verläuft jetzt durch den linken äußeren Gang bis zum Treppenhaus (hier in der Mitte der Abbildung). Dabei wird die Route nicht zum eigentlichen Treppenabgang berechnet, da die Stützung durch

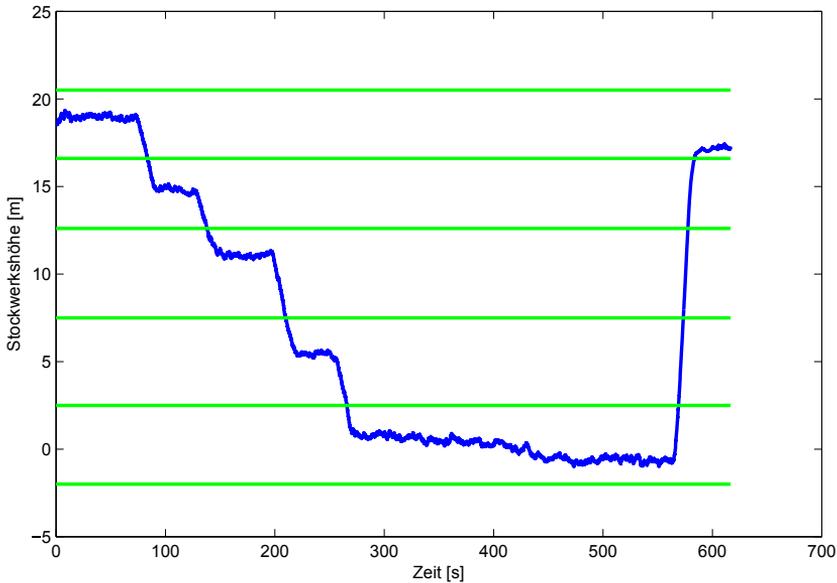


Abb. 5.5: Höhen aus Luftdruckmessungen für den zweiten repräsentativen Datensatz aus der KF-Schätzung (blau: z-Koordinate; grün: Stockwerksübergang aus Referenzmessung).

die aktuell ausgewählten Routing-Kanten die geringe Orientierungsänderung zur Kante hin korrigiert. Je nach Anpassung des Messrauschens zur Berechnung der Korrekturintensität ist der Korrektoreinfluss der Routing-Kanten steuerbar.

Zurück aus dem Erdgeschoss startet die Trajektorie im Fahrstuhl. Die Treppen- und Stockwerkserkennung liefert dazu eine Neu-Initialisierung der Position und Richtungsvorgabe für den Stockwerksübergang. So werden die Positionsunsicherheiten weitestgehend nicht in das nächste Stockwerk übernommen, was den Abschluss der Trajektorie in der Nähe des Referenzpunktes erlaubt.

Abb. 5.7 zeigt die im zweiten Obergeschoss gelaufene Route. Die Testperson kommt die Treppe herunter, biegt nach links ab und nimmt nach der Umrundung des dreieckigen Flurbereichs den direkten Weg zurück zur Treppe. Dabei wird die geringe Richtungsänderung zum Treppenabgang, wie schon im vierten Obergeschoss, aufgrund der Einstellungen im Filter zur Routing-Kante hin falsch

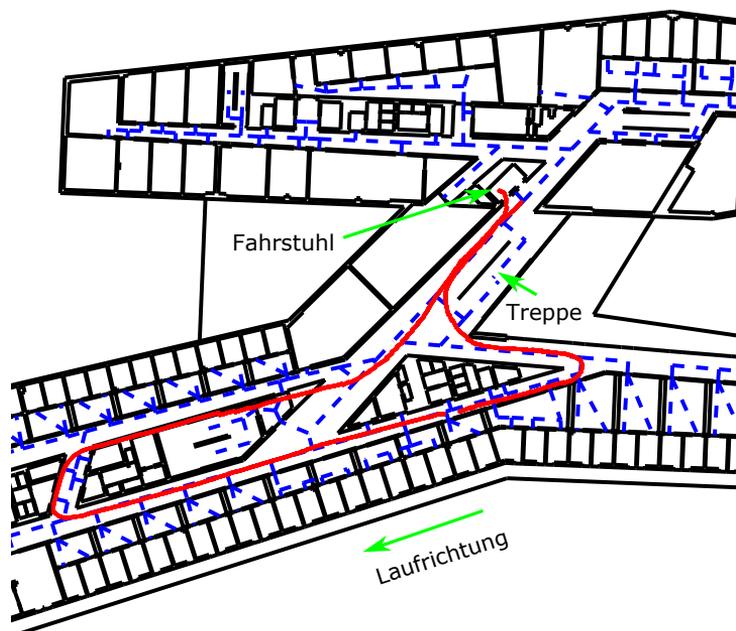


Abb. 5.6: KF-Trajektorie (rot) des räumlichen Testdatensatz für das vierte Obergeschoss. Start und Endpunkt an gleicher Position (blau: Routing-Kanten; schwarz: Grundriss).

korrigiert. Die Position wird aber bei der Stockwerkserkennung für das nächste Stockwerk erneut korrigiert. Zum einen liegt dies an den Stellgrößen zu den Korrekturen durch den Routing-Graphen – wie dem Messrauschen und dem erlaubten orthogonalen Abstand zur Auswahl einer Routing-Kante –, zum anderen gibt es durch den stark generalisierten Routing-Graphen keine direkte Routing-Kante vom Gang zum Treppenhaus, die im KF zur Stützung ausgewählt werden kann.

Die größten Fehler in der Trajektorie entstehen bei diesem KF-Ansatz im Erdgeschoss. Es ist geprägt von großen Freiflächen und enthält wenige Routing-Kanten, da nur wenige Ziele wie beispielsweise Mensa und Bibliothek vorhanden sind. Das KF versucht dennoch die Navigation bezogen auf die Routing-Kanten zu korrigieren. Abb. 5.8 zeigt die Trajektorie für das Erdgeschoss. Sie startet im Erdgeschoss auf der rechten Seite am Treppenhaus. Der große offene Bereich wird in der Nähe der Wände abgelaufen. Allerdings wird die Trajektorie schon zu Beginn, bezogen

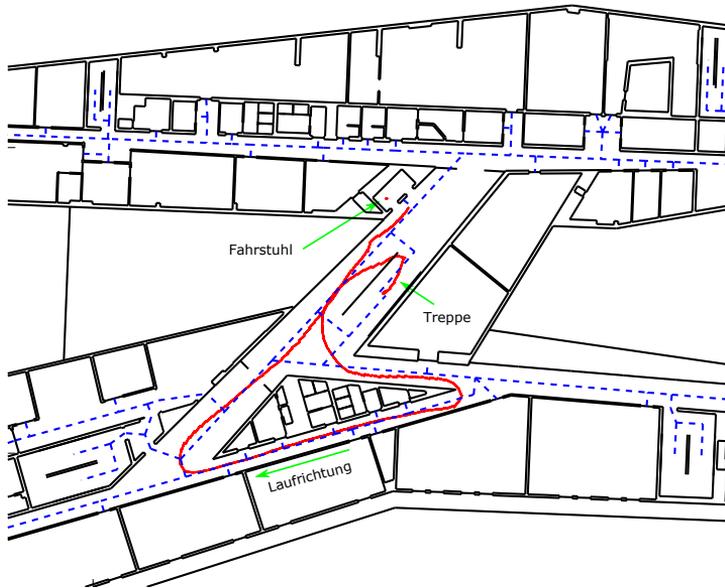


Abb. 5.7: Trajektorie (rot) des räumlichen Testdatensatz für das zweite Obergeschoss auf Basis des KF (blau: Routing-Kanten; schwarz: Grundriss).

auf die mittlere Routing-Kante, gestützt. Nach einigen Metern erfolgt in der Nähe des Fahrstuhls ein Stillstand von wenigen Minuten, der sich nicht negativ auf die Positionsschätzung auswirkt. Besonders bei der Abbiegung im linken Bereich der Trajektorie, nahe des Ausgangs, zeigt sich der Einfluss der Positionskorrektur durch die ausgewählte Routing-Kante, da die Trajektorie deutlich sichtbar zur Routing-Kante korrigiert wird. Nach der Durchquerung der Mensa verläuft die Trajektorie zum Fahrstuhl und mit ihm in das vierte Obergeschoss.

Die Testläufe zeigen im Ergebnis, dass die Korrekturen auf Basis des Routing-Graphen besonders in schmalen Gängen zu guten Verbesserungen der Positionsschätzung im KF-Ansatz führen. Bei gleicher Einstellung der Korrekturintensität im Filter rufen allerdings offene Bereiche mit wenigen Routing-Kanten hohe Messunsicherheiten hervor, sofern der Nutzer vom durch die Routing-Kante definierten Weg zu stark abweicht.

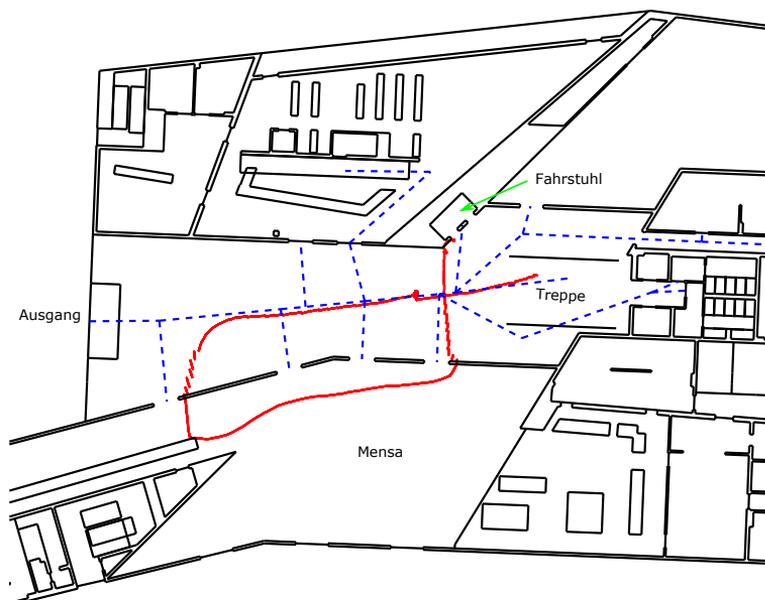


Abb. 5.8: Trajektorie (rot) des räumlichen Testdatensatz für das EG auf Basis des KF (blau: Routing-Kanten; schwarz: Grundriss).

5.2 Partikelfilter

5.2.1 Aufbau des Filters

Partikelfilter können im Gegensatz zum KF multivariate Wahrscheinlichkeiten darstellen und verarbeiten. Sie repräsentieren diese Unsicherheiten in Form von Partikeln, denen jeweils ein Gewicht zugeordnet wird. Diese Partikelgewichte werden durch Berücksichtigung von Korrekturinformation in Verbindung mit der Wahrscheinlichkeitsdichtefunktion (WDF) angepasst. Der prinzipielle Aufbau des PF ist in Kapitel 2.2.2 *Partikelfilter* und von Wendel [2011] beschrieben. Für die MEMS-INS-basierte Positionsschätzung wurde in Anlehnung daran ein Ansatz entwickelt, der Karten- und Routing-Daten zur Korrektur nutzt.

Wie schon beim KF-Ansatz werden die Sensordaten vor dem Einsatz im PF vorverarbeitet. Dazu wird die relative Höhe aus Barometerdaten für die Stockwerkserkennung genutzt, die verebnete z-Achse des Beschleunigungssensors für den Schrittzähler und die verebnete integrierte Drehrate der z-Achse des Gy-

roskops für die Orientierung verwendet. So lässt sich der Aufbau des PF stark vereinfachen und erlaubt, getrennt von Lage und Höhe, eine Positionsschätzung (67). Die Stockwerkserkennung wird außerhalb des PF berechnet und leitet ein Koordinaten-Update bei einem Stockwerksübergang für die Positionsschätzung ein.

$$\vec{p}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} \quad (67)$$

Die Anzahl der verwendeten Partikel im Filter hängt von der angestrebten Auflösung der Position ab und hat daher auch einen Einfluss auf die erreichbare Genauigkeit der Schätzung. Je nach bekannten Vorinformationen der zu schätzenden Parameter kann die Partikelanzahl weiter reduziert werden. Die MEMS-INS-basierte Positionsschätzung benötigt eine Initialisierung der Position, die auch als Vorinformation im PF dient. Daher werden für den hier entwickelten PF nur 100 Partikel für eine robuste Positionsschätzung benötigt. Eine Erhöhung der Partikelanzahl von bis zu 1000 Partikel führt im folgenden PF zu keiner signifikanten Verbesserung der Positionsschätzung.

Das Bewegungsmodell des PF basiert auf dem PDR, daher kann solch ein PF auch als PDR-PF benannt werden. Die Positionsberechnung wird mithilfe der Formel (68) auf Basis der Schritterkennung generiert. Die Laufrichtung r_i jedes Partikels und die Schrittlänge l_{step} werden jeweils mit einer definierten Unsicherheit als normalverteiltes Rauschen ϵ hinzuaddiert. Das erlaubt eine Verteilung der Partikel in einem Fächer in Laufrichtung des Nutzers und reduziert die Anzahl notwendiger Partikel i . Die Schrittlänge l_{step} wird wie beim KF-Ansatz als bekannt angenommen.

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} x_i^- \\ y_i^- \end{pmatrix} + \begin{pmatrix} \cos(r + \epsilon_{r_i}) & \sin(r + \epsilon_{r_i}) \\ -\sin(r + \epsilon_{r_i}) & \cos(r + \epsilon_{r_i}) \end{pmatrix} * \begin{pmatrix} l_{step} + \epsilon_{l_i} \\ 0 \end{pmatrix} \quad (68)$$

Die initialisierte Partikelwolke muss für einen PF grundsätzlich so stark gestreut sein, dass sich die zu schätzende Position mindestens durch einen Partikel repräsentieren lässt. Da für die Routenberechnung in der Indoor-Navigation und für

das PDR eine exakte Startposition notwendig ist, entfällt eine starke Streuung der Partikel. Der PF wird zu Beginn mit einer bekannten Position und Richtung initialisiert.

Bei jeder Schritterkennung werden auf Basis des PDR die Partikel propagiert. Anschließend können Grundriss und Routing-Graph zur Korrektur der Position eingebunden werden. Der Grundriss eines Stockwerks dient zum einen der topologischen Bedingung, dass nicht durch Wände gegangen werden kann, und zum anderen der Richtungskorrektur durch naheliegende Wände. Die Routing-Kanten befinden sich weitgehend auf den typischen Wegen im Gebäude und können daher, wie im KF, auch zur Positions- und Richtungskorrektur eingesetzt werden.

Das Prinzip der Korrektur durch Wände aus vektorbasierten Grundrissdaten zeigt Abb. 5.9. Es wurde bereits in Willemsen u. a. [2015a] beschrieben. Von der letzten wahrscheinlichsten Position (blau) werden aus dem gewichteten Mittel aller Partikel für den nächsten Berechnungsschritt die Partikel mit Schrittlänge und Orientierung propagiert. Bevor ein Resampling stattfindet, werden die Gewichte für jedes einzelne Partikel in Hinblick auf vorhandene Stützinformationen neu berechnet. In diesem Beispiel besitzen die roten Partikel ein sehr geringes Gewicht, da sie abhängig von der vorherigen wahrscheinlichsten Position hinter einer Wand liegen und eine Bewegung durch eine Wand nicht möglich ist. Die Berechnung der Schnittpunkte erfolgt nach den Formeln in Kapitel 4.3.1 *Weitere Eingrenzung von Stützungsmöglichkeiten*.

Die übrigen Partikel sind je nach Höhe des Gewichts in Graustufen dargestellt. Jedes Partikel bildet mit der letzten wahrscheinlichsten Position eine Linie. Die Differenz der Orientierung dieser Linie r_i und der für jedes Partikel ausgewählten Wandrichtung r_{wand_j} bildet den Bezugswert für die Berechnung der Gewichte mit der angepassten WDF. Die Berechnung der Gewichte w_i erfolgt nach (69). Das Messrauschen R_r repräsentiert die Unsicherheiten dieses Richtungsvergleichs und dient gleichzeitig zur Steuerung der Korrekturstärke. Die Auswahl der Bezugswand erfolgt für jedes Partikel gesondert und ist abhängig von der Höhe des orthogonalen Abstands.

$$w_i = \exp[-0,5 * (r_i - r_{wand_j}) * R_r^{-1} * (r_i - r_{wand_j})] \quad (69)$$

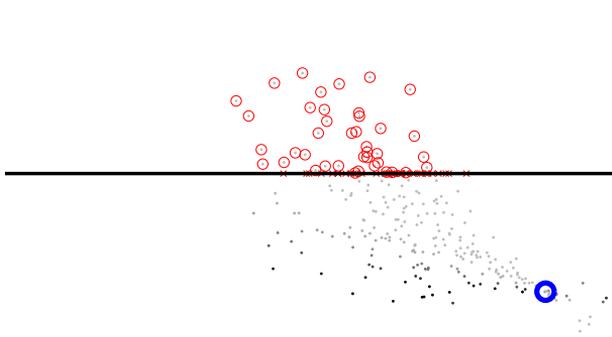


Abb. 5.9: Streuung von Partikeln auf Basis von Schrittlänge und Richtung sowie die anschließende Gewichte-Berechnung durch Wandinformationen des Grundrisses. (blau: aktuelle wahrscheinlichste Position; rot: Partikel mit minimalem Gewicht, da hinter der Wand liegend; Graustufen: Färbung nach Höhe des Gewichts, berechnet in Abhängigkeit der Richtungsdifferenz zwischen Wand und Partikellinie) [Willemsen u. a., 2015b].

Der Routing-Graph bietet eine weitere Möglichkeit zur Stützung der Position während einer Fußgängernavigation, da er die möglichen und typischen Wege im Gebäude abbilden kann. Wie bereits erwähnt ist aber die genaue Repräsentation des gelaufenen Weges – besonders in offenen Bereichen und in Kurven – nicht möglich, was in diesen Bereichen eine Verschlechterung der Schätzung durch die Korrektur bedeuten kann. Die Auswahl der Routing-Kante zur Stützung erfolgt daher in Abhängigkeit des orthogonalen Abstands des Partikels zur jeweiligen Routing-Kante sowie durch die Richtungsdifferenz zwischen Kantenrichtung und Partikelrichtung. Die Berechnungen zur Einbindung von Routing-Kanten sind in Kapitel 4.3.1 *Weitere Eingrenzung von Stützungsmöglichkeiten* beschrieben.

Abb. 5.10 zeigt die orthogonalen Abstände einer Partikelwolke, bezogen auf eine favorisierte Routing-Kante. In blau ist dabei die letzte wahrscheinlichste Position dargestellt. Die propagierten Partikel sind in Laufrichtung mit veranschlagter Richtung und Schrittlänge bestimmt. Für jedes Partikel wird der orthogonale Abstand zur Routing-Kante berechnet. Die Länge des orthogonalen Abstandes d_{ortho} wird verwendet, um mit der Gleichung (70) die Höhe des jeweiligen Partikelgewichts

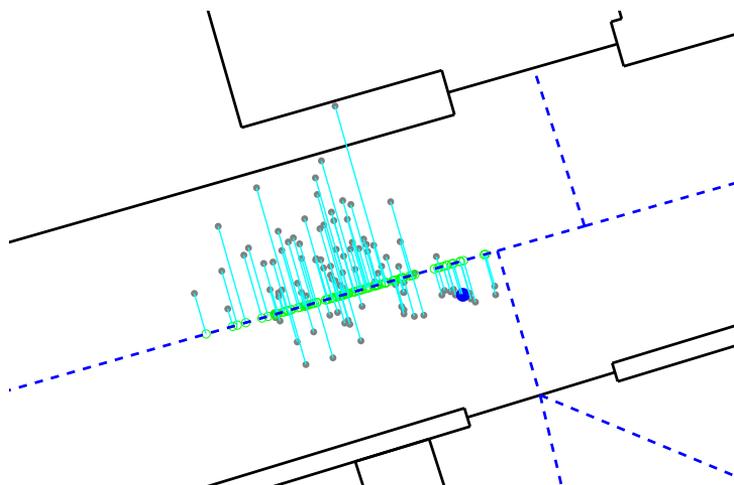


Abb. 5.10: Prinzip einer Gewichte-Berechnung der Partikel im PF auf Basis von Routing-Kanten (blauer Punkt: aktuelle Position; grau: gestreute Partikel; cyan: orthogonale Abstände zu ausgewählter Routing-Kante) [Willemssen u. a., 2015b].

w_i zu bestimmen. Mit R_{ortho} wird die Unsicherheit des orthogonalen Abstands abgebildet und damit die Intensität der Korrektur gesteuert.

$$w_i = \exp[-0,5 * (d_{ortho}) * R_{ortho}^{-1} * (d_{ortho})] \quad (70)$$

Durch die Korrektur auf Basis der Routing-Kante werden Partikel nahe der Routing-Kante höher gewichtet, dadurch wird das gewichtete Mittel der Schätzung zu den typischen Laufwegen basierend auf den Routing-Graphen gewichtet. Wird eine Routing-Kante mindestens dreimal ausgewählt, erfolgt eine Richtungskorrektur auf Basis der Richtungsdivergenz zwischen aktueller Partikelrichtung und Kantenrichtung.

Nach Berücksichtigung aller Korrekturmöglichkeiten werden die berechneten Partikelgewichte w_i für jedes Partikel multipliziert. Anschließend erfolgt eine Normierung der Partikel auf Basis der multiplizierten Gewichte. Mit diesen Gewichten wird das Resampling vorgenommen oder direkt das gewichtete Mittel der Partikel für den jeweiligen Berechnungsschritt gebildet, um die favorisierte Trajektorie zu erzeugen.

5.2.2 Ergebnisdiskussion zweier repräsentativer Routen

Datensatz 1 für 2D

Der entwickelte PF wird anhand der zwei in Kapitel 5.1.2 *Ergebnisdiskussion zweier repräsentativer Routen* vorgestellten repräsentativen Testdatensätze untersucht. Bei dem ersten Testdatensatz findet kein Stockwerkswechsel statt und die Trajektorie gilt ausschließlich für das vierten Obergeschoss des HCU Gebäudes. In Abb. 5.11 ist das Ergebnis der echtzeitfähigen Berechnung zu sehen. Die aus den Partikeln gewichteten Positionen werden durch rote Punkte markiert. Die für jeden Schritt verwendeten Partikel sind hier blau. Die Trajektorie auf Basis eines PDR, die hier zu Vergleichszwecken abgebildet ist, ist in Magenta dargestellt.

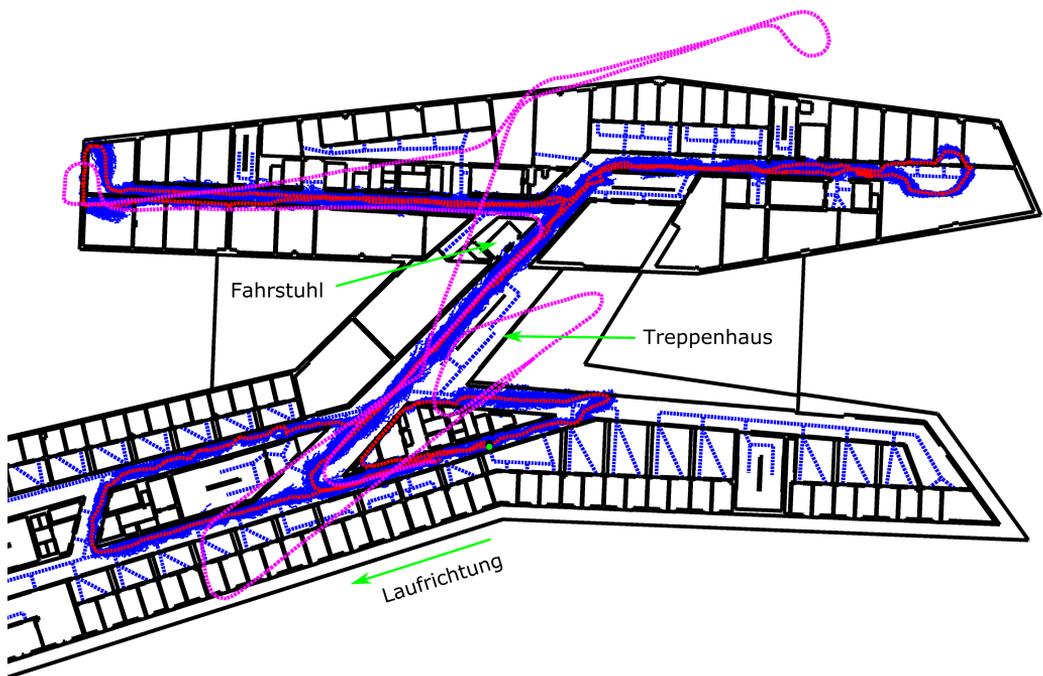


Abb. 5.11: Trajektorie für das vierte Obergeschoss aus einer Positionsschätzung auf Basis von PF mit Stützung durch Routing- und Kartendaten (blau: Partikel, rot: gewichtetes Mittel jedes Berechnungsschrittes im PF; magenta: Pedestrian Dead Reckoning ohne Korrekturen) [Willemsen u. a., 2015b].

Im Vergleich zum PDR ist gut zu erkennen, dass die Stützung aus Karte und Routing eine deutliche Verbesserung der real gelaufenen Trajektorie bewirkt. An den Enden der Flure, die keine Routing-Kanten enthalten – beispielsweise im linken oberen und rechten oberen Bereich der Abbildung – werden die Trajektorien nicht durch Routing-Kanten gestützt. In diesen offenen Bereichen wird lediglich die beschriebene Korrektur durch Grundrissinformationen durchgeführt.

Besonders in diesen Bereichen können Durchbrüche der Trajektorie durch Wände entstehen, wie hier im rechten oberen Bereich zu sehen ist. Ein Durchbruch darf nicht gänzlich unterbunden werden, damit sich fehlerhafte Schätzungen, die sich in Nebenräumen befinden nicht weiter verschlechtern. Die Partikelgewichte bei der Wandstützung werden daher nicht auf null gesetzt, sondern erhalten sehr geringe Gewichte, so dass ein Durchbruch möglich ist, sich aber durch den Einfluss weiterer höher gewichteter Partikel vermeiden lassen könnte. Die Trajektorie lässt sich somit durch die Wand berechnen, um, wie in der Abbildung auf der rechten oberen Seite zu erkennen, den richtigen Weg wieder zu erreichen.

Datensatz 2 für 2D+1D

Der zweite räumliche Datensatz enthält zusätzlich die im KF-Ansatz beschriebene Treppen- und Stockwerkserkennung mithilfe des Barometers. Position und Richtung werden beim Stockwerksübergang korrigiert. Die Trajektorie beginnt und endet im vierten Obergeschoss. Auf eine Darstellung der Höhen aus relativen Luftdruckmessungen wird verzichtet, da es keine signifikanten Veränderungen zu den KF-Ergebnissen aus Abb. 5.5 gibt. Als Auszug aus der Trajektorie werden die Obergeschosse 2, 4 und das Erdgeschoss ausgewählt.

Abb. 5.12 zeigt das vierte Obergeschoss mit Anfang und Abschluss der Trajektorie. Wegen der Positionskorrektur nach dem Stockwerkswechsel sind die Ergebnisse mit den KF-Ergebnissen vergleichbar. Anders als beim KF-Ansatz wird hier die Trajektorie bis zum Treppenabgang korrekt berechnet. Im Vergleich zu den Ergebnissen des KF (Abb. 5.6) sind ansonsten keine Unterschiede in der Qualität der Trajektorie erkennbar. Dies deutet auf einen hohen Einfluss der Positionskorrektur durch die Routing-Kanten hin, da die Nutzerpositionen in schmalen Gängen gut den Routing-Kanten zugeordnet werden können.

Im dritten Obergeschoss verläuft die Trajektorie durch einen langen Gang und mit

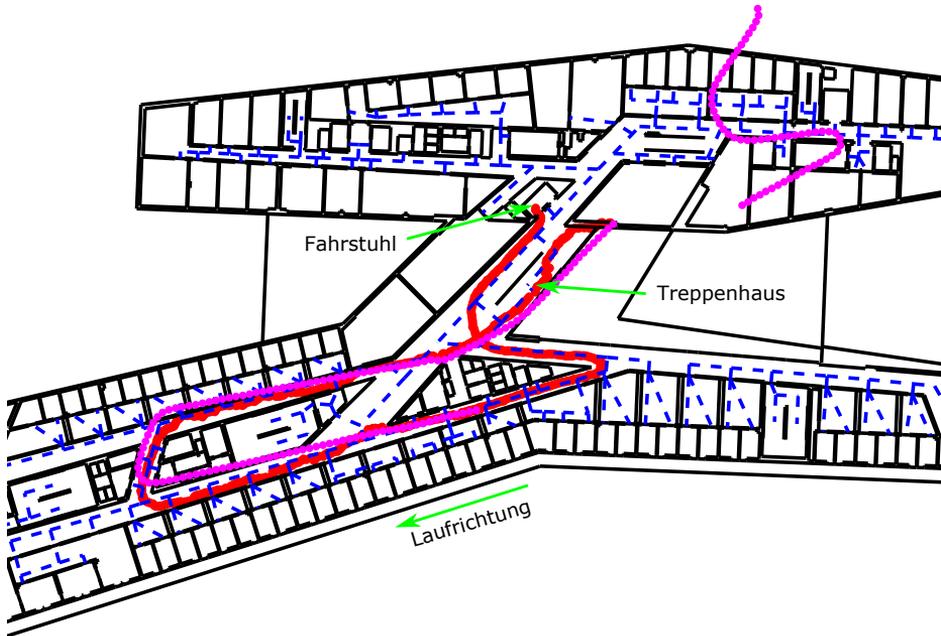


Abb. 5.12: Anfang und Ende der Trajektorie des zweiten Testdatensatzes im vierten Obergeschoss auf Basis von PF mit Positionskorrektur durch Routing- und Kartendaten (rot: gewichtetes Mittel jedes Berechnungsschrittes im PF; magenta: PDR).

einer Kehrtwende zurück zur Treppe. Anschließend wird das zweite Obergeschoss betreten und eine Runde um den dreieckigen Innenbereich abgelaufen (Abb. 5.13). Im Treppenabgang fällt besonders auf, dass die rote Trajektorie weit über das Treppenhaus hinausreicht. Die Unsicherheiten der relativen Luftdruckmessungen führen zu einer vergleichsweise späten Treppenerkennung, sodass viele Schritte auf den Treppen mit einer falschen Schrittlänge berechnet wurden. Zusätzlich bewirkt die zeitlich versetzte Erkennung des Stockwerkswechsels eine verlängerte Positionsschätzung im Ausgangsstockwerk.

Das Erdgeschoss des HCU-Gebäudes ist, wie dargelegt, von großen offenen Bereichen und wenigen Routing-Kanten geprägt. In Abb. 5.14 ist die Trajektorie aus dem PF rot dargestellt. Der Treppenabgang wird kaum von der Routing-Kante beeinflusst und endet in einer Punktwolke, die den mehrere Minuten andauernden Stillstand repräsentiert. Für wenige Minuten hat zwar eine Positionsschät-

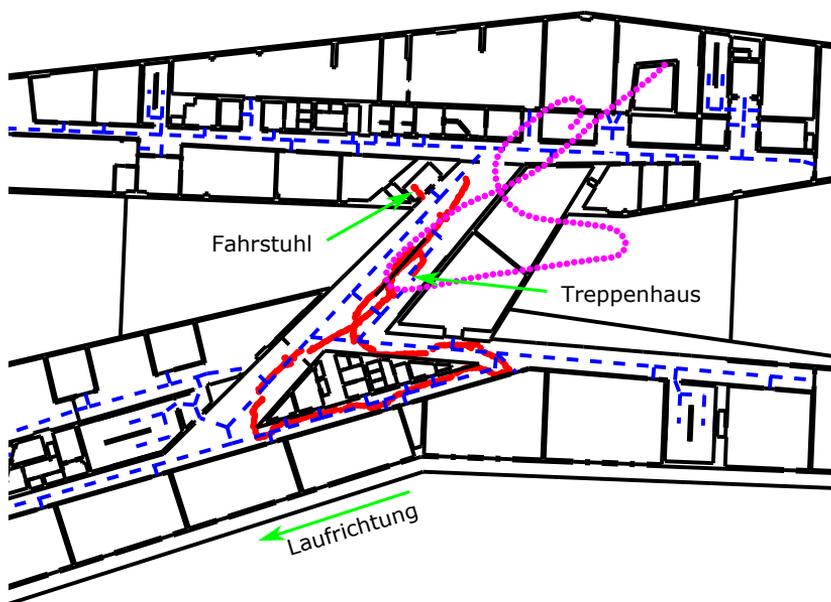


Abb. 5.13: Geschätzte Trajektorie des zweiten Testdatensatzes im zweiten Obergeschoss auf Basis von PF mit Positionskorrektur durch Routing- und Kartendaten (rot: gewichtetes Mittel jedes Berechnungsschrittes im PF; magenta: PDR).

zung stattgefunden, sie wurde allerdings nicht per Schritterkennung fortgeführt, sie wird aber durch die Korrektur mit den Routing-Kanten über den gesamten Zeitraum beeinflusst.

Daran lässt sich der Einfluss der Korrektur auf die Partikelwolke über einen längeren Zeitraum beobachten. Je nach Intensität des definierten Rauschens der Stützung kann die Korrektur mehr oder weniger stark ausfallen. Anschließend führt die Trajektorie durch den Mensabereich zurück zum Fahrstuhl. Wie zuvor beim KF-Ansatz wird der reale Weg durch den großen offenen Bereich im Vergleich zu allen anderen Stockwerken am wenigsten eingehalten. Beim PF-Ansatz liegen die Gründe dafür vor allem im Einfluss der Routing-Kanten in der Positionskorrektur.

5.3 Topologischer Ansatz auf einem Routing-Graphen

An dieser Stelle lässt sich festhalten, dass mit Kalman Filter und Partikelfilter eine autonome Positionsschätzung durch das Smartphone prinzipiell möglich ist. Sie ist

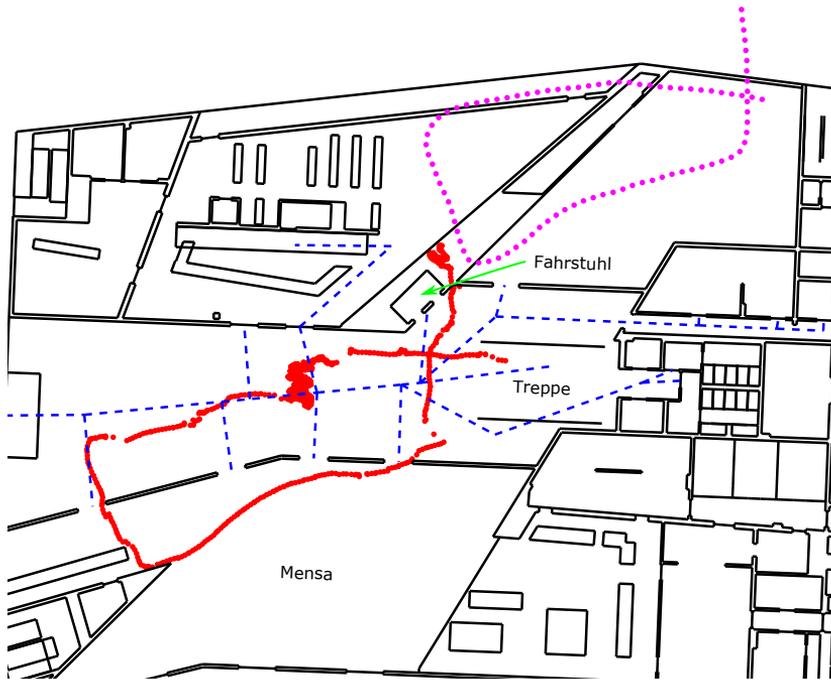


Abb. 5.14: Erdgeschoss der Trajektorie des zweiten Test-Datensatzes auf Basis von PF mit Positionskorrektur durch Routing- und Kartendaten (rot: gewichtetes Mittel jedes Berechnungsschrittes im PF; magenta: PDR).

allerdings auf eine Stützung durch Kartendaten und/oder Routing-Graphen angewiesen, die einen hohen Rechenaufwand für die Berücksichtigung von Wänden und Routing-Kanten mit sich bringen. Ein Schwachpunkt besteht auch im negativen Einfluss der Stützung in Kurven, wenn die stark generalisierten Routing-Kanten nicht exakt dem typischen Laufweg folgen.

Um diese Nachteile möglichst auszugleichen, wird eine alternative Vorgehensweise untersucht. Dabei wird das Prinzip des Map Matchings, das üblicherweise bei GNSS-gestützter Fahrzeugnavigation zum Einsatz kommt, genutzt. Die Positionsschätzung mit Hilfe von Beschleunigungsmesser und Gyroskop findet nach diesem Ansatz nur auf dem Routing-Graphen (Kanten-Knoten-Modell) statt. Die damit auftretenden Schwierigkeiten, hervorgerufen durch die starke Generalisierung des Graphen, müssen bei der Schätzung berücksichtigt werden.

5.3.1 Zustandsschätzung aus MEMS-Sensordaten

Ein Vorteil einer MEMS-INS-Positionsschätzung auf dem Routing-Graphen besteht darin, dass sie unabhängig von jeder weiteren Stützinformation erfolgen kann. Ein Nachteil ist allerdings die Anpassung der Nutzerbewegung an den generalisierten Routing-Graphen im Gebäude, da der gelaufene Weg nicht mit der Lage der Routing-Kanten übereinstimmen muss. Beim nun beschriebenen Ansatz, dessen Grundkonzeption in Willemsen u. a. [2015b] dargelegt wurde, wird die Navigation mit möglichst wenigen Zustandsabfragen auf den Kanten und Knoten des Routing-Graphen verfolgt. Auf die abgefragten Zustände folgen jeweils unterschiedliche Herangehensweisen bei der Suche und dem Übergang zwischen Routing-Kanten. Dabei soll die Positionsschätzung weitgehend robust erfolgen, um parallel berechnete mögliche Trajektorien zu vermeiden, wie sie beim Map Matching auf Basis eines PF vorkommen [Ascher, 2014].

Für die Positionsschätzung werden die bisher favorisierten MEMS-Sensoren so wie bei den KF- und PF-Ansätzen eingesetzt. Der Barometer liefert relative Höhenänderungen zur Stockwerksunterscheidung. Die vom Gyroskop ermittelten Drehraten werden zu Winkeln integriert. Der Beschleunigungssensor findet Anwendung als Schrittzähler und zur Transformation des Smartphone-Koordinatensystems in den Horizont. Auf eine Schrittlängenschätzung wird verzichtet und stattdessen eine mittlere Schrittlänge angenommen.

Initialisierung der Positionsschätzung

Zu Beginn der Positionsschätzung muss basierend auf einer Startposition die zu dem Zeitpunkt wahrscheinlichste Routing-Kante ausgewählt werden. Die Startposition kann beispielsweise eine Raum- oder Türkoordinate sein, die in einer Applikation per QR-Code am Türschild oder über die Eingabe einer Raumnummer übergeben werden könnte. Die Berechnung zur Auswahl der Routing-Kante erfolgt durch die Bestimmung aller orthogonalen Abstände der Startkoordinate zu den umliegenden Routing-Kanten. Die mögliche Laufrichtung wird bei der ersten Schritterkennung aus der Summe der integrierten Drehraten des Gyroskops und der zuvor übergebenen initialen Richtung berechnet. Der Fußpunkt auf der ausgewählten Routing-Kante ist zugleich der Startpunkt für die relative Positionsschätzung auf dem Routing-Graphen.

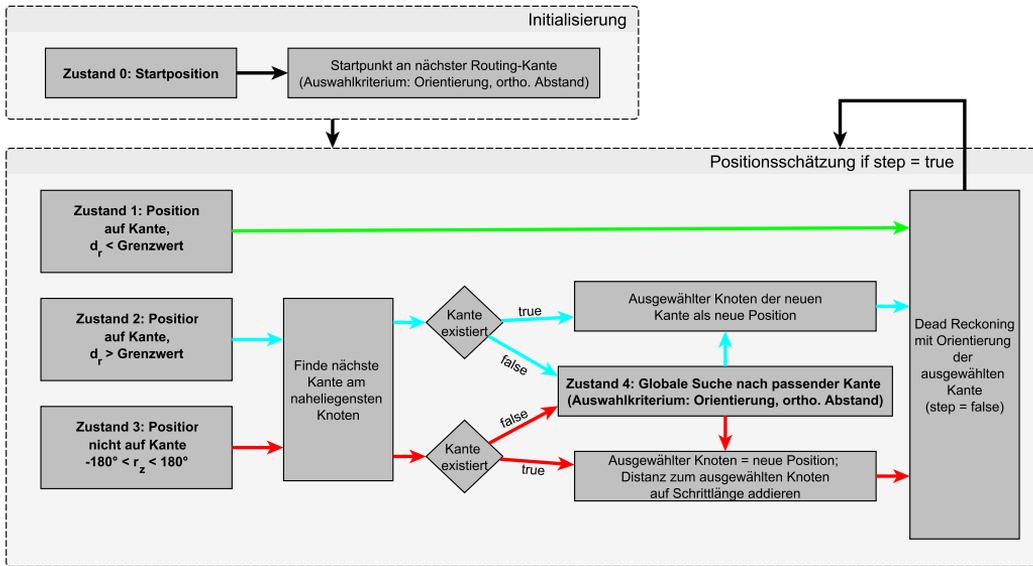


Abb. 5.15: Ablaufdiagramm der Positionsschätzung nach dem topologischen Ansatz auf dem Routing-Graphen, abgeleitet nach [Willemsen u. a., 2015b].

Positionsschätzung

Die Basis der Positionsschätzung bildet das PDR, wobei allerdings die Orientierung zur Berechnung der nachfolgenden Positionen von der aktuell ausgewählten Routing-Kante übernommen wird. Die Orientierung per Gyroskop r_z dient so nur der Auffindung der entsprechenden Routing-Kante, dass dadurch den Betrachtungszeitraum der integrierten von Drift beeinflussten Drehwinkel sinken lässt. Abb. 5.15 und Abb. 5.16 zeigen ein Ablaufdiagramm und die grafische Umsetzung dieses Ansatzes.

Die Initialisierung zu Beginn liefert die erste Bezugskante für das kantenbasierte PDR. Anschließend wird eine Positionsschätzung nach jedem detektierten Schritt durchgeführt, daher entfällt eine zusätzliche Zustandserkennung für den Stillstand. Die Positionsschätzung erfolgt nach vier klar voneinander trennbaren Zuständen.

Im *Zustand 1* wird eine Fortbewegung entlang der aktuell gewählten Kante verfolgt. Dabei wird eine Unsicherheit für die Detektion einer Kantenrichtung mit 10° angenommen. Dieser Wert beschreibt den Unsicherheitsbereich der Orientie-

rungsänderung. Die aktuelle Kante bleibt ausgewählt, solange die Richtungsdivergenz d_r den Grenzwert nicht überschreitet und sich die aktuelle Position zwischen den Knotenpunkten der Routing-Kante befindet. Der Grenzwert ist variabel einstellbar und in erster Linie abhängig von der vorherrschenden Gebäudestruktur. Bei einem Gebäude mit ausschließlich rechtwinkligen Routing-Kanten kann der Grenzwert auf beispielsweise 45° erhöht werden. Sind die genannten Kriterien erfüllt, findet nach Formel (71) das PDR auf der Routing-Kante statt. In die Rotationsmatrix (45) wird nun die Richtung der favorisierten Routing-Kante eingebunden und daher im folgenden mit $R_{z_{Kante_k}}$ bezeichnet. Anschließend wird die Schrittdetektion wieder aktiv.

$$x_i = x_{i-1} + R_{z_{Kante_k}} * t_{step} \quad (71)$$

Der *Zustand 2* beschreibt eine Richtungsänderung zwischen zwei Schritten auf der ausgewählten Bezugskante. Die Laufrichtung ist in diesem Fall mit der zugeordneten Routing-Kante nicht mehr vereinbar, da die Richtungsdivergenz nun die für den Grenzwert festgelegten 10° beträgt. Daher wird der Knoten mit der kürzesten Distanz zur aktuellen Position der zugeordneten Routing-Kante ausgewählt und anschließend die per Gyroskop ermittelte Richtung mit allen Kanten, die vom Knoten abgehen, verglichen. Ausgewählt wird dann die Kante mit der niedrigsten Richtungsdivergenz, die sich innerhalb des Grenzwerts von 10° befindet. Für die Positionsberechnung wird im ersten Schritt der Knoten $P_{k,j}$ dieser Kante als neue Position angenommen und das PDR von dort entlang der neuen Routing-Kante ausgeführt (72). Dies minimiert den Einfluss der stark generalisierten Laufwege gemäß dem Routing-Graphen. Befindet sich keine Routing-Kante innerhalb des Suchbereichs von 10° , so wird der *Zustand 4* aktiv.

$$x_i = P_{k,j} + R_{z_{Kante_k}} * t_{step} \quad (72)$$

Im *Zustand 3* liegt die aktuelle Position nicht mehr auf der Kante. Wie im *Zustand 2* wird eine neue Routing-Kante vom nächstliegenden Knoten her gesucht. Ist eine neue Kante ausgewählt, so muss der zu weit gelaufene Weg berücksichtigt werden. Dies geschieht im PDR durch die Summe aus dem Abstand $d_{P_{k,j},x_{i-1}}$ der letzten

Positionsschätzung zum ausgewählten Knoten und der Schrittlänge (73). Befindet sich keine Routing-Kante innerhalb des Suchbereichs von 10° , wird der *Zustand 4* aktiv.

$$x_i = P_{k,j} + R_{z_{Kante_k}} * (t_{step} + d_{P_{k,j},x_{i-1}}) \quad (73)$$

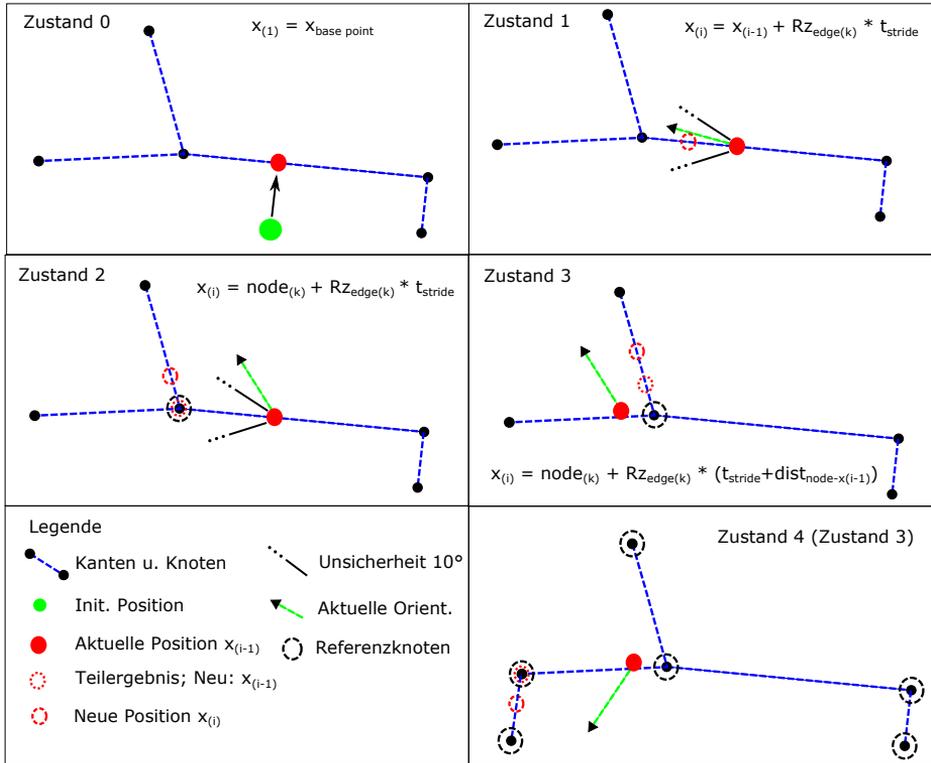


Abb. 5.16: Grafisch visualisierte Positionsberechnung der jeweiligen Zustände des topologischen Ansatzes zur Positionsschätzung [Willemsen u. a., 2015b].

Aufgrund hoher Abweichungen der Schrittlänge bei langen oder sehr kurzer Kanten (≤ 2 Schritte) kann es dazu kommen, dass falsche Bezugsknoten für die Kantenzuordnung ausgewählt werden. Die eigentlich passende Kante hat dann keinen Anschluss zum aktuell ausgewählten Knoten. Im *Zustand 4* findet daher eine globale Suche nach einer passenden Routing-Kante mit allen Knoten des jeweiligen Stockwerks statt. Die Auswahl der neuen Kante ist abhängig von der

Orientierung zum Gyroskop und von dem orthogonalen Abstand. Die Berechnung der neuen Position erfolgt anschließend auf Basis des zuvor erkannten Zustands.

Dieser Ansatz nutzt die Orientierung aus den Drehraten für jeden Schritt als einen relativen Vergleich zu den Routing-Kanten. Der Einfluss der Gyroskop-Drift wird dadurch minimiert, da durch den kürzeren Beobachtungszeitraum länger zurückliegende Abweichungen in der Orientierung nicht abgebildet werden. Zur Realisierung muss aber eine Richtungskorrektur der aus Drehraten berechneten Drehwinkel in Abhängigkeit von den Routing-Kanten Verwendung finden. Diese erfolgt in Anlehnung der Richtungskorrektur im KF-Ansatz und liefert hier die aktuelle Richtung der Routing-Kante als neuen Startwert, sofern die Routing-Kante mindestens dreimal hintereinander der Positionsschätzung zugeordnet wird.

5.3.2 Ergebnisdiskussion zweier repräsentativer Routen

Zur Untersuchung und zur Vergleichbarkeit der zuvor beschriebenen Ansätze auf Basis von KF und PF werden die repräsentativen Testdaten (OG4: 8 min 400 m; OG4-EG-OG4: 10 min 600 m) erneut verwendet.

Datensatz 1 für 2D

Abb. 5.17 zeigt die berechnete Trajektorie im vierten Obergeschoss des HCU-Gebäudes auf Basis des kantenbasierten PDR. Die Startposition ist mit einem blauen Kreis markiert. Der Endpunkt schließt mit einer Abweichung von 2 m auf den Referenz-Endpunkt ab. Die Trajektorie ist rot und die verwendeten Routing-Kanten sind in gestricheltem Cyan dargestellt. Die berechnete Trajektorie entspricht im Rahmen der Genauigkeitsanforderungen der Referenzstrecke.

Damit allerdings der kantenbasierte PDR-Ansatz für diesen Datensatz funktionieren kann, mussten zuvor die offenen Bereiche um Kanten und Knoten erweitert werden. Die modifizierten offenen Bereiche befinden sich im linken und rechten oberen Bereich des Grundrisses. Ein Routing-Graph ist hier nicht notwendig, da die Navigation zu einem offenen Bereich zwar sinnvoll sein kann, allerdings geschlossene Polygone in einem Raum nicht für einen Routing-Graphen angedacht sind, da dies einen zusätzlichen Rechenaufwand für das Routing bedeuten würde.

In Abb. 5.17 ist zu erkennen, dass der kantenbasierte Ansatz eine raumgenaue

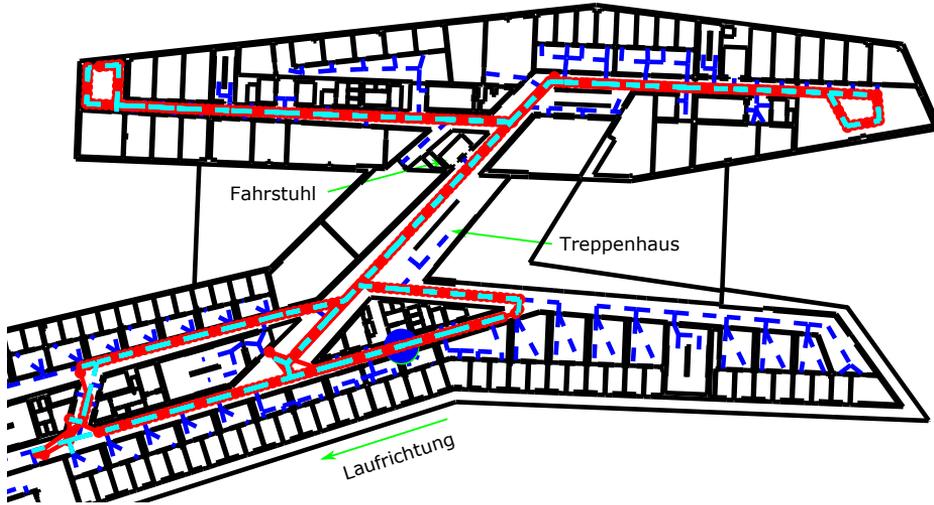


Abb. 5.17: Trajektorie des ersten Datensatzes auf Basis des topologische Ansatzes im vierten Obergeschoss des HCU-Gebäudes (rot: Trajektorie; cyan: verwendete Routing-Kanten) [Willemsen u. a., 2015b].

Positionsschätzung einhält. Durch regelmäßige Neu-Initialisierungen der aus Gyroskopdaten abgeleiteten Orientierung ist eine langfristige Navigation ohne jeden weiteren Einsatz von Korrekturen möglich. Abb. 5.18 zeigt den Vergleich zwischen den für die Positionsschätzung ausgewählten Kantenrichtungen (rot) und der Orientierung aus Drehraten des Gyroskops (schwarz). Eine Korrektur der Orientierung erfolgt nur bei mehrfacher Auswahl einer Bezugskante. Dies soll die Nutzung kurzer Kanten zur Korrektur minimieren, die besonders in Bereichen vorkommen, bei denen Richtungsänderungen wahrscheinlich sind. Der verwendete Winkelbereich beträgt $-\pi$ bis $+\pi$.

Größere Positionsabweichungen können in diesem Ansatz durch die Aktivierung von *Zustand 4* erkannt werden. Abb. 5.19 zeigt einen Ausschnitt der Trajektorie, bei der der *Zustand 4* aktiv wurde. Die für die Auswahl neuer Routing-Kanten eigentlich erforderlichen Übergangsknoten sind zu weit von der aktuellen Position entfernt. Daher wird ein anderer Knoten zur Kantensuche verwendet. An diesem Knoten lässt sich an keine passende Kante anknüpfen, da die Richtungsunterschiede zu den jeweiligen Kanten zu groß sind. Nun wird der *Zustand 4* aktiv, damit im gesamten Stockwerk nach einer passenden Kante gesucht werden kann.

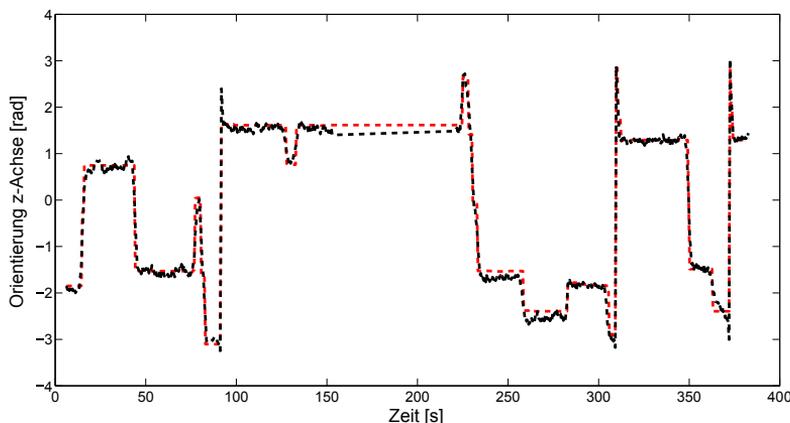


Abb. 5.18: Vergleich der Orientierung zwischen Gyroskopdaten (schwarz) und ausgewählten Routing-Kanten (rot) des ersten Datensatzes im vierten Obergeschoss [Willemsen u. a., 2015b].

Die Kante mit der geringsten Richtungsdifferenz zum aktuellen Drehwinkel und dem geringsten orthogonalen Abstand zur aktuellen Position wird für die weitere Positionsschätzung ausgewählt. An dem Beispiel in Abb. 5.19 finden mehrmals hintereinander fehlerhafte Zuordnungen statt, da der Nutzer eine Drehung vollzieht. Da im folgenden Berechnungsschritt erneut keine passende Kante gefunden wird, tritt *Zustand 4* wieder in Kraft. Abschließend wird die passende Routing-Kante ausgewählt und die Trajektorie aufgrund der Distanzkorrekturen im PDR an nahezu richtiger Stelle fortgeführt.

Datensatz 2 für 2D+1D

Für den Stockwerksübergang wird die Treppenerkennung, wie im KF-Ansatz beschrieben, in diesen Ansatz eingebunden, damit die Schrittlänge bei Treppen angepasst werden kann. Bei jedem detektierten Stockwerksübergang wird der Initialisierungsschritt ausgeführt und eine passende Kante im neuen Stockwerk gefunden. Von dieser Kante startet das kantenbasierte PDR nach dem in Abb. 5.16 beschriebenen Muster erneut. Im Unterschied zum Datensatz 1 sollen in dieser Auswertung die Auswirkungen einer Nutzerabweichung vom dem durch den Routing-Graph repräsentierten Weg deutlich werden.

Die Ergebnisse für die räumliche Trajektorie zeigen exemplarisch Abb. 5.20, Abb.

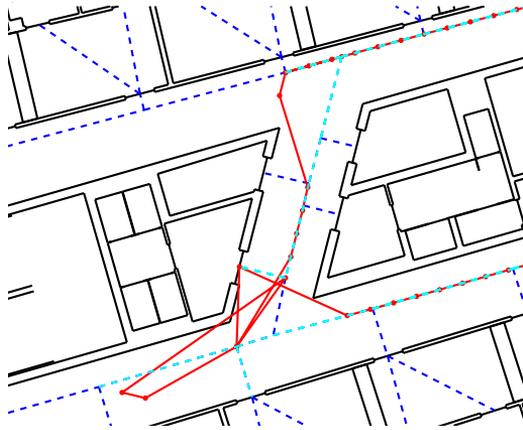


Abb. 5.19: Detail der Trajektorie aus Abb. 5.17 berechnet nach dem topologischem Ansatz: Hier Zustand 4 aktiv, globale Suche nach neuer Bezugskante [Willemsen u. a., 2015b].

5.21 und Abb. 5.22 für das vierte und zweite Obergeschoss sowie für das Erdgeschoss. Start- und Endpunkt der Referenzstrecke sind identisch (grüner Punkt) und befinden sich im vierten Obergeschoss.

Die Trajektorie im vierten Obergeschoss zeigt bis zum Stockwerksübergang die gleiche Positionsgenauigkeit – wie schon die bisherigen Ansätze auf Basis von KF und PF. Allerdings wird die Treppe nicht beschriftet, da der Routing-Graph den direkten Weg nicht repräsentiert und die Personenbewegung auf bestehende Kanten abgebildet werden. Die zurückkommende Trajektorie aus dem Fahrstuhl lässt sich nur durch den Einsatz von *Zustand 4* auf den richtigen Weg zum Ausgangspunkt hin berechnen.

Auf dem Hinweg erfolgt der Übergang zwischen den Stockwerken durch das Haupttreppenhaus. Gegenüber den vorherigen Ansätzen mit KF und PF zeigt sich im zweiten Obergeschoss der störende Einfluss der Kanten-Knoten-Struktur besonders auffällig. Die Berechnungen liefern zwar noch weitgehend Positionen in den richtigen Bereichen, allerdings funktioniert die Kantenzuordnung nur bedingt und der *Zustand 4* tritt vergleichsweise oft in Aktion (Abb. 5.21). Für einen verbesserten Einsatz der aktuellen kantenbasierten Positionsschätzung muss also der Routing-Graph den typischen Laufwegen besser angepasst werden.

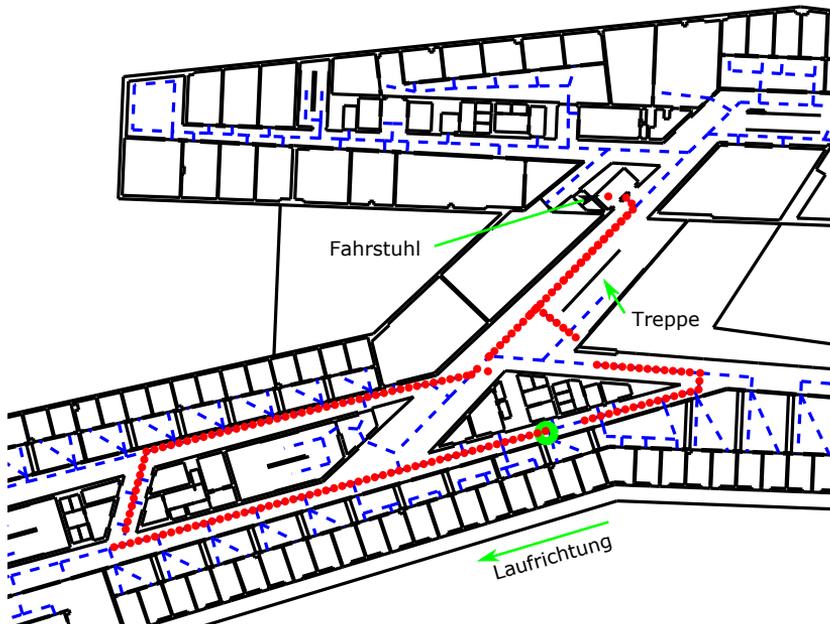


Abb. 5.20: Trajektorie (rot) des zweiten räumlichen Testdatensatzes, Positionsschätzung basierend auf den topologischem Ansatz im vierten Obergeschoss.

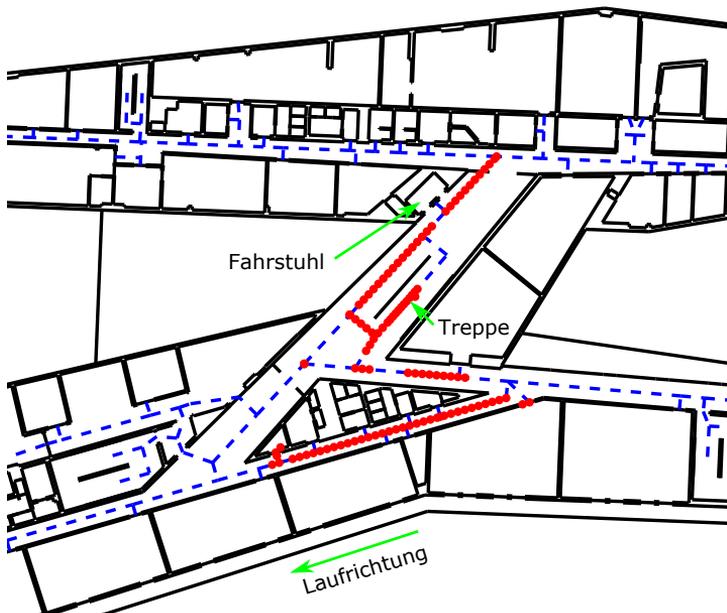


Abb. 5.21: Trajektorie (rot) des zweiten räumlichen Testdatensatzes für das zweite Obergeschoss.

Der große offene Bereich im Erdgeschoss des HCU-Gebäudes enthält wenige angrenzende Räume und daher nur wenige Routing-Kanten. Allerdings erlauben große, offene Bereiche dem Nutzer ein hohes Maß an Bewegungsfreiheit. Die negativen Einflüsse des Ansatzes werden daher hier besonders deutlich, sobald der Nutzer vom vorgegebenen Weg (Routing-Kanten) abweicht. Der Datensatz enthält genau in diesen Bereichen eine Trajektorie, die kaum mit dem Routing-Graph übereinstimmt. Daher ist in Abb. 5.22 die rot gefärbte Trajektorie stark durch die wenigen Routing-Kanten beeinflusst. Der Positionsfehler vor dem Stockwerksübergang in dem jeweiligen Stockwerk wird durch die Positionskorrektur beim Übergang in das nächste Stockwerk aber nicht übertragen, da in den aktuellen Daten der richtige Ort des Stockwerkübergangs, Fahrstuhl oder Treppe, zugeordnet werden kann.

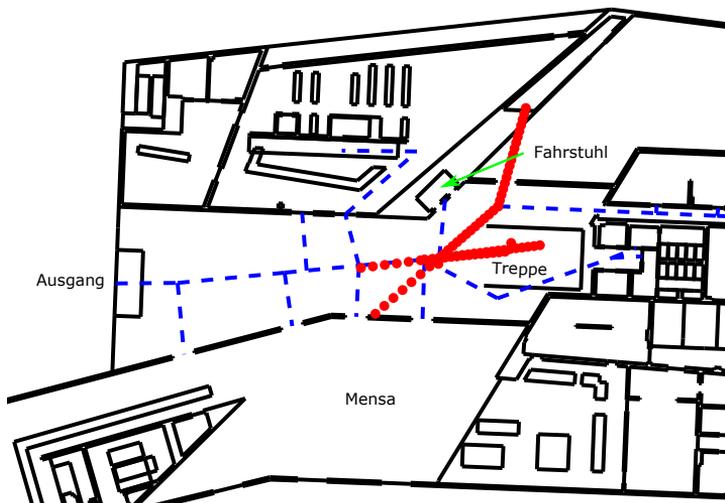


Abb. 5.22: Trajektorie (rot) des zweiten räumlichen Testdatensatzes für das Erdgeschoss.

Diese Ergebnisse zeigen, dass eine Navigation mit dem vorgestellten Ansatz nur unter der Bedingung einer einfachen Start-Ziel-Suche anwendbar ist und kein freies Bewegen im Gebäude unterstützt. Die Funktionalität des Ansatzes hängt stark von der Netzinfrastruktur des Routings ab. Von Vorteil ist allerdings die relative Positionsbestimmung, die in Bezug zur vorherigen Kante stattfindet. Dadurch haben die charakteristisch hohen Unsicherheiten, die im PDR in kurzen Zeitabständen produziert werden, weniger Einfluss auf die Positionsschätzung. Im Vergleich

zum KF und PF ist die Positionsschätzung relativ einfach umsetzbar und weniger rechenintensiv.

5.4 Vergleich und Diskussion der vorgestellten Ansätze

Mit dem Kalman Filter, dem Partikelfilter und dem topologischen Ansatz wurden drei unterschiedliche Methoden zur Positionsschätzung in GNSS-abgeschatteten Bereichen vorgestellt. Jeder Ansatz hat seine Stärken und Schwächen. Im Folgenden werden die Unsicherheiten anhand eines Koordinatenvergleichs vorgestellt. Besondere Schwerpunkte in diesem Abschnitt sind die Korrektoreinflüsse durch Karten und Routing-Graph auf die jeweiligen autonomen Positionsschätzer. Die verwendeten Koordinaten in diesem Vergleich basieren auf die in dieser Arbeit präsentierten Ergebnisse. Diese sind durch die Variation der im jeweiligen Filter möglichen Stellgrößen iterativ ermittelt und stellen die optimale Konfiguration in Abhängigkeit der verwendeten Testdaten dar. Zum Abschluss wird eine mögliche Erweiterung der Positionsschätzung durch einen Einsatz einer infrastrukturbasiereten Methode diskutiert.

Vergleich der beschriebenen Positionsschätzverfahren

Die Anwendung der verwendeten zwei Testdatensätze auf die Positionsschätzverfahren erlaubt einen direkten Vergleich. Der erste Datensatz im vierten Obergeschoss, der das Verfahren ohne Stockwerkswechsel wiedergibt, belegt für alle entwickelten Algorithmen in den Trajektorien vergleichbar gute Ergebnisse. Ein Koordinatenvergleich zu den Referenzpunkten aller untersuchten Ansätze zeigt Tab. 5.1. Aufgelistet sind die Differenzen fünf ausgewählter Positionen auf den Trajektorien zu den Referenzpunkten. Die Referenzpunkte wurden örtlich durch Markierungen auf dem Fußboden repräsentiert und in der Aufzeichnung der Trajektorie durch einen Zeitstempel beim Überschreiten gekennzeichnet. Die zu diesem Zeitpunkt aus der Positionsschätzung erhaltene Koordinate wurde für den Koordinatenvergleich herangezogen.

Die Positionsunsicherheiten im PDR steigen aufgrund der MEMS-Sensoren mit fortlaufender Positionsschätzung an. Daher beschreiben die fünf Referenzpunkte unterschiedliche Positionen über den gesamten gelaufenen Weg. Die Ergebnisse

Tab. 5.1: Differenzen zu Referenzpositionen einzelner Positionen auf den Trajektorien berechnet mit KF, PF und topologischem Ansatz des ersten Testdatensatzes im vierten Obergeschoss, ausgewählt in Abhängigkeit zur Länge der Laufstrecke [Willemsen u. a., 2015b]

ID	KF [m]	PF [m]	Kantenbasiertes PDR [m]
1 (80 m)	0,7	1,8	1,3
2 (120 m)	1,5	1,4	0,5
3 (220 m)	3,1	3,7	3,4
4 (330 m)	0,8	1,7	2,9
5 (400 m)	1,3	4,9	0,2

belegen maximale Abweichungen von 4,9 m zu den Referenzpunkten. Für den Datensatz 1 kann daher insgesamt die angestrebte Positionsgenauigkeit von 1-5 m erreicht werden.

Allerdings konnte anhand der Abbildungen der Trajektorien aufgezeigt werden, dass das primäre Ziel der präzisen Raumzuordnung nicht überall gewährleistet ist. Nur für den kantenbasierten PDR-Ansatz konnte eine Raumzuordnung sicher gestellt werden. Da sich der kantenbasierte PDR-Ansatz aber nur auf dem Routing-Graphen bewegen kann, sind Unsicherheiten durch das generalisierte Kanten-Knoten-Modell prinzipiell möglich. Zudem implizieren baulich offene Bereiche hohe Unsicherheiten, sobald der Nutzer von den definierten Zielwegen abweicht.

Für einen detaillierten Vergleich der Abweichungen zur tatsächlich gelaufenen Route sowie als Basis zur Diskussion der Arbeitsweise zeigt die Abb. 5.23 die Abweichungen der drei beschriebenen Algorithmen zur Referenz in jedem Schritt. Hierfür wurde die Referenztrajektorie als offenes Polygon konstruiert und für jeden Schritt in der Positionsschätzung der orthogonale Abstand zur Trajektorie berechnet. Die Differenzen repräsentieren daher den kürzesten Abstand der Positionsschätzung zur real gelaufenen Strecke.

KF und PF zeigen gegenüber dem kantenbasiertem PDR in den Abweichungen ein aufgelagertes minimales Rauschen, das die Arbeitsweise mit verrauschten Daten erkennen lässt. Weitgehend werden hier die Differenzen von kleiner 5 m eingehal-

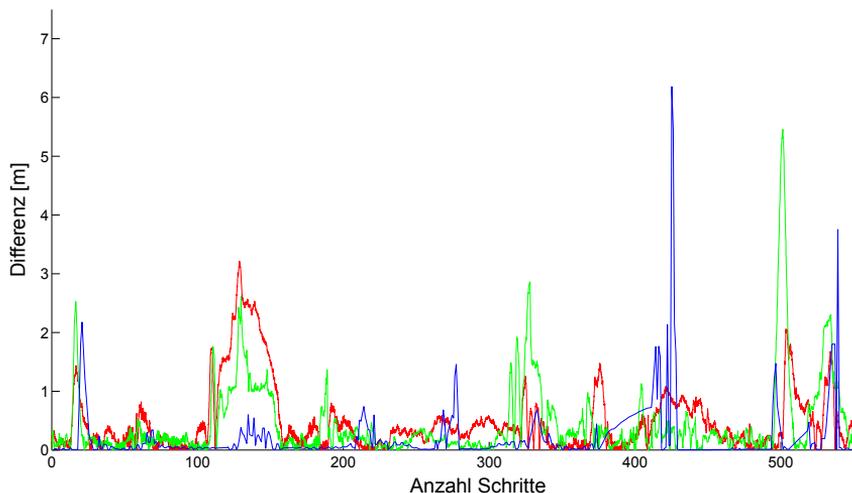


Abb. 5.23: Differenzen zur Referenztrajektorie für alle drei verwendeten Algorithmen in jedem Schritt (rot: Kalman Filter; grün: Partikelfilter; blau: kantenbasiertes PDR).

ten. Allerdings gibt es Ausnahmen im kantenbasierten PDR. An diesen Positionen wurde der Zustand 4 aktiv und erlaubte der Trajektorie, den richtigen Weg wieder aufzunehmen. Beim PF gibt es bei ca. 500 Schritten eine erhöhte Abweichung, die allerdings wieder durch eine Korrektur mit Routing-Kanten reduziert wird. Bei genauerer Betrachtung zeigt sich schon bei ca. 15 Schritten eine ansteigende Abweichung zur Referenz. An dieser Position fand die erste Richtungsänderung statt und die Algorithmen werden durch den Routing-Graph zur Mitte des Weges hin korrigiert. Der Nutzer hat die Kurve allerdings nicht mittig abgelaufen. Auf der anschließenden geradlinigen Route sinken die Abweichungen wieder. Ab 110 Schritten und ab 320 Schritten beginnen die beiden offenen Bereiche, die in der Trajektorie abgelaufen wurden. Hier ist des kantenbasierte PDR nur aufgrund der zusätzlichen manuell gesetzten Routing-Kanten berechenbar und die vergleichsweise guten Ergebnisse sind nicht repräsentativ.

Das vierte Obergeschoss ist geprägt von langen schmalen Gängen, wodurch alle MEMS-INS-basierte Anwendungen gute Ergebnisse liefern. Beim zweiten Da-

tensatz ist die Stockwerksunterscheidung erforderlich und die Stockwerke weisen zudem offene Bereiche auf.

Die Stockwerksunterscheidung liefert für die Positionsschätzung ein Koordinaten-Update (CUPT). Dazu muss allerdings der richtige Stockwerksübergang wie das Treppenhaus oder der Fahrstuhl zugeordnet werden. Dies erfolgt über einen Koordinatenvergleich und benötigt daher eine entsprechende Genauigkeit der Position vor dem Stockwerksübergang. Tab 5.2 zeigt die Koordinatendifferenzen bei einer Stockwerks-erkennung für alle drei Algorithmen zu allen Möglichkeiten des Stockwerkswechsels im HCU-Gebäude auf.

Für jeden Algorithmus kann trotz der zum Teil hohen Positionsunsicherheiten das richtige Treppenhaus oder der Fahrstuhl zugeordnet werden. Fahrstuhl und Treppe werden, wie schon beschrieben, durch die Schritterkennung voneinander unterschieden. Daher sind in der Tabelle nur die fettgedruckten Stockwerksübergänge möglich. Der Übergang am Haupttreppenhaus vom OG4 in das OG3 ist deutlich von den anderen Treppenhäusern unterscheidbar. Dies gilt auch für den Übergang zwischen OG2 nach OG1, da es hier keine Treppe zwischen EG und OG1 gibt. Einzig die Verwendung des Fahrstuhls für den Übergang vom EG zurück in das OG4 lässt eine Fehlzuordnung zu, da zwei Fahrstühle direkt nebeneinander liegen. Da der Abstand beider Fahrstühle geringer als 2 Meter ist, wird ein Fahrstuhl für das CUPT frei definiert und in den Testdaten der *Fahrstuhl-links* verwendet. Eine mittlere Koordinate für beide Fahrstühle ist prinzipiell möglich, macht aber eine Anpassung des PF notwendig, da dieser mit einer Wandkorrektur in der Positionsschätzung arbeitet.

Die drei vorgestellten Verfahren führen besonders im Erdgeschoss aufgrund der offenen Bereiche zu hohen Positionsunsicherheiten, die zum Teil auf fehlerhaften Korrekturen des Routing-Graphs zurückzuführen sind. Die Abweichungen übersteigen die angestrebte Positionsgenauigkeit. Beim KF-Ansatz ergibt sich im Erdgeschoss ein Maximalfehler von 10 m, beim PF-Ansatz von 10 m und beim kantenbasierten Ansatz von 25 m. Das zeigt, dass eine einheitliche Verarbeitung von Grundriss und Routing-Graph zur Korrektur nicht sinnvoll ist, da sie je nach Infrastruktur eine wesentliche Verbesserung der Positionsgenauigkeit, aber zugleich eine Verschlechterung der Positionsgenauigkeit bedeutet.

Tab. 5.2: Koordinatendifferenzen vor dem Stockwerksübergang zu den Korrekturpositionen für KF, PF und kantenbasierten PDR. Mögliche Orte des Stockwerksübergangs bzw. für ein CUPT sind fettgedruckt. Sie ergeben sich aus der Treppenerkennung und Luftdruckveränderungen. Fettgedruckte Differenzen repräsentieren die geringste Koordinatendifferenz und damit den ausgewählten Stockwerksübergang.

	KF	PF	Topo.
OG4 nach OG3 [m]			
Treppe EG-OG1	6,7	3,5	6,9
Haupttreppen OG1-OG4	8,6	11,4	7,7
Nebentreppen	32,4-41,8	28,6-46,1	32,1-40,9
Fahrstuhl links	2,6	7,6	2,6
Fahrstuhl rechts	2,7	6,6	3,4
Lastenfahrstuhl	35,5	30,7	36,4
OG2 nach OG1 [m]			
Treppe EG-OG1	6,9	6,2	8,2
Haupttreppen OG1-OG4	8,8	11,9	14,9
Nebentreppen	32,1-41,8	29,5-45,6	27,2-48,4
Fahrstuhl links	2,4	5,5	7,8
Fahrstuhl rechts	2,5	4,0	5,9
Lastenfahrstuhl	35,6	31,9	29,6
EG nach OG4 [m]			
Treppe EG-OG1	7,9	10,2	13,6
Haupttreppen OG1-OG4	8,1	13,9	20,9
Nebentreppen	31,1-40,6	30,9-45,6	23,1-53,9
Fahrstuhl links	1,6	5,0	13,2
Fahrstuhl rechts	2,8	3,1	11,2
Lastenfahrstuhl	37,0	33,5	26,0

Diskussion über den Einsatz des Routing-Graphen zur Positionskorrektur

Während also der Routing-Graph in schmalen Gängen eine gute Alternative zur Korrektur der geschätzten Position abgibt, erzeugt die Stützung durch Routing-Kanten in offenen Räumen zusätzliche Abweichungen. Im KF und PF kann die Korrekturintensität durch das Messrauschen gesteuert werden. Dies erlaubt eine variable Einstellung für verschiedene Gebäude- bzw. Raumtypen, wenn, wie im vorliegenden Fall, die Stützung aus Routing-Kanten im Erdgeschoss zu größeren Unsicherheiten führen kann, während in den schmalen Gängen des vierten Obergeschosses die Korrekturen eine deutliche Minimierung der Positionsunsicherheiten bedeuten können. Der Einfluss der Routing-Kanten in offenen Bereichen würde dadurch minimiert. Allerdings finden bei beiden Verfahren fehlerhafte Korrekturen in Bereichen von Kurven statt, da die Kanten meist nicht den gelaufenen Weg in Kurven repräsentieren. Grundsätzlich erweisen sich die per Routing-Graph gestützten KF und PF gegenüber dem kantenbasierten Verfahren als besser steuerbar und lassen sich der real gelaufenen Trajektorie eher anpassen. Dem steht die raumgenaue Zuordnung des kantenbasierten PDR gegenüber, das robuster auf Nutzerverhalten reagieren kann, sofern die Routing-Kanten den gelaufenen Weg repräsentieren. Für solch einen Einsatz zur Positionsschätzung auf Basis des kantenbasierten PDR muss das Routing-Netz allerdings für jedes Gebäude manuell angepasst werden oder für die typischen Laufwege neu erstellt werden.

Diskussion über den Einsatz von Karten zur Positionskorrektur

Die Kartenstützung kann zur Korrektur der Orientierung dienen und die Umsetzung der raumgenauen Positionsschätzung sicherstellen. Der Ansatz auf Grundlage des PF enthält die Korrektur durch Wände, allerdings ist der Durchbruch durch die Wand möglich, damit die Positionsschätzung trotz höherer Unsicherheiten fortgeführt werden kann. Die Wand-Geometrien lassen sich im PF vergleichsweise gut zur Stützung einbinden. Erforderlich sind dazu nur einfache Vektoroperationen: Mit Geradenschnitt und Richtungsdivergenzen lassen sich die Schnittpunkte mit den umliegenden Wänden berechnen und die Partikelgewichte in Abhängigkeit von Abständen und Richtungsdivergenzen leicht anpassen.

Bei dem KF-Ansatz wurde auf die Positionskorrektur auf Basis von Grundriss-

daten verzichtet. Die Korrektur wie beim PF ist zwar prinzipiell umsetzbar, aber es gibt nur eine Zielposition, die verbessert werden kann. Würde beispielsweise bei der Zielbestimmung eine gegenüberliegende Wand durchbrochen, könnte der Schnittpunkt mit der Wand als neue Position angenommen werden. Für die folgende Schätzung würde wiederum ein Wanddurchbruch erforderlich. Daher eignet sich der Grundriss in dem hier realisierten KF-Ansatz nur zur Richtungskorrektur, die aber durch den Einsatz des Routing-Graphen weitgehend Berücksichtigung findet. Daher wurde auf eine Implementierung im KF verzichtet.

Abb. 5.24 skizziert zwei Szenarien, die die Probleme beim Einsatz von Grundrissdaten zur Positionskorrektur weiter verdeutlichen. Dargestellt sind insgesamt drei Zustände, die mit oder ohne einen Wanddurchbruch auskommen, damit die Trajektorie ihren Weg wieder korrekt aufnimmt oder fortsetzt.

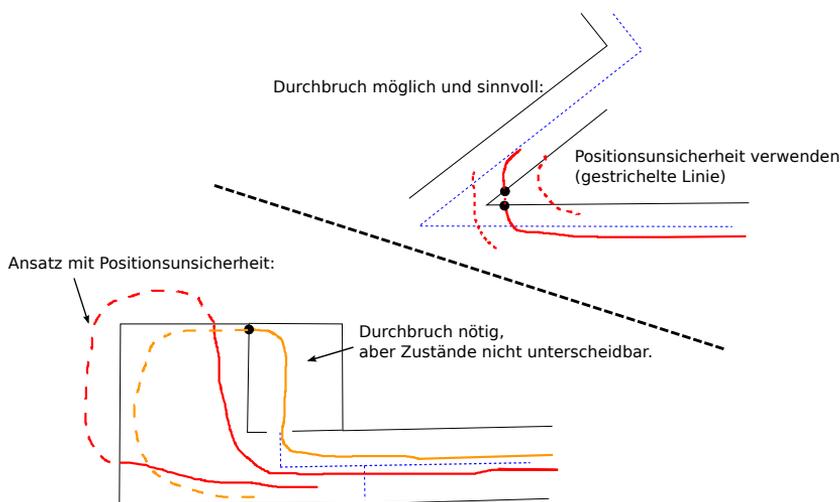


Abb. 5.24: Auswirkungen der Positionskorrektur basierend auf Wandabfragen (oben: Anwendung eines Unsicherheitsbandes, um Ecken überqueren zu können; unten: zwei Zustände, rot: Stauchung durch Wandstützung, gelb: Abbruch der Berechnung, wenn der Filter den Durchbruch nicht ermöglicht).

Das erste Beispiel im rechten oberen Bereich zeigt ein Durchqueren der Trajektorie durch eine Wand. An dieser Stelle ist ein Durchbruch sinnvoll. Denn wegen Unsicherheiten bei der Positionsschätzung verläuft die Trajektorie in der Kurve durch die Wand und würde hinter der Wand auf dem richtigen Weg weiterführen. Solche Positionsfehler können vorkommen, da fehlerhafte Schrittlängen zur Be-

rechnung zu kurzer oder zu langer Trajektorien führen. Im vorliegenden Szenario könnte die geschätzte Positionsgenauigkeit verwendet werden, um zu prüfen, ob die Wand einen Durchgang oder eine Tür aufweist. Die gestrichelten Linien in der Abbildungen stehen für eine solche Überprüfung.

Das zweite und das dritte Beispiel im linken unteren Bereich der Abb. 5.24 belegen anhand von zwei weiteren Situationen, dass dieser Ansatz bei der gegebenen Gebäudeinfrastruktur nicht immer sinnvoll ist. In Gebäuden mit vielen Räumen kann eine ungenaue Schrittlänge bei längeren geradlinigen Strecken eine zu frühe oder zu späte Richtungsänderung der Navigation zur Folge haben. Wie am Beispiel der gelben Trajektorie dargestellt, werden irrtümlich Räume angesteuert oder es werden wie bei der roten Trajektorie Raumgrenzen durchlaufen. Nach Abfrage des Unsicherheitsbandes würde die rote Trajektorie gestaucht und im richtigen Raum weiter verlaufen. Für die rote Trajektorie würde dies einen positiven Einfluss auf die Positionsschätzung bedeuten. Erfolgt für den gelben Datensatz nicht der Durchbruch durch die Wand, divergieren die vorgestellten Ansätze auf Basis von KF und PF.

Außer der Anpassung der Korrekturintensität im Filter lässt sich das Problem der Kartenstützung durch modifizierte Grundrisse weiter minimieren. Dazu können Türen im Grundriss abgebildet werden, die das Filter wie eine Wand behandelt und damit im Prinzip nur Flure für die Navigation zulässt.

Zusätzliche Stützung auf Basis infrastrukturbasierter Positionsschätzverfahren

Eine weitere Möglichkeit, die hohen Unsicherheiten der MEMS-INS-basierten Positionsschätzung zu korrigieren, bieten infrastrukturbasierte Positionsschätzer. Die Positionsschätzung entspricht dann allerdings einer hybriden Vorgehensweise.

Besonders gering ist der Implementierungsaufwand bei WLAN basierten Verfahren, da für diese üblicherweise ein Netzwerk existiert. In dieser Arbeit wurde das WLAN Fingerprinting am HCU-Gebäude umgesetzt, um auch die Chancen infrastrukturbasierter Methoden abschätzen zu können. Eine MEMS-INS-basierte Positionsschätzung kann durch ein Update der Koordinaten korrigiert werden. Allerdings führen die per WLAN Fingerprinting ermittelten Positionen in offe-

nen Bereichen zu hohe Unsicherheiten, sodass eine Korrektur in den Bereichen, in denen die hier vorgestellten Verfahren Schwächen aufweisen, nicht sinnvoll ist.

Die Kombination dieser beiden Verfahren wäre aber interessant – vor allem für den Initialisierungsschritt, damit sich QR-Codes oder Nutzereingaben erübrigen. Da die Erstellung der Fingerprints und die Unterhaltung aktueller Daten beim WLAN Fingerprinting einen hohen Aufwand mit sich bringen, stellt das Fingerprinting auch zur Initialisierung im Hinblick des Aufwands keine wirkliche Alternative dar.

Daher müssen andere infrastrukturbasierte Methoden gefunden werden, die mit relativ geringem Implementierungsaufwand auskommen. So hat eine hybride Positionsschätzung den Vorteil, dass die Infrastruktur nicht für jeden Bereich des Gebäudes vorliegen muss. Vorhandene Lücken können durch die MEMS-INS-basierte Positionsschätzung abgefangen werden. Dabei können die Stützinformationen Lage- und Raumstrecken sowie Richtungen und 2D Positionen sein, die eine Position oder Richtung im Filter korrigieren lassen. Ein Beispiel auf Basis eines künstlichen Magnetfeldes wird in Real Ehrlich u. Blankenbach [2014] beschrieben. Hier werden Lagestrecken auf Basis künstlicher Magnetfelder simuliert, um die Partikelgewichte in einem PF auf Basis eines PDR anzupassen.

Der Einsatz der infrastrukturbasiert erzeugten Messelemente erfolgt im KF- und PF-Ansatz nach den gleichen Prinzipien wie bei der Einbindung von Informationen abgeleitet aus dem Grundriss und dem Routing-Graph. Die Implementierungsaufwand zur Anpassung der Filter ist daher als gering anzunehmen. Für den kantenbasierten PDR-Ansatz bietet sich eine Implementierung auf Basis des *Zustand 4* an, der global nach passenden Routing-Kanten sucht. Diese globale Suche kann dann durch zusätzliche infrastrukturbasiert erzeugten Messelementen robuster erfolgen.

Fazit

Der Vorteil eines PF besteht in der Möglichkeit, beliebige Verteilungen mit den gestreuten Partikeln zu modellieren. Dadurch lassen sich Grundriss und Routing-Kanten vergleichsweise einfach zur Korrektur der Position einbinden. Allerdings ist eine eindeutige Identifikation eines Raumes so nicht sicherzustellen. Gegenüber dem PF kann das KF nur normalverteilte Zustände annehmen, die Einbindung topologischer Informationen gestaltet sich daher vergleichsweise schwierig. Das

kantenbasierte PDR auf dem Routing-Graphen ist wenig rechenintensiv und erlaubt eine klare Identifikation eines Raumes, da die Positionsschätzung nur über die Kanten erfolgen kann. Das kantenbasierte PDR ist weniger anfällig für fehlerhafte Schrittlängen, denn es erlaubt die Annahme einer nutzerunabhängigen mittleren Schrittlänge. Damit die Positionsschätzung in jeder Umgebung möglich ist, ist eine manuelle Modifikation des Routing-Graphen erforderlich. Um weitere Unsicherheiten auszuschließen, sind für diesen Ansatz lediglich die direkten Streckenverbindungen zwischen Start und Ziel abzulaufen.

Der Einsatz einer infrastrukturbasierten Methode zur Korrektur besonders in den offenen Bereichen erscheint sinnvoll, ist aber aufgrund des zusätzlichen Implementierungsaufwandes und des Wartungsaufwandes zu verwerfen. Um eine möglichst autonome Positionsschätzung zu gewährleisten, muss ein Algorithmus entwickelt werden, der dem Nutzer die freie Bewegung in jedem Raumtypus erlaubt, aber dennoch gegebene Karten- und Routinginformationen zur Sicherstellung der raumgenauen Positionsschätzung verwenden kann.

5.5 Finaler Fusionsalgorithmus - Kantenbasiertes Partikelfilter

Zur MEMS-INS-basierten Positionsschätzung ist aufgrund der jeweils auftretenden Schwächen keiner der drei Ansätze für sich genommen tauglich. Daher wird eine Fusion des Partikelfilters und des kantenbasierten PDR angestrebt, um die Schwächen des einen Ansatzes durch die Stärken des anderen zu kompensieren. Ziel ist es, die freie Positionsschätzung im Raum, die per PF möglich ist, mit einer erhöhten Raumgenauigkeit, wie sie der kantenbasierte Ansatz ermöglicht, zu kombinieren. Vorteile dieses Ansatzes wären der Wegfall der manuellen Modifikation des Routing-Graphen und eine geringere Abhängigkeit des Routing-Graphen in offenen Bereichen.

5.5.1 Fusion von Partikelfilter und topologischem Ansatz

Das kantenbasierte PDR funktioniert für die den Laufwegen angepassten Routing-Graphen vergleichsweise gut. In großräumigen Bereichen versagt dieser Ansatz meist aufgrund der Abweichungen der Nutzerbewegung vom Routing-Graphen.

Finden ein oder mehrere Partikel keine zuverlässig bestimmbare Kante, dann kann das Partikel vom Routing-Graph entkoppelt werden und als PDR PF propagiert werden, so wie in Kapitel 5.2 *Partikelfilter* beschrieben. Anders als bei den bisher erörterten PDR PF entfällt die Stützung auf Basis von Routing-Kanten.

Die Besonderheit dieses PF besteht in der Steuerung der Positionsschätzung allein durch Richtungsdivergenzen. Das für jeden Schritt neu berechnete Partikelgewicht w_i basiert nur auf der propagierten Richtung r_i . Das Gewicht wird auf Grundlage der WDF mit der Differenz zwischen der Richtung der ausgewählten Routing-Kante r_{Kante_j} und der aktuellen Partikelrichtung berechnet (74). Die Steuerung der Gewichts Berechnung wird mit dem Messrauschen für die Richtungsbestimmung R_r durchgeführt.

$$w_i = \exp[-0,5 * (r_i - r_{Kante_j}) * R_r^{-1} * (r_i - r_{Kante_j})] \quad (74)$$

Die Positionen x_i , y_i und z_i werden dem Partikelvektor angefügt. Sie verändern sich nur in Abhängigkeit der Partikelrichtung und der ermittelten Zustände. In Formel (75) wird der Partikelvektor \vec{p} angezeigt. Neben den beschriebenen Variablen wird die ID der ausgewählten Kante ID_{Kante} mitgeführt.

$$\vec{p} = \begin{pmatrix} r_i \\ w_i \\ x_i \\ y_i \\ z_i \\ ID_{Kante} \end{pmatrix} \quad (75)$$

In Widyawan u. a. [2008] wird eine Anwendung eines Backtracking Partikelfilters (BPF) anhand einer MEMS-basierten Positionsschätzung vorgestellt. Bei fehlerhaften Abschnitten der Trajektorien können durch eine Rückrechnung der Trajektorie neue Positionen geschätzt werden, um zum richtigen Weg zurückzukommen. Was Nutzer während der Navigation irritieren könnte, bringt für MEMS-INS-

basierte Anwendungen den Vorteil, dass sich der Weg ohne erneute Initialisierung nachträglich korrekt bestimmen lässt.

Im kantenbasierten PF soll dies mit Mehrfachlösungen durch eine Aufteilung der Partikel ermöglicht werden. Durch die Zuordnung der Partikel zu den Routing-Kanten ist eine Aufteilung zu mehreren Trajektorien möglich. Die maximale Anzahl möglicher Routen entspricht dann der Anzahl der Partikel. Per Resampling im PF werden die unwahrscheinlichen Routen automatisch reduziert, da den dort zusammengeführten Partikeln aufgrund der Richtungsdivergenzen weniger Gewicht zukommt. Dem Nutzer wird anschließend nur die wahrscheinlichste Position ausgegeben.

Für die Aufteilung der Partikel zu möglichen Routen müssen diese im Filter anhand der Routing-Kanten getrennt werden. Die entkoppelten Partikel, die auf Basis des PDR PF berechnet sind, bilden weitere mögliche Routen, lassen sich aber nicht über die Routing-Kanten zuordnen. Daher werden sie im Folgenden untereinander durch den räumlichen Abstand zueinander zu Punktgruppen zusammengefasst, die jeweils eine mögliche Position repräsentieren.

Dem Nutzer muss anschließend die wahrscheinlichste aller parallel geschätzten Routen für die Navigation ausgegeben werden. Für die Auswahl werden zwei Kriterien untersucht: erstens die Summe der Partikelgewichte einer Route und zweitens die Anzahl der Partikel für die jeweiligen Routen.

Das Konzept für den Ablauf des kantenbasierten PF präsentiert Abb. 5.26. Die Initialisierung erfolgt über die Eingabe einer Position und einer Startrichtung. Zusätzlich wird die Position durch die Fußpunktberechnung auf die erste favorisierte Kante übertragen und aus den Startwerten werden die Partikelrichtungen generiert.

Außerhalb der Schätzung werden die relativen Höhen zur Stockwerksunterscheidung errechnet. Die Schritterkennung, die nach dem bisherigen Ansatz stattfindet, ist Auslöser für die Berechnung im PF. Ein Stillstand muss daher nicht zusätzlich detektiert werden. Das Filter sucht für jedes Partikel die umliegenden Routing-Kanten, um mögliche kantenbasierte Positionen berechnen zu können. In Abhängigkeit von der Zustandserkennung gibt es vier Möglichkeiten zur Fortführung

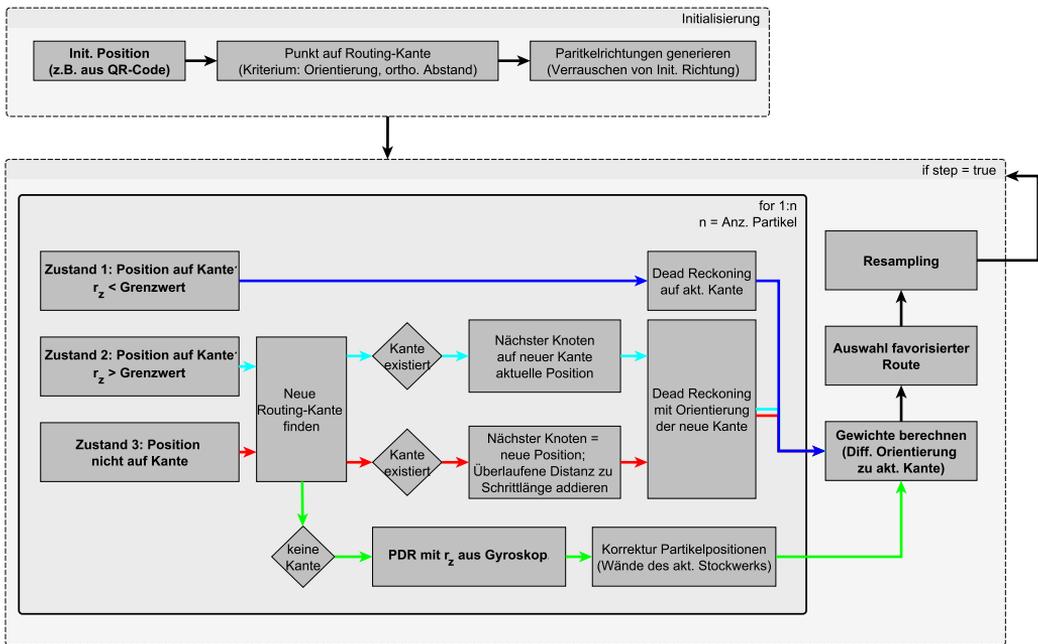


Abb. 5.26: Konzeptioneller Ablauf des kantenbasierten Richtungspartikelfilters.

der Trajektorie. Zustände 1 bis 3 sind aus dem kantenbasierten PDR abgeleitet, bei dem die Positionen auf den Routing-Kanten fixiert sind. Zustand 4 aus dem kantenbasierten PDR, der für die globale Suche nach passenden Kanten im Gebäude eingesetzt wird, wird hier nicht verwendet. Stattdessen wird ein PDR PF mit Berücksichtigung des Grundrisses in der Positionsschätzung eingesetzt, der aktiv wird, sobald sich in Abhängigkeit von einem Grenzwert – in den folgenden Ansätzen 10° – keine passende Routing-Kante für das ausgewählte Partikel definieren lässt. Der Grenzwert ist aus Testberechnungen abhängig von der Raumstruktur und den vorhandenen Routing-Graphen für jedes Gebäude abzuleiten. Mit Grenzwerten zwischen 5° und 15° wurden vergleichbar gute Ergebnisse im HCU-Gebäude erreicht.

Nach der Positionsschätzung in Abhängigkeit der zuvor definierten vier Möglichkeiten werden für alle Partikel die Gewichte durch die WDF bestimmt. Dies erfolgt aus der Differenz zu den Kantenrichtungen mit den Partikelrichtungen. Beim PDR PF geschieht dies aus der Differenz zur aktuellen Richtung, die aus Gyro-

skopdaten abgeleitet wurden. Für die Bestimmung der möglichen Routen werden die Partikel den Routing-Kanten zugeordnet bzw. die per PDR PF berechneten Positionen zu Punktgruppen in Abhängigkeit eines definierten Abstandes zueinander zusammengefügt. Die Zuordnung der Partikelpositionen basiert auf einer Mittelwertberechnung, bei der nacheinander neue Partikelpositionen der Mittelbildung hinzugefügt werden. Überschreitet die Standardabweichung des Mittels den Grenzwert, wird die hinzugefügte Position einer anderen Punktgruppe zugeordnet.

Entsprechend der Partikelanzahl pro Routing-Kante bzw. Punktgruppe oder der Summe der Gewichte der jeweiligen Routen kann die optimale Route ausgewählt werden. Zum Abschluss findet ein Resampling statt, bei dem die Partikel mit hohen Gewichten auf Kosten von niedrig gewichteten Partikeln reproduziert werden. Sobald ein neuer Schritt erkannt wird, wiederholt sich der beschriebene Ablauf.

5.5.2 Ergebnisdiskussion zweier repräsentativer Routen

Zum Vergleich der Ansätze auf Basis von KF, PF und kantenbasiertem PDR werden die repräsentativen Testdaten (OG4: 8 min 400 m; OG4-EG-OG4: 10 min 600 m) erneut verwendet. Die Karten und der Routing-Graph sind nicht, wie im kantenbasiertem PDR, editiert. Der Routing-Graph beinhaltet nur die für die Routenberechnung notwendigen Kanten und Knoten, wie sie zuvor für das Routing erstellt wurden.

Datensatz 1 für 2D

Die Ergebnisse der 2D-Trajektorie sind in Abb. 5.27 und in Abb. 5.28 aufgeführt. Der einzige Unterschied zwischen beiden Abbildungen besteht in der Auswahl der favorisierten Positionen aus der Partikelwolke. In Abb. 5.27 wird die favorisierte Trajektorie aus der Anzahl der den Kanten zugeordneten Partikel abgeleitet. Eine weitere, der Wahrscheinlichkeit besser angepasste Methode zur Trajektorienbestimmung besteht in der Verwendung der Summe der Partikelgewichte zu den jeweiligen Kanten. Die Ergebnisse zeigt Abb. 5.28. Die favorisierten Routen sind magenta bzw. blau dargestellt. Grün sind Partikel markiert, die allerdings zum Großteil durch die favorisierte Route verdeckt werden. Die rote Trajektorie re-

präsentiert das nicht korrigierte PDR. Der Partikelfilter wurde mit 100 Partikeln berechnet. Für den PDR PF wird die im PF-Ansatz verwendete Wandstützung implementiert, sodass ein Durchbrechen durch Wände erschwert wird.

Die Ergebnisse für den 2D-Datensatz dokumentieren im Vergleich zu den bisherigen drei Ansätzen zur Realisierung gute Testergebnisse. Die hauptsächliche Verbesserung besteht in der eindeutigen Raumidentifikation in den offenen Bereichen ohne Zuhilfenahme von Routing-Kanten. Da die Trajektorie hauptsächlich in Fluren mit Routing-Kanten verläuft, ist eine weitgehend relative Routing-Kantenbasierte Positionsberechnung möglich. In den offenen Bereichen fließen daher vergleichsweise weniger Unsicherheiten in die Positionsschätzung ein.

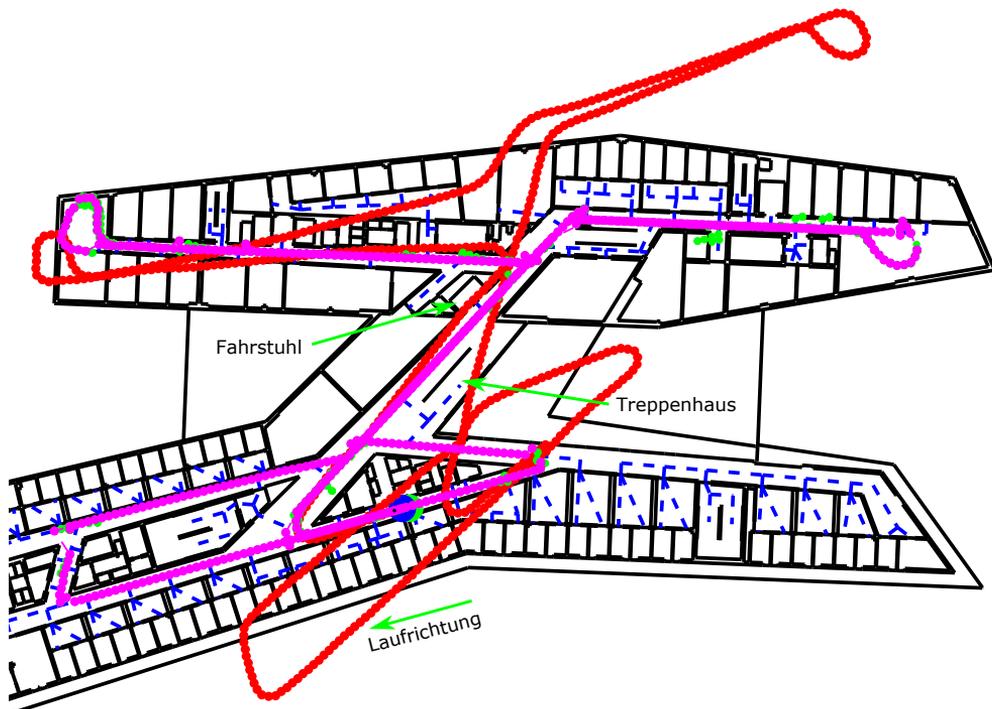


Abb. 5.27: Trajektorie im vierten Obergeschoss des HCU-Gebäudes basierend auf dem kantenbasierten Richtungspartikelfilter mit favorisierter Route, ausgewählt auf Basis der einer Punktgruppe zugeordneten Anzahl von Partikeln (magenta: favorisierte Route, grün: Partikel, rot: PDR).

Je nach Höhe des Grenzwerts für die erlaubte Richtungsdivergenz eines Partikels zur ausgewählten Kante findet das Koppeln bzw. Entkoppeln von dem Routing-

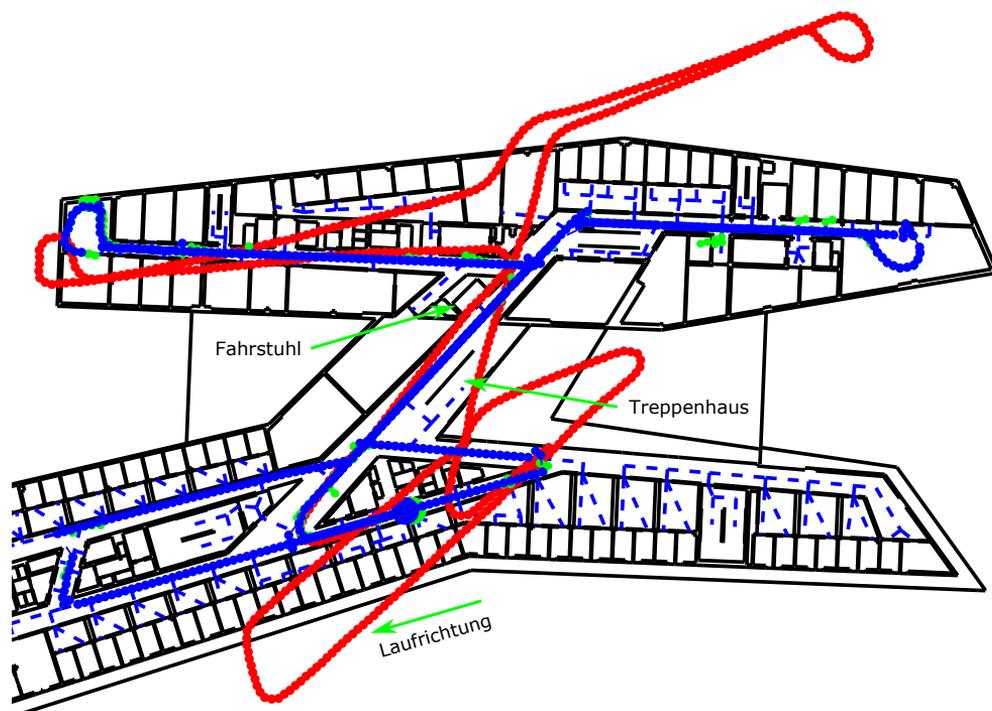


Abb. 5.28: Trajektorie im vierten Obergeschoss des HCU-Gebäudes basierend auf dem kantenbasierten Richtungspartikelfilter mit favorisierter Route, ausgewählt in Abhängigkeit der Gewichtssumme der einer Punktgruppe zugeordneten Partikel (blau: favorisierte Route, grün: Partikel, rot: PDR).

Graph früher statt und erlaubt auch in Kurven ein kurzzeitiges Entkoppeln von der Routing-Kante bei der Positionsschätzung. Die Unterschiede zwischen den favorisierten Routen, die dem Nutzer zur Navigation zur Verfügung gestellt werden, sind hauptsächlich in den offenen Bereichen deutlich, an denen der PDR PF aktiv ist.

Die Abb 5.29 zeigt Detailansichten aus Abb. 5.27, um die Arbeitsweise des kantenbasierten PF zu veranschaulichen. Links ist der Übergang der Partikel in einen offenen Bereich zu sehen. Es ist zu erkennen, dass sich der Hauptteil der Partikel von der Kante trennt und als PDR PF fortgeführt wird. Da es im PDR PF viele mögliche freie Routen geben kann, muss hier eine Auswahl zusammenhängender Partikel stattfinden. Sie erfolgt nach dem räumlichen Abstand der Partikel zueinander. Grenzwert dafür ist eine Standardabweichung S_0 der ausgewählten Punkte

von 0,5 Meter. Erzeugt ein zugefügter Punkt ein höheres S_0 , so wird dieser einer anderen Punktgruppe zugeordnet. Nach jedem Schritt werden die jeweiligen Punkthaufen neu bestimmt. Bei der Rückkehr der Trajektorie zu den Routing-Kanten erkennt man ein erneutes Ankoppeln der Trajektorie an die Routing-Kante.

Der Ausschnitt in der Mitte der Abb. 5.29 zeigt die Trajektorie, die vom rechten Gang kommend nach rechts in den größeren Gang verläuft. Je nach Festlegung der erlaubten Richtungsdivergenz zwischen Partikelrichtung und der zugehörigen Kantenrichtung erfolgen das Entkoppeln und das Ankoppeln der Partikel unterschiedlich schnell. Bei diesen Einstellungen berechnet der Ansatz die real gelaufene Strecke. Bei genauerer Betrachtung der Kanten fällt auf, dass die Partikel nach Ankoppeln der Kante stärker streuen als zuvor. Dies wird durch die Positionsschätzung im PDR PF hervorgerufen, da die Positionen im Raum direkt mit den verrauschten Partikelrichtungen verarbeitet werden.

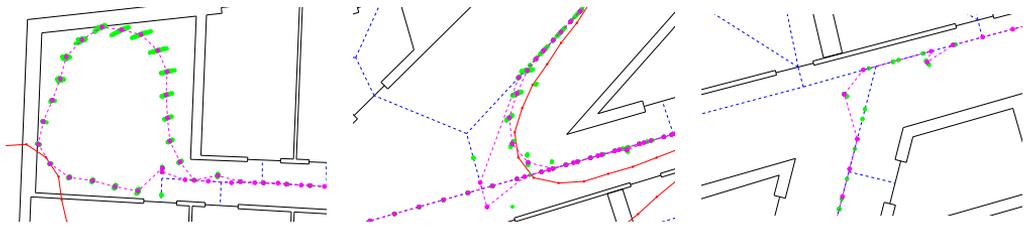


Abb. 5.29: Details einer Trajektorie des ersten Testdatensatzes im vierten Obergeschoss, berechnet nach dem kantenbasierten PF aus Abb. 5.27 (links: Übergang offener Bereich mit Wandstützung, Mitte: Entkoppeln von Kanten, rechts: mehrere Routen).

Die rechte Grafik in Abb. 5.27 zeigt den Richtungswechsel nach einem längeren geraden Streckenabschnitt. Die Trajektorie führt von rechts kommend links in den mittleren Gang. Die Partikel (grün) erlauben mehrere unterschiedliche Routen für die favorisierte Trajektorie (magenta). Lange, gerade Strecken minimieren den Vorteil des kantenbasierten Ansatzes, weniger anfällig auf Schrittlängen zu reagieren. Dies bedeutet für die kantenbasierten Positionsschätzungen weiterhin eine Abhängigkeit zur Schrittlängendefinition. In diesem Bereich der Trajektorie findet allerdings ein erneutes Ankoppeln an den Routing-Graphen statt.

Für einen Vergleich der Ausgangsalgorithmen, PF und kantenbasierten PDR mit der Fusion dem kantenbasierten PF wird der Koordinatenvergleich für jeden Schritt

berechnet, um die Auswirkungen der Arbeitsweise mit den jeweiligen einzelnen Algorithmen vergleichen zu können. In Abb 5.30 sind die orthogonalen Abstände zur Referenztrajektorie für den Datensatz 1 aufgeführt. Es lässt sich deutlich eine Verbesserung durch die Fusion erkennen, besonders in den offenen Bereichen ab Schritt 110 und 320. Hier sei noch einmal daran erinnert, dass für den kantenbasierten PDR zusätzliche Routing-Kanten generiert wurden, die beim kantenbasierten PF keine Anwendung fanden.

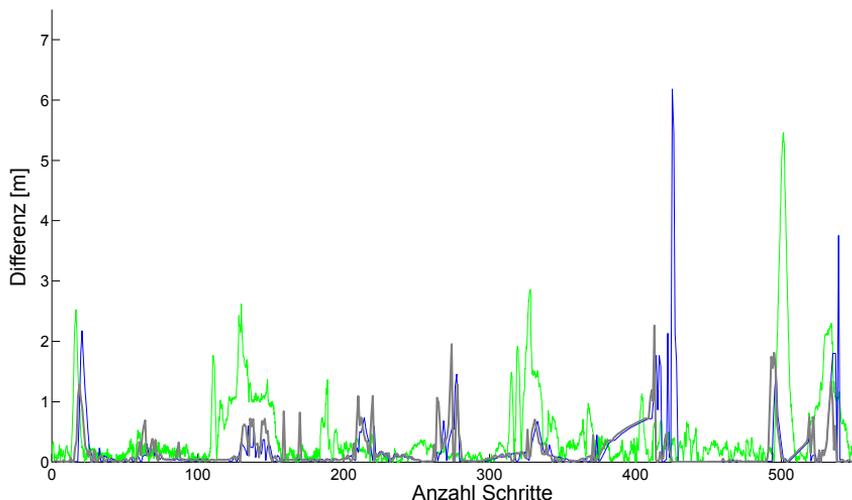


Abb. 5.30: Differenzen zur Referenztrajektorie für in jedem Schritt für den PF- und topologischem Ansatz sowie der Fusion, dem kantenbasierten PF. (grau: kantenbasierter PF ausgewählt nach Anz. der Routenpartikel; grün: Partikelfilter; blau: kantenbasiertes PDR).

An einigen Stellen wird die gleiche Arbeitsweise zwischen kantenbasierten PDR und kantenbasierten PF deutlich, da die Differenzen sich nahezu gleichen. Zwei Beispiele befinden sich bei 400 und 500 Schritten. Wie beim vorherigen Vergleich der drei Ausgangsalgorithmen ist auch hier zu Beginn, in der ersten Kurve bei ca. 15 Schritten, der Einfluss der Routing-Kanten erkennbar, der beim kantenbasierten PF allerdings geringer ausfällt. Es zeigt sich, dass die Unsicherheiten in der Positionsschätzung durch die Kombination der beiden Algorithmen im kantenbasierten PF signifikant minimiert werden. Allerdings repräsentiert das vierte

Obergeschoss mit den weitgehend schmalen Fluren und nur kleinen offenen Bereichen eine vergleichsweise einfache Infrastruktur für die Trajektorienschätzung.

Datensatz 2 für 2D+1D

Daher wird der kantenbasierte PF um die Stockwerks- und Treppenerkennung erweitert und in den weiteren Stockwerken mit dem zweiten Datensatzes untersucht. Wie auch bei den Ausgangsalgorithmen findet beim Stockwerksübergang ein CUPT statt, um nicht die Unsicherheiten in das neue Stockwerk zu übertragen und um die Orientierung neu zu initialisieren.

Die Ergebnisse des zweiten Datensatzes werden für das Erdgeschoss (Abb. 5.33), das zweite Obergeschoss (Abb. 5.32) und das vierte Obergeschoss (Abb. 5.31) dargestellt. Die favorisierten Routen auf Basis der Anzahl der Partikel (magenta) und der Summe der Partikelgewichte (blau) sind überlagert dargestellt.

Im vierten Obergeschoss unterscheiden sich die Ergebnisse von der real gelaufenen Strecke kaum. Bei genauerer Betrachtung sind vermehrt Partikel (grün) sichtbar, die sich von den Routing-Kanten absetzen und als PDR-PF berechnet werden. Dies zeigt, dass einige Partikelrichtungen den Grenzwert überschreiten und die Positionsschätzung als PDR PF fortführen. Gegenüber den bisherigen drei beschriebenen Ansätzen wird das Treppenhaus dadurch erreicht, dass ein Entkoppeln der Positionsschätzung aufgrund der erhöhten Richtungsdivergenz zu den Routing-Kanten stattfand.

Im zweiten Obergeschoss wird die Treppe mit einer 180°-Kehrtwende verlassen und ein kleiner Rundgang durch den Flur abgelaufen, bevor es über das Treppenhaus in das erste Obergeschoss geht. In Abb. 5.32 zeigt das PDR (rot) gegenüber der Referenztrajektorie starke Abweichungen, was zum Teil der fehlenden Korrektur beim Stockwerksübergang geschuldet ist. Auch beim kantenbasierten PF ist eine erhöhte Abweichung im Bereich 2 bis 3 m von der real gelaufenen Route erkennbar. In diesem Fall liegt der Fehler an der verspäteten Treppendetektion durch Luftdruckmessungen. Dies erzeugt einen Distanzfehler in Laufrichtung auf der Treppe und führt zu einem längeren Treppenabgang, daher verkürzt sich das anschließende gegensätzliche Streckenstück, das zum Rundgang führt, um ca. 2 m. Die ungenaue Positionsschätzung wird anschließend durch einen Wanddurchbruch und das Ankoppeln an die Routing-Kanten kompensiert. Das zeigt sich deutlich

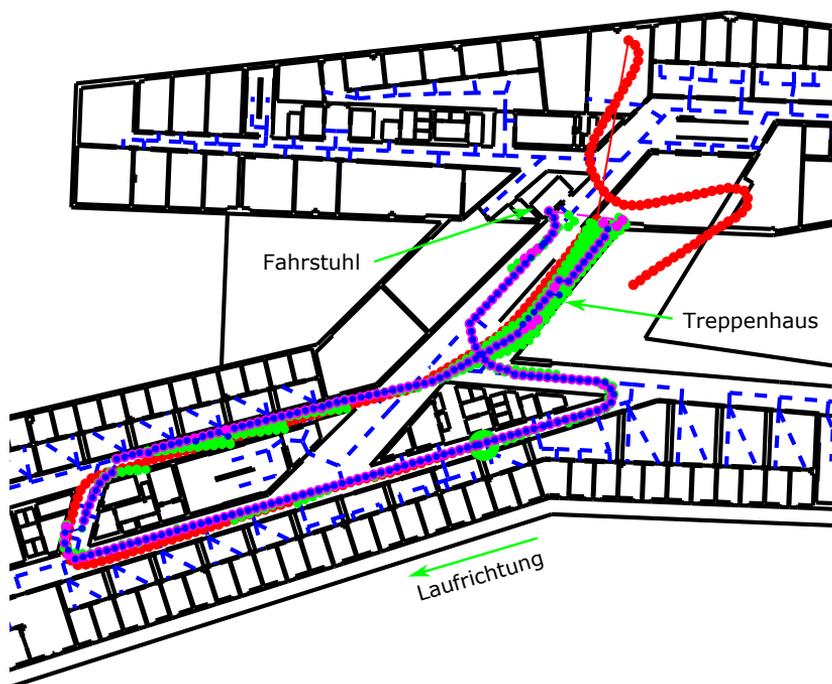


Abb. 5.31: Trajektorie des zweiten Testdatensatzes für das vierte Obergeschoss auf Basis des kantenbasierten PF (rot: PDR, grün: Partikel, magenta: favorisierte Route nach Anz. Partikel, blau: favorisierte Route nach Summe Gewicht; magenta und blau überlagern sich).

bei den favorisierten Positionen. Dabei scheint die Zielroute in Abhängigkeit der Gewichtssumme einen schnelleren Weg zurück auf die Referenztrajektorie zu berechnen. Das Ankoppeln der Trajektorie an den Kanten ist zum Teil sprunghaft. Dreimal findet eine Rückkopplung der favorisierten Trajektorie aus PDR PF Partikeln an eine Routing-Kante statt und erlaubt eine der Referenztrajektorie weitgehend angepasste Positionsschätzung. Es wird hier allerdings deutlich, dass der Stockwerksübergang und besonders die Treppenerkennung hohe Unsicherheiten in der Positionsschätzung hervorrufen können und eine effizientere Treppenerkennung erforderlich machen.

Im Erdgeschoss (Abb. 5.33) haben die bislang vorgestellten Ansätze zum PF und der topologische Ansatz weitgehend versagt, die Stützkriterien für die Routing-Kanten üben in den offenen Bereichen einen zu starken Zwang aus, da die Einstel-

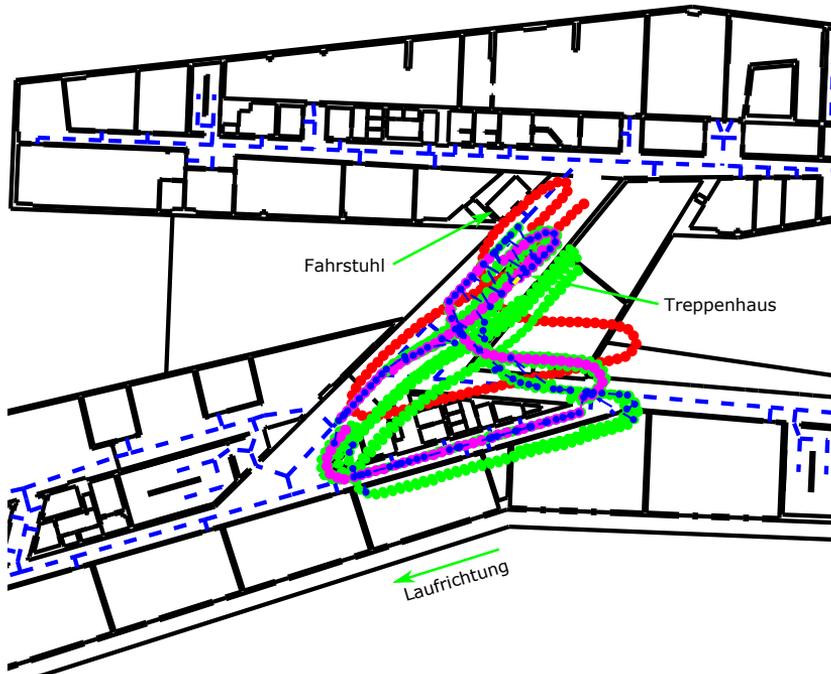


Abb. 5.32: Trajektorie des zweiten Testdatensatzes für das zweite Obergeschoss auf Basis des kantenbasierten PF (rot: PDR, grün: Partikel, magenta: favorisierte Route nach Anz. Partikel, blau: favorisierte Route nach Summe Gewicht; magenta und blau sind überlagert dargestellt).

lungen durchgängig für das gesamte Gebäude gelten sollen. Ziel des Ansatzes ist zwar eine Stützung auf Basis der Routing-Kanten, allerdings soll eine stark fehlerhafte Trajektorienbestimmung in offenen Bereichen vermieden werden. Im Erdgeschoss findet nur das PDR PF Anwendung, da die Routing-Kanten zu weit entfernt oder zu große Richtungsabweichungen zu den Partikelrichtungen aufweisen. Allerdings stellt auch dieses Ergebnis nur eine verbesserte Annäherung an die Referenztrajektorie dar und es treten Abweichungen zur Referenz von bis zu 6 m auf. Daher ist auch hier eine Positionsstützung beim Stockwerksübergang notwendig, damit die Fehler nicht in andere Geschossen übertragen werden.

In Tab 5.2 in Kapitelabschnitt 5.4 *Vergleich und Diskussion der vorgestellten Ansätze* wurden die Koordinatendifferenzen für die drei Ausgangsalgorithmen zu allen Treppen und Fahrstühlen im Gebäude aufgeführt, um den richtigen Stock-

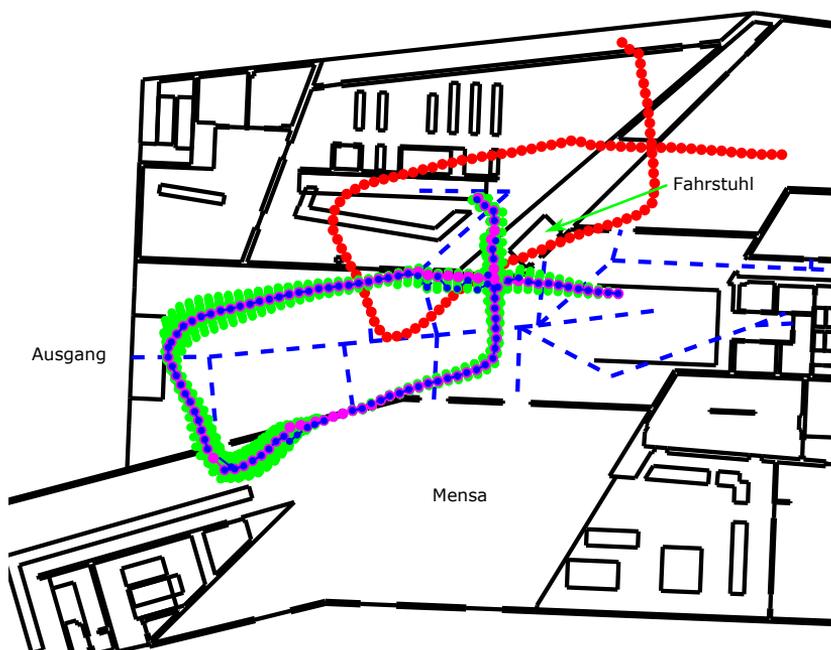


Abb. 5.33: Trajektorie des zweiten Testdatensatzes für das Erdgeschoss auf Basis des kantenbasierten PF (rot: PDR, grün: Partikel, magenta: favorisierte Route nach Anz. Partikel, blau: favorisierte Route nach Summe Gewicht; magenta und blau sind überlagert dargestellt).

werksübergang für das CUPT zu identifizieren. In Tab 5.3 befinden sich nun die Koordinatenvergleiche für den Zeitpunkt der Stockwerksidentifizierung für das kantenbasierten PF. Zum Vergleich werden die Ergebnisse des PF und des kantenbasierten PDR erneut aufgeführt. Es können alle Stockwerksübergänge richtig identifiziert werden, solange der Schrittzähler Treppen und Fahrstühle unterscheiden lässt. Gegenüber den Ausgangsalgorithmen zeigt sich nur im ersten Übergang vom OG4 nach OG3 eine Verschlechterung der geschätzten Position zur Position des Stockwerksübergangs zum kantenbasierten PDR. Grundsätzlich beeinflussen die Treppen die Positionsschätzung stark und die zu erreichende Zielgenauigkeit wird durch die verspätete Treppenerkennung fast nie erreicht.

Die Testdaten belegen eine gute Einsatzmöglichkeit des kantenbasierten PF für die autonome Positionsschätzung. Gegenüber den auf KF, PF und Topologie basierenden Ansätzen zeigen sie signifikante Verbesserungen. Allerdings lassen die

Tab. 5.3: Koordinatendifferenzen vor dem Stockwerksübergang zu den Korrekturpositionen für das kantenbasierten PF sowie PF und kantenbasierten PDR. Mögliche Orte des Stockwerksübergangs bzw. für ein CUPT werden fett gedruckt dargestellt und ergeben sich aus der Treppenerkennung und Luftdruckveränderungen. Fettgedruckte Differenzen repräsentieren die geringste Koordinatendifferenz und damit den ausgewählten Stockwerksübergang.

	PF	Topo.	k. PF Anz.	k. PF Gew.
OG4 nach OG3 [m]				
Treppe EG-OG1	3,5	6,9	3,0	2,8
Haupttreppen OG1-OG4	11,4	7,7	11,3	9,7
Nebentreppen	28,6-46,1	32,1-40,9	28,7-46,0	30,4-44,4
Fahrsstuhl links	7,6	2,6	7,9	6,7
Fahrsstuhl rechts	6,6	3,4	6,9	6,2
Lastenfahrsstuhl	30,7	36,4	30,7	32,4
OG2 nach OG1 [m]				
Treppe EG-OG1	6,2	8,2	4,1	6,9
Haupttreppen OG1-OG4	11,9	14,9	6,8	7,7
Nebentreppen	29,5-45,6	27,2-48,4	33,1-41,4	32,1-40,8
Fahrsstuhl links	5,5	7,8	5,8	2,6
Fahrsstuhl rechts	4,0	5,9	6,3	3,4
Lastenfahrsstuhl	31,9	29,6	35,2	36,4
EG nach OG4 [m]				
Treppe EG-OG1	10,2	13,6	14,7	14,7
Haupttreppen OG1-OG4	13,9	20,9	14,0	13,9
Nebentreppen	30,9-45,6	23,1-53,9	24,9-44,5	24,8-44,5
Fahrsstuhl links	5,0	13,2	5,7	5,6
Fahrsstuhl rechts	3,1	11,2	5,9	6,0
Lastenfahrsstuhl	33,5	26,0	39,9	40,0

Testdaten keine Rückschlüsse auf die unterschiedlichen Nutzerprofile zu. Um einen Überblick über die Übertragbarkeit des aktuellen Entwicklungsstandes auf ungeübte Smartphone-User zu erhalten, werden Testdaten einer größeren Nutzergruppe im folgenden Kapitel untersucht.

5.5.3 Untersuchungen zur Zuverlässigkeit

Die autonome MEMS-INS-basierte Positionsschätzung unterliegt vielen Einflüssen. Sie können sich aus der Software, den verwendeten Algorithmen oder dem Nutzerverhalten ergeben, deren Zusammenspiel je nach Smartphone und Nutzer variiert. Die folgende Liste gibt einen Überblick über die Einflüsse und Parameter, die eine Schätzung mit den kantenbasierten PF verfälschen können.

- Abtastrate Sensoren
- Schritterkennung
- Treppenerkennung
- Qualität Karte
- Gebäudetopographie
- Rauschen für Partikelstreuung
- Anzahl Partikel
- Messunsicherheiten des INS
- Schrittlänge
- Zeitpunkt Erkennung Stockwerksübergang
- Routing-Graph
- Grenzwert Kopplung Routing-Kanten
- Rauschwert für w_i in WDF

Die Abtastrate und die Messunsicherheit der MEMS-Sensoren wirken direkt auf die Orientierungsbestimmung, Verebnung und die Schritterkennung. Ist die Abtastrate der Sensoren zu gering, werden Schritte durch den Schrittzähler nicht erkannt. Schnelle Drehungen des Nutzers können durch das Gyroskop nicht vollständig erfasst werden. Das Schrittmuster im Beschleunigungsmesser variiert erheblich durch das individuelle Laufverhalten des Nutzers. Die vom Schrittzähler nicht oder zu viel erkannten Schritte wirken maßgeblich durch Distanzfehler in der Positionsschätzung. Eine hohe Abweichung der Schrittlänge zur angenommenen mittleren Schrittlänge wirkt in gleicher Weise. Die verspätete Treppen-erkennung, die aus relativen Luftdruckmessungen abgeleitet ist, erzeugt zudem zu lange Laufstrecken, da die Treppenstufenlänge üblicherweise kürzer als die mittlere Schrittlänge ist. Der Zeitpunkt der Detektion des neuen Stockwerks stellt eine weitere

Fehlerquelle dar, wenn beispielsweise Position und Richtung zu einem falschen Zeitpunkt korrigiert werden.

Im kantenbasierten PF sollen diese Einflüsse durch die Zustandserkennung auf dem Routing-Graphen minimiert werden. Die maßgeblichen Parameter für den kantenbasierten PF sind die Anzahl der Partikel, der Grenzwert der Richtungs-differenz für die Kopplung des Partikels zu den Routing-Kanten, die Streuung der Partikel auf Basis der Gyroskop-Orientierung und das Messrauschen für die Berechnung der Partikelgewichte w_i in der WDF.

Die bisherigen Einstellungen des Filters sollen für eine Einschätzung der Anwendbarkeit des aktuellen Entwicklungsstands auf 46 Datensätze angewendet werden. Dazu haben 24 Personen jeweils zwei vorgegebene Routen mit unterschiedlichen Smartphones abgelaufen. Für den Testlauf ohne Stockwerksübergang liegen 22 Datensätze vor, die vom Seminarraum (Raum 3.110) im HCU-Gebäude zum Eingang der Instrumentenausgabe (Raum 3.006) und zurück zum Seminarraum führen. Für die Untersuchung mit Stockwerksübergängen liegen 24 Datensätze vor. Die hier vorgegebene Route führt vom Seminarraum (Raum 3.110) über das Haupttreppenhaus zum Hauptaussgang im Erdgeschoss und zurück über den Fahrstuhl zum Seminarraum. Beide Routen repräsentieren eine typische Navigationsanfrage zur Instrumentenausgabe bzw. zum Ausgang des Gebäudes. Jede Route enthält eine 180° Drehung und die Routen mit Stockwerkswechsel berücksichtigen Treppenhaus und Fahrstuhl.

Die Datensätze repräsentieren keine normalverteilte Stichprobe der Gesellschaft, liefern aber einen fundierten Eindruck zur Anwendbarkeit des vorgestellten Fusionsalgorithmus gemäß dem aktuellen Entwicklungsstand. Für den 2D-Datensatz liegen 4 Trajektorien zu weiblichen und 18 zu männlichen Nutzern vor. Der zweite Datensatz enthält Messdaten zu 18 männlichen und 6 weiblichen Nutzern. Der erste Datensatz ohne Stockwerkswechsel erlaubt die Nutzung von Smartphones ohne Barometer. Insgesamt wurden sieben verschiedene Smartphones verwendet, vor allem die Testgeräte Google Nexus 4 und Samsung Galaxy Nexus. Zwei bis vier Datensätze wurden mit Smartphones der Modelle Samsung Galaxy S4 und Sony Z3 generiert. Das Nexus 7, das Samsung Galaxy S3 Neo und das Samsung

Galaxy Node 3 Neo besitzen keinen Barometer und werden jeweils nur für den 2D-Datensatz eingesetzt.

Für jeweils zwei Trajektorien der beiden Datensätze ließen sich keine brauchbaren Positionen berechnen. Die aufgezeichneten Sensordaten wiesen kein brauchbares Schrittmuster auf. Dies zeigt, dass bei der MEMS-INS-basierten Positionsschätzung die Qualität der Positionsschätzung stark von dem Nutzerverhalten abhängt. Diese Einflüsse können zwar bei Personen ohne Hintergrundwissen durch Bedienungsanweisungen minimiert werden, allerdings lassen sich die Unsicherheiten durch fehlerhafte Bedienung nicht völlig ausschließen.

2D Auswertung

Zur Veranschaulichung der Ausgangsdaten zeigt Abb. 5.34 die mittels PDR berechneten 20 Trajektorien für den ersten Datensatz im dritten Obergeschoss. Die Trajektorie startet im linken unteren Bereich an der Eingangstür zum Seminarraum (Raum 3.110). Anschließend führt sie am Haupttreppenhaus vorbei zur Instrumentenausgabe im Nordteil des HCU-Gebäudes. Die Ergebnisse belegen schon nach wenigen Metern große Unsicherheiten beim PDR. Dies liegt zum einen an einer ungenauen Startausrichtung des Nutzers, die beim PDR nicht per Routing-Kante korrigiert wird, und zum anderen an den hohen Messunsicherheiten der aus Drehraten berechneten Orientierung. Hieran wird deutlich, wie sehr sich die Ausgangsdaten der 20 Nutzer für das kantenbasierte PF unterscheiden und wie wichtig mithin das Benutzerverhalten für die Qualität der Positionsschätzung per Smartphone ist.

Jede Trajektorie wurde mit dem kantenbasierten PF ausgewertet. Die favorisierten Positionen in Abhängigkeit von der Anzahl der Partikel und der Summe der Partikelgewichte sind in Abb. 5.35 und Abb. 5.36 dargestellt. Die Farbe steht für den jeweiligen Nutzer.

Gegenüber den Ergebnissen des PDR zeichnen die Trajektorien deutlich den abgelaufenen Weg nach. Die Routen überlagern sich verstärkt auf den Routing-Kanten. Bei einigen Trajektorien kommt es zu Abweichungen zwischen Start- und Endposition und einer Überschreitung der Zielgenauigkeit von 5 Metern. Auffällig bei diesen Trajektorien ist das frühzeitige Entkoppeln von den Routing-Kanten, so

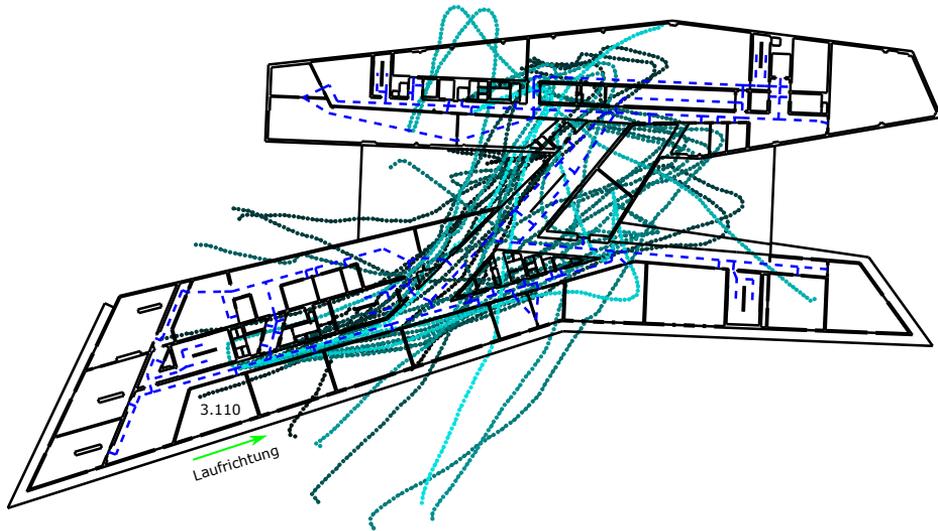


Abb. 5.34: Trajektorien von 20 Nutzern berechnet mit Pedestrian Dead Reckoning für eine Route im dritten Obergeschoss zwischen Raum 3.110 und Instrumentenausgabe (schwarz-cyan: jeweils eine Route bzw. Nutzer).

dass die Positionsschätzung hauptsächlich über den implementierten PDR PF läuft.

Die gewählte Test-Route zeigt eine Abhängigkeit zur Schrittlänge, da sie lange Strecken ohne Richtungsänderung enthält und den Partikelpositionen bei zu großen Positionsabweichungen keine Routing-Kante zugeordnet werden kann. Im oberen Teil der Abbildung ist eine Linksabbiegung zur Instrumentenausgabe (Raum 3.006) notwendig. Hier streuen die Trajektorien besonders, da der Einfluss der Schrittlänge aufgrund der langen und geraden Strecke zunimmt. Den Nutzern wurde es überlassen, welchen Eingang sie zur Instrumentenausgabe wählen, daher verlaufen einige Trajektorien in den Raum und andere stehen für eine Umkehr im Flur. Auffällig sind zwei Trajektorien, die den Bogen darstellen, den die Nutzer um den dreieckigen Mittelbau des südlichen Gebäudeteils genommen haben. Diese Routen sind besonders gut dargestellt. Vermutlich wird hier die Schrittlänge durch die zusätzlichen Kurven besser abgefangen oder die Personen weisen genau die mittlere angenommene Schrittlänge auf.

Im Vergleich der beiden favorisierten Routen zeigen sich qualitativ keine Unter-

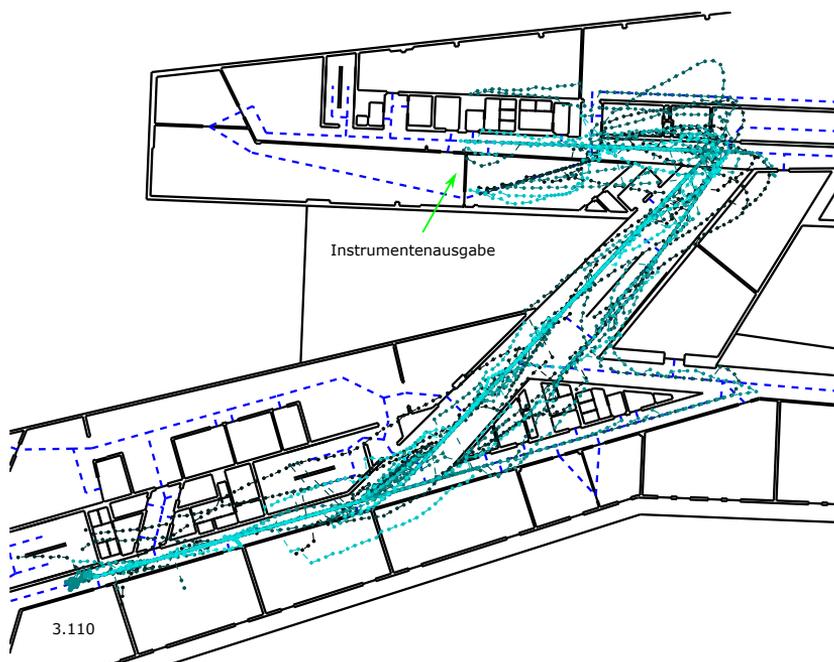


Abb. 5.35: Trajektorien von 20 Nutzern berechnet mittels kantenbasiertem Partikelfilter. Auswahl der favorisierten Route nach Anzahl zugeordneter Partikel der jeweiligen Routing-Kanten (schwarz-cyan: jeweils eine Route bzw. Nutzer).

schiede. Allerdings gibt es in einer der Trajektorien ausgewählt nach der Summe der Partikelgewichte (Abb. 5.36) Positionssprünge, die durch zur Trajektorie senkrecht verlaufene Linien sichtbar sind. Hier scheinen zwei Punktgruppen nahezu das gleiche Gewicht zu vereinen, was nach den jeweiligen Schritten, je nach Gewichtsanzpassung zu einem Wechsel der favorisierten Route führt.

Für einen Koordinatenvergleich wurden die Start- und Endpunkte jeder Trajektorie eines Nutzers miteinander verglichen. Der Start- und der Endpunkt jeder Trajektorie sollen identisch sein. Die Instrumentenausgabe (Raum 3.006) als Zwischenpunkt der Route ist variabel, da einige Nutzer vor der Eingangstür kehrt gemacht haben und andere in den Raum eingetreten sind. Die Tab. 5.4 zeigt die Differenzen zwischen Start- und Endpunkt für die favorisierten Routen des kantenbasierten PF sowie für das PDR. Zur besseren Übersicht sind die Differenzen in Klassen eingeteilt. Es wird außerdem unterschieden nach dem Geschlecht sowie nach dem Smartphonemodell.

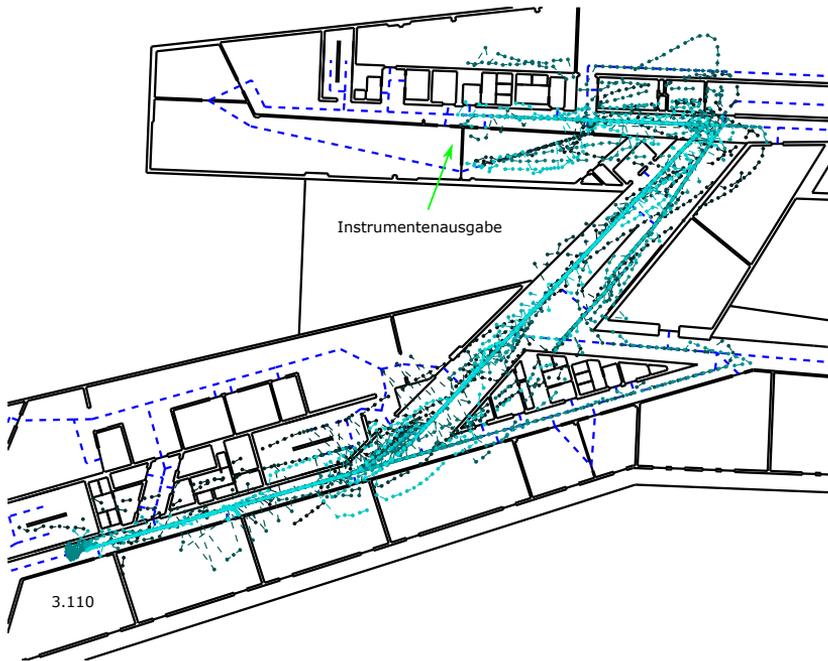


Abb. 5.36: Trajektorien von 20 Nutzern berechnet mittels kantenbasiertem Partikelfilteransatz. Auswahl der favorisierten Route nach der Gewichtssumme der zugeordneten Partikel der jeweiligen Routing-Kanten (schwarz-cyan: jeweils eine Route bzw. Nutzer).

Das Ziel einer maximalen Positionsabweichung von kleiner fünf Meter wird beim kantenbasierten PF für 14 Trajektorien erreicht. Das PDR ohne Korrekturen liefert keine brauchbare Trajektorie in der Zielgenauigkeit. Die Berechnung der favorisierten Routen in Abhängigkeit der Anzahl und der Summe der Gewichte zugeordneter Partikel weisen untereinander nur leichte Unterschiede in den Genauigkeitsklassen auf. Die Ergebnisse, klassifiziert nach männlichen und weiblichen Nutzern, basieren auf der favorisierten Route mit Hilfe der zugeordneten Anzahl an Partikel im kantenbasierten PF. Sie zeigen keine Auffälligkeiten im Hinblick auf das Geschlecht, allerdings erlaubt die geringe Anzahl weiblicher Nutzer auch keine wirklich repräsentative Aussage. Aufgrund der kleinen Stichprobe lassen sich auch keine Auffälligkeiten nach genutzten Smartphones ausmachen. Allerdings kommt es beim Galaxy Nexus (GN) im Vergleich zu einer hohen Anzahl an Ausreißern. Insgesamt belegen die empirischen Ergebnisse, dass der kantenbasierte PF in 70 %

Tab. 5.4: Vergleich der Differenzen zwischen Start- und Endpunkt der insgesamt 20 verwendeten 2D-Trajektorien im dritten Obergeschoss nach den unterschiedlichen Berechnungsmethoden. Zuordnung in Genauigkeitsklassen.

Diff.	Algorithmus			Geschlecht		Smartphonemodell		
	kanten. PF nach Anz.	kanten. PF nach Gew.	PDR	w. (4)	m. (16)	Smartphone andere (6)	GN (8)	N4 (6)
0-5 m	14	14	0	2	12	5	4	5
5-10 m	2	1	2	1	1	0	2	0
+10 m	4	5	18	1	3	1	2	1

der Fälle ausreichende Positionsgenauigkeiten für die Indoor-Navigation liefert. In 30 % der Fälle erreicht die Positionsschätzung mit den voreingestellten Parametern nicht die Zielgenauigkeit von fünf Meter. Durch eine weitere Anpassung der Parameter im Filter wäre zwar eine leichte Verbesserung der Positionsschätzung zu erreichen, sie lässt sich aber in dieser Stichprobe keinem Nutzertypus wie Geschlecht, Smartphonemodell oder Körpergröße zuordnen.

2D+1D Auswertung

Der 2D+1D Datensatz beinhaltet in jeder Trajektorie vier Stockwerke (OG3-EG-OG3). Auf eine Darstellung der Stockwerke OG1 und OG2 wird verzichtet, da sich die kritischen Bereiche im dritten Obergeschoss sowie im Erdgeschoss befinden. Im dritten Obergeschoss befindet sich der Start- und Endpunkt der Route beim Seminarraum (Raum 3.110) und im Erdgeschoss befindet sich der große offene Eingangsbereich mit wenigen Routing-Kanten. Für einen Überblick der Qualität der Eingangsdaten werden die 22 Datensätze mit dem PDR berechnet und in Abb. 5.37 sowie Abb. 5.38 für die ausgewählten Stockwerke dargestellt. Die Trajektorien variieren schon zu Beginn sehr stark und der Positionsfehler wird, da beim Stockwerksübergang keine Korrektur stattfindet, ins nächste Stockwerk übertragen.

Die 22 verwertbaren Datensätze wurden mit dem kantenbasierten PF berechnet. Die favorisierten Trajektorien sind in Abhängigkeit zur Anzahl zugeordneter Partikel in Abb. 5.39 und Abb. 5.40 zu sehen. Zu Beginn der jeweiligen Trajektorien

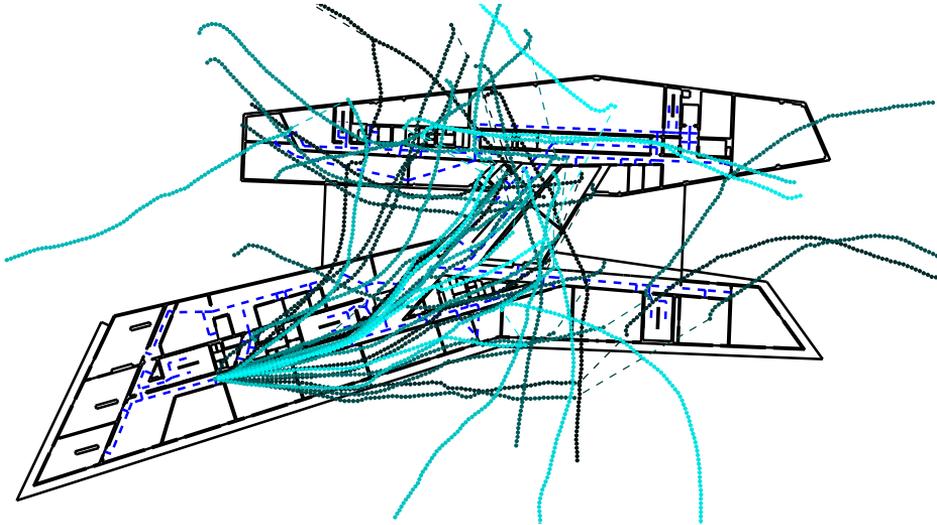


Abb. 5.37: Trajektorien im dritten Obergeschoss berechnet auf Basis von PDR für die 22 verwendeten Datensätze der räumlichen Untersuchung (schwarz-cyan: jeweils eine Route bzw. Nutzer; farblich skalierte Routen können abbildungsübergreifend zugeordnet werden).

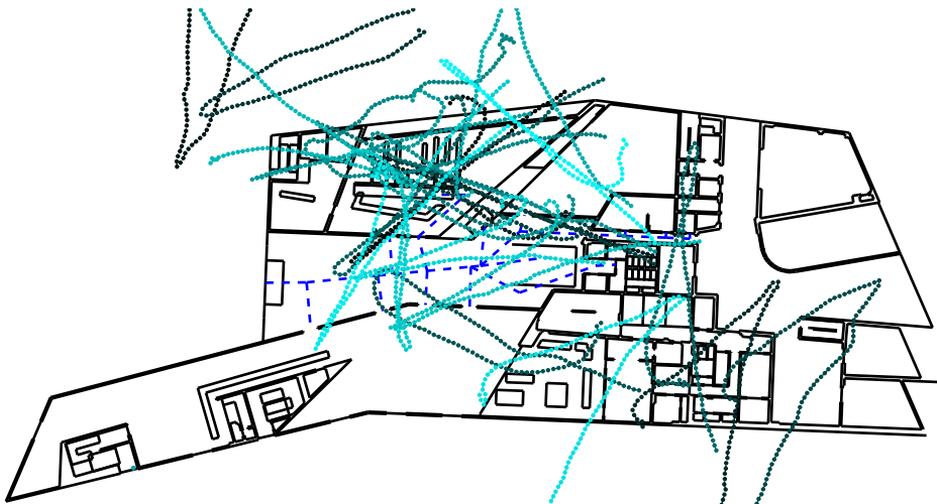


Abb. 5.38: Trajektorien im Erdgeschoss berechnet auf Basis von PDR für die 22 verwendeten Datensätze der räumlichen Untersuchung (schwarz-cyan: jeweils eine Route bzw. Nutzer; farblich skalierte Routen können abbildungsübergreifend zugeordnet werden).

wird die fehlerhafte Ausrichtung des Nutzers in der Startrichtung durch den Einsatz von Routing-Kanten im Filter reduziert. Im dritten Obergeschoss startet die Route beim Raum 3.110, verläuft weiter durch das Treppenhaus bis in das Erdgeschoss und mit dem Fahrstuhl zurück in das dritte Obergeschoss zum Raum 3.110.

Im dritten Obergeschoss werden viele Routen parallel zum Treppenhaus berechnet, da ein Entkoppeln von den Routing-Kanten nicht stattfindet. Die Mehrzahl der Routen überlagert sich auf den Routing-Kanten, nur vereinzelt weichen Trajektorien stark vom Referenzweg ab. Im Erdgeschoss rauschen die Positionen gegenüber den Ergebnissen aus dem dritten Obergeschoss stärker. Zwar ist der Großteil der 22 Datensätze den Routing-Kanten zugeordnet, die zum Ausgang führen, allerdings weist ein Teil der Trajektorien keine zufriedenstellenden Ergebnisse auf. Besonders die 180°-Kehrtwende führt zu großen Unsicherheiten, da die Routing-Kante auf dem Rückweg hin zum Fahrstuhl von dem Großteil der Partikel nicht genutzt wird.

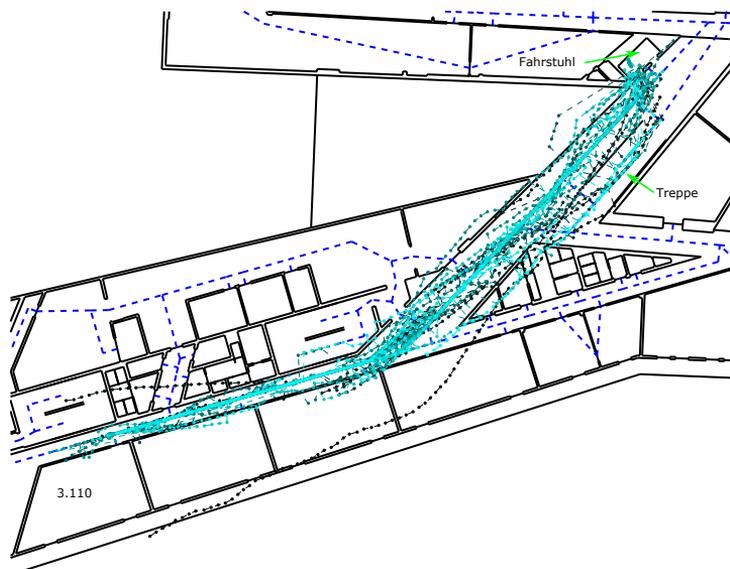


Abb. 5.39: Trajektorien im dritten Obergeschoss berechnet auf Basis des kantenbasierten PF für die 22 verwendeten Datensätze der räumlichen Untersuchung. Favorierte Route durch die Partikelanzahl bestimmt. (schwarz-cyan: jeweils eine Route bzw. Nutzer; farblich skalierte Routen können abbildungsübergreifend zugeordnet werden).

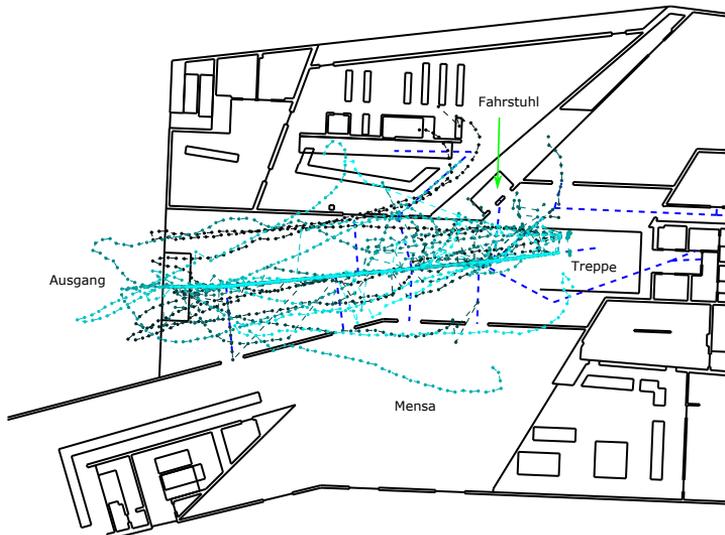


Abb. 5.40: Trajektorien im Erdgeschoss berechnet auf Basis des kantenbasierten PF für die 22 verwendeten Datensätze der räumlichen Untersuchung. Favorisierte Route durch die Partikelanzahl bestimmt. (schwarz-cyan: jeweils eine Route bzw. Nutzer; farblich skalierte Routen können abbildungsübergreifend zugeordnet werden)

Die zweite favorisierte Möglichkeit, um die Trajektorien aus der Partikelwolke abzuleiten, ergibt sich aus der Bestimmung der Summe der Partikelgewichte zu jeder Routing-Kante. Gegenüber dem vorherigen Ansatz zur Partikelanzahl können mit diesem Ansatz stark gewichtete Positionen bevorzugt werden, obwohl die Anzahl der zugeordneten Partikel gering ist. Die Ergebnisse sind in Abb. 5.41 und Abb. 5.42 dargestellt. Wie zuvor beim Ansatz der Partikelanzahl streuen einzelne Trajektorien besonders im Erdgeschoss stark. Eine relevante Veränderung im Bezug zur erreichten Positionsgenauigkeit wird gegenüber der Auswahl nach der Partikelanzahl nicht ersichtlich. Allerdings kommt es bei drei Routen zu Positionssprüngen von wenigen Metern zwischen möglichen parallelen Trajektorien. Dies ergibt sich, wenn sich das Gewicht zwischen zwei Partikelwolken nahezu gleich aufteilt, aber durch die Summe des Gewichts abwechselnd einer der beiden Partikelwolken für die favorisierte Trajektorie ausgewählt wird. Dies lässt sich auf Trajektorien zurückführen, bei dem die Masse an Partikel als PDR PF berechnet wird. Dabei wird die Berechnung der Gewichte schnell gleichartig, da sich sie sich nur auf die Differenz zur Orientierung aus Gyroskopdaten stützt.

Eine Alternative zur Berechnung der Partikelgewichte der Partikel im PDR PF ist zurzeit noch nicht gefunden. Hier könnte aber eine Kombination der beiden Auswahlmöglichkeiten zur Bestimmung der favorisierten Routen helfen. Letztendlich muss ein Wechsel der favorisierten Route ermöglicht werden, damit falsche Routen auch verlassen werden können.

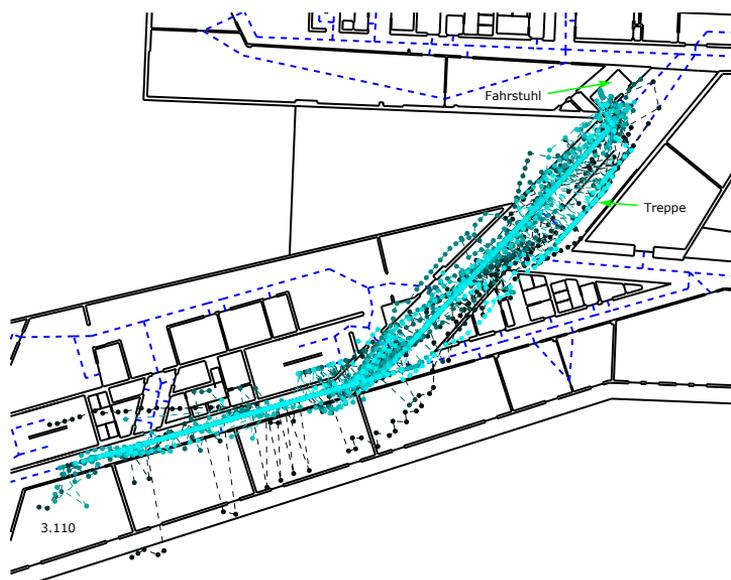


Abb. 5.41: Trajektorien im dritten Obergeschoss berechnet auf Basis des kantenbasierten PF für die 22 verwendeten Datensätze der räumlichen Untersuchung. Favorisierte Route durch die Summe der Partikelgewichte bestimmt. (schwarz-cyan: jeweils eine Route bzw. Nutzer; farblich skalierte Routen können abbildungsübergreifend zugeordnet werden).

Der Koordinatenvergleich erfolgt wie im 2D-Datensatz zwischen Start- und Endkoordinaten jeder Trajektorie. Tab. 5.5 zeigt die Differenzen aller 22 Trajektorien in den drei Genauigkeitsklassen an. Bei 14 bzw. 15 Datensätzen liegen die Differenzen für den kantenbasierten PF innerhalb der Zielgenauigkeit von <5 Meter. Gegenüber den 2D-Datensätzen gibt es nur einen Datensatz im Bereich $+10$ m. Da beim Stockwerksübergang eine Positions- und Richtungskorrektur stattfindet, ist die Route zwischen Fahrstuhl und Seminarraum (Raum 3.110) kürzer als die Strecke im 2D-Datensatz und unterliegt weniger Fehlereinflüssen. Voraussetzung ist der richtige Zeitpunkt der Erkennung des Stockwerksübergangs, damit die

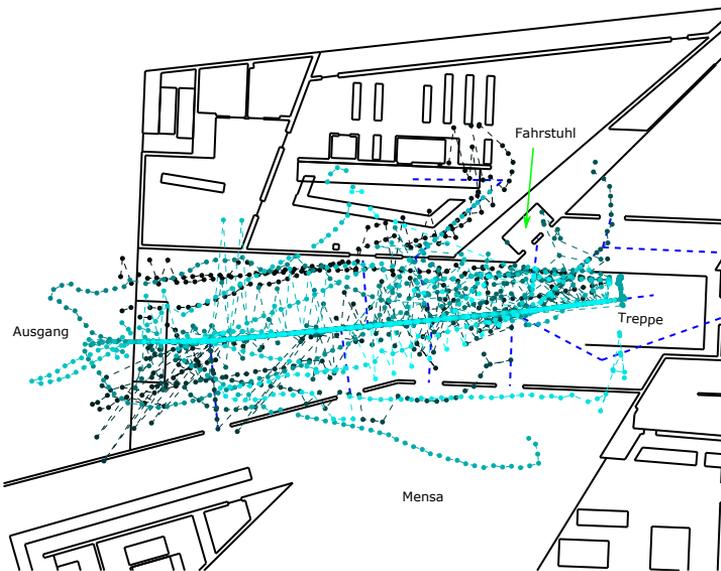


Abb. 5.42: Trajektorien im Erdgeschoss berechnet auf Basis des kantenbasierten PF für die 22 verwendeten Datensätze der räumlichen Untersuchung. Favorisierte Route durch die Summe der Partikelgewichte bestimmt. (schwarz-cyan: jeweils eine Route bzw. Nutzer; farblich skalierte Routen können abbildungsübergreifend zugeordnet werden).

Korrekturen keine zusätzlichen Fehler verursachen. Die Ergebnisse des PDR sind aufgrund der fehlenden Korrekturen für die Positionsschätzung nicht verwendbar.

Die Koordinatenvergleiche für Geschlecht und Smartphonemodell entstammen der favorisierten Route auf Basis der Anzahl zugeordneter Partikel. Aufgrund der gesellschaftlich nicht repräsentativen Datensätze dieser Stichprobe können Unterschiede nach Geschlecht und Smartphone-Modell nur vermutet werden. Die Ergebnisse jedenfalls zeigen keine Auffälligkeiten.

Fazit

Insgesamt zeigt sich, dass der in dieser Arbeit entwickelte kantenbasierte Partikelfilter in der Praxis zur Positionsschätzung verwendet werden kann. Voraussetzungen sind allerdings ein geübtes Nutzerverhalten und im Smartphone integrierte Sensoren mit hoher Abtastrate. Je nach Route wirken sich die beschriebenen Einflüsse unterschiedlich stark aus. Lange und gerade Strecken erfordern beispielsweise eine festgelegte Schrittlänge. Abweichungen lassen sich im Filter nicht immer

Tab. 5.5: Vergleich der Differenzen zwischen Start- und Endpunkt der insgesamt 22 verwendeten räumlichen Trajektorien nach den unterschiedlichen Berechnungsmethoden. Zuordnung in Genauigkeitsklassen.

Diff.	Algorithmus			Geschlecht		Smartphonemodell		
	kanten. PF nach Anz.	kanten. PF nach Gew.	PDR	w. (6)	m. (16)	Smartphone andere (3)	GN (10)	N4 (9)
0-5 m	15	14	0	5	10	1	7	7
5-10 m	6	7	2	1	6	1	3	2
+10 m	1	1	20	0	0	1	0	0

korrigieren, da der Ansatz fehlerhafte Schrittlängen nur bei Abbiegungen minimiert.

Einen großen Einfluss auf die Navigation in mehreren Stockwerken hat die zeitliche Erkennung des Stockwerksübergangs. Zwar lassen sich die Etagen deutlich unterscheiden, aber der Zeitpunkt der Erkennung ist abhängig vom Messrauschen des Barometers und beeinflusst den Zeitpunkt der Positions- und Richtungskorrektur. Eine Kombination mit anderen Sensordaten wäre wünschenswert, um den Zeitpunkt des Stockwerksübergangs hardware- und nutzerunabhängig identifizieren zu können. In diesem Zusammenhang könnte die Treppenerkennung überarbeitet werden, damit auch hier die Schrittlängen Anpassung den Treppen besser zugeordnet werden kann.

Anhand der 46 Datensätze ergibt sich, dass eine Positionsschätzung auf Basis des kantenbasierten PF in 70 % der Fälle erfolgreich verlaufen ist. Eine Anpassung der Parameter im PF erhöht zwar die Positionsgenauigkeit einzelner Datensätze, allerdings lässt sich die Anpassung der Parameter nicht verallgemeinern.

6 Implementierung für eine Navigationsanwendung

Für eine Navigation in GNSS abgeschatteten Bereichen bieten sich unterschiedliche Verfahren der Positionsschätzung an. In dieser Arbeit wird eine möglichst autonome Variante angestrebt. Die Realisierung einer Navigationsanwendung benötigt unabhängig von der Variante der Positionsschätzung zusätzliche Komponenten. Das Routing und die Gebäudepläne sind zur Indoor-Navigation unerlässlich und werden im Folgenden diskutiert. Dazu werden am Beispiel einer Fußgänger-Indoor-Navigation durch das HCU-Gebäude zwei Routingalgorithmen umgesetzt und Ansätze zur vollautomatischen Generierung von Karten- und Routing-Daten beschrieben. Zum Abschluss werden die im Rahmen dieser Arbeit zu Evaluierungszwecken entwickelten Apps vorgestellt.

6.1 Weitere Komponenten

6.1.1 Routenberechnung

Sinn und Zweck einer Navigationsapplikation ist die Zielführung. Dazu werden Routingalgorithmen umgesetzt, die auf der Graphentheorie beruhen (siehe Kapitelabschnitt *2.3 Gebäudepläne und Routing-Netz*). Der Dijkstra Algorithmus und der Floyd Warshall Algorithmus [Wilde u. a., 2014; Dalladas u. a., 2014] werden in diesem Kapitel auf ihre Eignung zur Indoor-Navigation untersucht.

Das anhand typischer Laufwege manuell erstellte Routing-Netz, das von CAD-Plänen des Gebäudes abgeleitet ist und bereits zur Positionsschätzung genutzt wurde, soll auch die Basis beider Algorithmen abgeben. Die entsprechenden Daten liegen im Textformat vor und enthalten Knoteninformationen, wie die Koordinaten, den Raumbezug und mögliche Nachbarknoten sowie die Kosten für die Kanten. Die Algorithmen wurden mithilfe der Software Matlab entwickelt und auf das vorhandene Routing-Netz appliziert.

Die Grundlage für den Dijkstra Algorithmus bildet die Kantenliste. Sie enthält jeweils zwei Knotenverbindungen und deren Kosten, für die im vorliegenden Ansatz die Entfernungen stehen. Die Länge der Kantenliste entspricht dabei der Anzahl der Kanten. Aus der Kantenliste wird anschließend eine Adjazenzmatrix

erstellt, die daher auch als Nachbarschaftsmatrix bezeichnet wird. Sie speichert die Knotenbeziehungen untereinander und weist entsprechend die Spalten- und Zeilenlänge in Abhängigkeit von der Anzahl der Knoten auf. Sofern zwei Knoten durch eine Kante verbunden sind, werden die Kosten im zugehörigen Feld der Matrix abgelegt. Die übrigen nicht vorhandenen Verbindungen erhalten ein unendlich hohes Gewicht, da kein direkter Weg existiert. Die Diagonale der Adjazenzmatrix wird mit Nullen aufgefüllt, da jeder Knoten mit sich selbst eine Entfernung gleich null besitzt und daher keine Kosten aufweisen kann.

Mit Hilfe der Adjazenzmatrix wird der Dijkstra-Algorithmus angewendet. Schrittweise wird die günstigste Verbindung vom Startpunkt fortgeführt, wie in Kapitel 2.3 *Gebäudepläne und Routing-Netz* erläutert. Die ausgewählte Route mit den minimalen Kosten wird anschließend im sogenannten *Backtracking* zurückgerechnet, um die Knoten und Kanten der Route zuzuordnen und den Weg grafisch darstellen zu können.

Der Floyd-Warshall-Algorithmus basiert auf einer erweiterten Adjazenzmatrix. Dafür kann die zuvor im Dijkstra-Algorithmus generierte Kantenliste und Adjazenzmatrix ebenfalls Verwendung finden. Während der Dijkstra-Algorithmus für jeden Aufruf die kürzeste Verbindung zwischen zwei Knoten ermittelt, berechnet der Floyd-Warshall-Algorithmus einen zusätzlichen Schritt im Post Processing und bestimmt vorab die kürzesten Verbindungen zwischen allen möglichen Knotenpaaren.

Um die Voraussetzungen dafür herzustellen, wird die Adjazenzmatrix zu einer Distanzmatrix erweitert. Gegenüber dem Dijkstra-Ansatz, in dem nur die direkten Kosten zwischen benachbarten Knoten berücksichtigt werden, erlaubt die Distanzmatrix für jedes Knotenpaar die geringsten Gesamtkosten – auch über mehrere Knoten hinweg. Die Kosten sind über den Abgriff aus der Distanzmatrix bekannt. Zur Berechnung der kürzesten Verbindung wird außerdem eine Pfadmatrix benötigt, denn die Distanzmatrix enthält nur die Kostensummen. Die Pfadmatrix erlaubt eine vom Zielpunkt rückwärtige iterative Zuordnung benachbarter Knotenpunkte auf dem Weg bis zum Startknoten. Die Berechnung der Route ist aufwendiger als mit dem Dijkstra Algorithmus, wenn die Pfadmatrix für jede Route

neu berechnet wird. Das ist allerdings nur bei einer Veränderung der Infrastruktur erforderlich, sofern sie sich im Kanten- und Knotenmodell niederschlägt.

Beide Verfahren sollen nun anhand von fünf Routen im HCU-Gebäude verglichen werden (Tab. 6.1). Der Dijkstra Algorithmus benötigt für jede Strecke nahezu identische Rechenzeiten – unabhängig von der Zahl der Knoten. Anders der Floyd-Warshall Algorithmus. Seine Rechenzeit steigt mit der Knotenanzahl an, da die Rückrechnung mithilfe der Pfadmatrix abhängig von der Anzahl der Kanten ist. Beide Algorithmen liefern die gleichen, kürzesten und in diesen Beispielen richtigen Verbindungen.

Tab. 6.1: Vergleich der Rechenzeit für fünf Testrouten mit dem Floyd-Warshall und dem Dijkstra Algorithmus (abgeleitet nach [Dalladas u. a., 2014])

von	nach	Algorithmus	Länge [m]	Anz. Knoten	Zeit [s]
Mensa	4.183	Floyd-Warshall	71	19	2,46
		Dijkstra			2,02
Ausgang	WC 3. OG	Floyd-Warshall	93	22	2,05
		Dijkstra			2,01
Aufzug EG	5.006a	Floyd-Warshall	74	26	3,33
		Dijkstra			2,02
Mensa	Hausmeisterei	Floyd-Warshall	43	9	0,27
		Dijkstra			2,02
Poststelle	4.175	Floyd-Warshall	98	22	3,71
		Dijkstra			2,02

Die Berechnungen wurden auf einem handelsüblichen Laptop durchgeführt. Es ist davon auszugehen, dass die Rechenzeit auf einem Smartphone sich – je nach Leistung – davon erheblich unterscheidet. Der Dijkstra-Algorithmus erfordert aufgrund der direkten Eingabe der Kantenliste bei sich ändernden Kanten-Knoten weniger Aufwand und die Zeit, die er zur Berechnung der Testrouten benötigt, variiert kaum. Das spricht für seine Implementierung auf dem Smartphone. Unabhängig vom verwendeten Routing-Algorithmus kann die ausgegebene Route zur gewichteten Stützung im Fusionsfilter verwendet werden. Im kantenbasierten

Ansatz kann die Auswahl der Folgekante für jedes Partikel durch die erhöhte Gewichtung der der Route zugeordneten Kanten erfolgen und so die optimale Route für die Positionsschätzung favorisieren.

6.1.2 Raumattribute

Neben Routing und Gebäudeplänen zur Visualisierung sind auch Informationen zur spezifischen Raumnutzung nötig. In der HCU sind dies *Labor*, *Büro*, *WC*, *PC-Pool* und *Seminarraum*. Dieser Basisverwendung werden weitere Attribute wie Raumnummer, Labortyp und Arbeitsplatz der Mitarbeiter zugeordnet, damit das Routing sowohl nutzungs- als auch personenbezogen erfolgen kann. Voraussetzung ist, dass diese Raumattribute den Knoten des Routings zugeordnet werden. Dazu muss eine Schnittstelle zwischen Routing und dem Gebäudeinformationen hergestellt werden.

Eine sinnvolle Verwendung der Raumattribute kann sich auch aus der Kombination mit den Grundrissen im Vektorformat ergeben. Grundrissdaten ohne jegliche Informationen sind für die Nutzer oft unergiebig. Sind den Räumen Attribute zugeordnet, so lassen sich auf dem Display Informationen zum Raumtypus und zur Nutzung in Form von Icons einblenden. Diese Icons können durch die Knoten des Routing-Graphen verortet werden.

Die Gebäude- und Nutzungsinformationen können den Mitarbeiterlisten und Plänen des Facility Managements entnommen werden. Den Daten sind die Koordinaten der Räume und die entsprechenden Routing-Knoten zuzuordnen, damit sie für die Zielführung und als Orientierungshilfen nutzbar sind. Speichergrundlage kann eine Datenbank oder eine einfache Liste bzw. Tabelle mit den definierten Attributen sein.

Für das Betriebssystem Android wird eine SpatiaLite-Datenbank erstellt, die diese Informationen bereitstellt und einen Zugriff über Indizes erlaubt. Die Datenbank lässt sich direkt in die Android-Anwendung einbinden und dem Nutzer über das Herunterladen der Anwendung zur Verfügung stellen. Eine Aktualisierung der Datenbank ist durch eine Aktualisierung der App möglich. Denkbar ist auch eine lokale Speicherung in einem gesonderten Verzeichnis des Smartphones; dadurch

ist die Aktualisierung einzelner Komponenten möglich, ohne die gesamte App neu zu installieren. Bei serverbasierten Systemen können alle Basisdaten dezentral auf dem Server abgelegt werden und sind vergleichsweise einfach zu pflegen. Einziger Nachteil ist die Notwendigkeit einer aktiven Netzwerkverbindung vor einer Navigation.

Während die meisten infrastrukturbezogenen Informationen zu Raumnutzung und Raumnummer lange Zeit unverändert bleiben, verändern sich die personenbezogenen Daten ziemlich häufig – an der HCU etwa aufgrund befristeter Arbeitsverträge. Zumindest für die Aktualisierung der Mitarbeiterlisten ist daher eine serverbasierte Datenhaltung vorteilhaft oder eine regelmäßige Aktualisierung der App erforderlich.

6.1.3 Erstellung von Visualisierungs- und Routinggrundlage

Gebäudeplan und Routing-Graph sind für die Indoor-Navigation notwendig und werden im Vektor- oder Rasterformat eingesetzt. Im Sinne einer komfortablen Implementierung der Indoor-Navigation sollen Routing-Graph und Gebäudeplan möglichst aus vorhandenen Datenmaterial automatisch abgeleitet werden können. Im Rahmen dieser Arbeit werden drei kartenbasierte Komponenten favorisiert:

- Stockwerkspläne im Vektorformat
 - mit zusätzlichen Symbolen der Raumnutzung zum Zweck der Visualisierung,
 - als Grundriss zur Korrektur in der Positionsschätzung (nur Wände).
- Routing-Graph für Wegberechnung und Korrektur in der Positionsschätzung.

Die Stockwerkspläne für die Visualisierung und die Grundrisse für die Korrektur der Positionsschätzung basieren auf der gleichen Datengrundlage, sie stellen nur unterschiedliche Bearbeitungsebenen dar. Der Routing-Graph muss allerdings – basierend auf den Gebäudegrundrissen – neu erstellt werden. Dazu werden jetzt Möglichkeiten vorgestellt, die eine halbautomatische Erstellung von Gebäudeplan und Routing-Graph zulassen, sofern lediglich Teilinformationen vorliegen.

Stockwerksplan aus Flucht- und Rettungsplan

In Peter u. a. [2012] wird die Erstellung eines Grundrisses von dem Foto eines Flucht- und Rettungsplanes beschrieben. Ein Beispielfoto des vierten Obergeschosses der HCU ist in Abb. 6.1 zu sehen. Flucht- und Rettungspläne, die den Grundriss der jeweiligen Stockwerke abbilden, sind öffentlich zugänglich. Sie enthalten aussagekräftige Symbole zur schnellen Orientierung. Daher ist die Übernahme eines Grundrisses aus einem Foto nicht ohne Bildverarbeitung möglich und entsprechend aufwendig.



Abb. 6.1: Fotografie eines Flucht- und Rettungsplans für die Nordseite des vierten Obergeschosses im HCU-Gebäude.

Zur Verarbeitung eines Bildes mit abgebildetem Grundriss sind in Anlehnung an Peter u. a. [2012] folgende Arbeitsschritte nötig:

- Anpassung des Bildes: Projektive Entzerrung auf Ebene sowie Anpassen von Helligkeit und Farbe.
- Hervorhebung des Vordergrunds: Berechnung eines Grauwertbilds und Verschiebung der Grauwertgrenzen, um die Grundrissinformationen hervorzuheben.
- Berechnung eines Binärbildes aus dem Grauwertbild (um die Symbole auszublenzen).
- Ableitung der Konturen: Ableitung der Konturen der Wände aus dem Bi-

närbild, das nur noch Grundrissinformationen enthält, mit Hilfe von Kantenoperatoren.

- Berechnung der Vektordaten (optional je nach Ansatz): Bestimmung von Vektordaten aus den Konturen
- Transformation für die Navigation: Transformation der lokalen Bildkoordinaten in das Bezugskoordinatensystem für die Positionsschätzung; dazu sind zwei Translationen, eine Rotation und der Maßstab, zu bestimmen.

Diese Arbeitsschritte zielen auf die Verarbeitung eines Fotos zu einem Binärbild mit entzerrten Grundriss ab. Das fertige Binärbild wird anschließend einer Kantenextraktion unterzogen. Das Ergebnis sind weitgehend die Konturen des Grundrisses, die sich nun in Vektordaten konvertieren lassen. Zum Schluss kann das Pixelkoordinatensystem in das Koordinatensystem der Navigationsanwendung übertragen werden.

Teilergebnisse der Übertragung des beschriebenen Ansatzes auf das HCU-Gebäude zeigt Abb. 6.2 für den Nordteil des vierten Obergeschosses. Für die Berechnung diente das Foto von Abb. 6.1. In einem ersten Schritt (Abb. 6.2, oben) wurde das Foto manuell zugeschnitten, dann wurden Helligkeit und Farbe angepasst. Anschließend wurden die farbbasierten Symbole durch einen Austausch von Farbwerten zu Weiß aus der Grafik entfernt und die übrigen Daten mit einem Schwellwert zu einem Binärbild berechnet (Abb. 6.2, mitte). Zum Abschluss erfolgt eine Ableitung von Linien aus dem Binärbild (Abb. 6.2, unten). Der weitgehend automatisierte Prozess liefert vergleichsweise gut brauchbare Basisdaten. Allerdings sind weitere manuelle Anpassungen nötig, da die Symbole des Flucht- und Rettungsplans nicht komplett entfernt und die Kanten, abhängig von Bildauflösung und Kontrast, nicht vollständig dargestellt werden.

Voraussetzung für die Nutzung eines Flucht- und Rettungsplans ist die maßstabsgetreue Repräsentation des tatsächlichen Stockwerks. Er sollte den gesamten Gebäudebereich widerspiegeln. Falls mehrere Flucht- und Rettungspläne vorliegen, sollten sie sich dank eines einheitlichen Maßstabs und identischer Darstellungsweise kombinieren lassen. Dies ist beim HCU-Gebäude möglich, wo die Flucht-

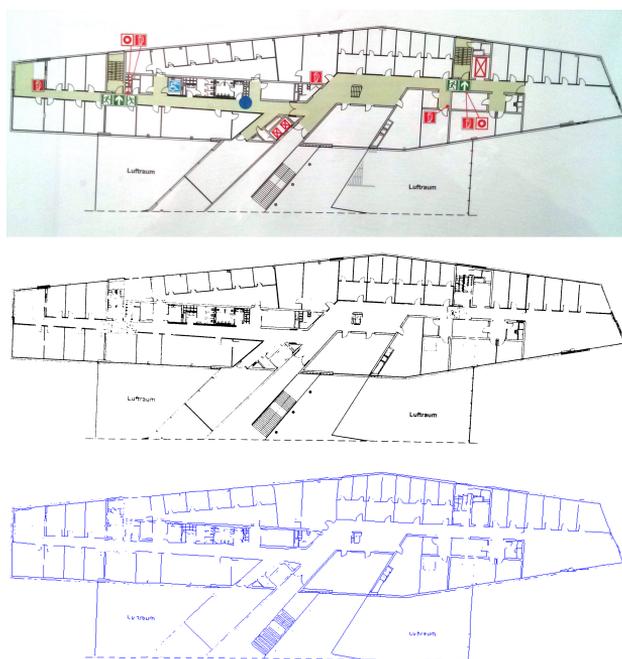


Abb. 6.2: Visualisierung der Teilergebnisse einer Berechnung des vektorbasierten Grundrisses aus einem Bild des Flucht- und Rettungsplans der Nordseite des vierten Obergeschosses der HCU (oben: Helligkeit und Farbe angepasst; Mitte: Farben reduziert und Binärbild erzeugt, unten: Vektordaten aus Binärbild).

und Rettungspläne sich in den Grenzbereichen zwischen dem Übergang zwischen Nord- und Südteil überschneiden.

Insgesamt stellen Flucht- und Rettungspläne eine interessante Alternative als halbautomatische Berechnung von Grundrissen für die Navigation dar, sofern keine digitalen Grundrissdaten zur Verfügung stehen.

Routing-Graph abgeleitet aus Stockwerksplänen

Der Routing-Graph für die innenraumbasierte Fußgängernavigation basiert auf den typischen und möglichen Laufwegen innerhalb eines Gebäudes. Da automatisch aus Kartendaten abgeleitete Informationen diese nicht unbedingt wiedergeben, ist die manuelle Erstellung des Kanten-Knoten-Modells durch eine ortskundige Person vorzuziehen. Um den Arbeitsaufwand gering zu halten, kann die Erstellung von Kanten und Knoten dennoch von Interesse sein. Die so generierten

Daten müssten abschließend einer manuellen Überprüfung und Anpassung unterzogen werden.

Als Vorinformationen für die automatische Kanten- und Knoten-Berechnung werden die Grundrisse des Gebäudes vorausgesetzt. Sie können im Vektor- oder im Rasterformat vorliegen. Um die Funktionsweise eines Routings zu gewährleisten, muss die automatische Kantenberechnung folgende Kriterien erfüllen [Merkel u. a., 2015]:

- mindestens ein Knoten pro Raum,
- ein Knoten muss mit mindestens einem anderen Knoten verbunden sein,
- Kanten (Verbindungen) dürfen keine Wände schneiden,
- Mehrfachverbindungen zu Knoten müssen möglich sein,
- Kreisverbindungen innerhalb von Räumen sind zu vermeiden.

Vektorbasierte Grundrisse erlauben eine Berechnung von Kanten für das Routing mithilfe geometrischer Beziehungen. Prinzipiell können die zentralen Linien zwischen Wänden als Laufwege unterstellt werden. Eine Berechnung aller zentralen Linien zwischen den Wänden stellt daher eine Möglichkeit zur Berechnung der primären Kanten dar. Die Verfahren zur Berechnung der Kanten und Knoten basieren auf Skelettierungsansätzen.

Das *strightskeleton* beschreibt die Verkleinerung der Raumfläche durch iteratives paralleles Verschieben der Wandlinien um einen definierten Abstand [Merkel u. a., 2015; Haurert u. Sester, 2008]. Die zentralen Punkte der kleinsten Flächen definieren die Knoten und können anschließend in Anlehnung an die zuvor beschriebenen Bedingungen miteinander verbunden werden. Die Schwächen dieses Ansatzes bestehen in der anschließenden Berechnung der Kanten, da die berechneten Knoten nicht an den Eingängen optimiert sind und so die Verbindungen zwischen den Räumen mathematisch nur schwer zu realisieren sind.

Eine weitere Methode ist die *konforme Delaunay Triangulation*, bei der die Raumfläche in Dreiecke unterteilt wird. Die Basis für jedes Dreieck bildet eine Wandlinie. Für jede Dreiecksseite im Raum, der zwei Dreiecke zuzuordnen sind, wird der Mittelpunkt der Linie berechnet [Merkel u. a., 2015; Haurert u. Sester, 2008]. Diese

Mittelpunkte werden anschließend verbunden. Der so generierte Graph benötigt noch manuelle Anpassungen, da sonst der Wechsel der Räume durch die Türen hindurch nicht darstellbar ist.

Beide Verfahren lassen sich verbessern durch die Ergänzung von Türpunkten, die den Übergang zwischen den Räumen unterstützen. Für den automatischen Prozess müssen Türen in den Grundrissdaten detektiert werden können. Liegen CAD-Bezugsdaten vor, so stehen oft auch Metadaten zur Verfügung, auf die zugegriffen werden kann. Eine Umsetzung auf Basis der konformen Delaunay Triangulation zeigt Abb. 6.3. Da die Gebäudestruktur der HCU vergleichsweise komplex ist und die Daten nicht in definierten Raumpolygonen vorliegen, musste hier der Ansatz weiter angepasst werden. Nach der Bestimmung von möglichen Routenpunkten aus der Triangulationsberechnung werden Punkte nahe an Wänden in den jeweiligen Raum hinein verschoben [Merkel u. a., 2015]. Anschließend werden die Punkte unter Berücksichtigung der oben genannten Kriterien miteinander verbunden. Abb. 6.3 zeigt das vollautomatisch generierte Ergebnis, das ohne manuelle Nachbesserung nicht funktioniert. So müssen nicht notwendige Kanten reduziert und nicht berechnete Kanten nachträglich eingefügt werden.



Abb. 6.3: Ergebnis einer automatischen Kantengenerierung für einen Routing-Graphen aus einem vektorbasierten Grundriss des zweiten Obergeschosses der HCU (grün: berechnete Kanten, rot: Grundriss) [Merkel u. a., 2015].

Rasterbasierte Grundrisse erlauben eine Berechnung von Kanten für das Routing mithilfe von Bildverarbeitungsalgorithmen. Ein Ansatz zur Verarbeitung von Ras-

terdaten in Routing-Kanten ist die Erosion. Erosion und Dilatation sind Grundoperationen der morphologischen Bildverarbeitung [Jähne, 2005]. Die Erosion beschreibt die Vergrößerung schwarzer Pixelbereiche in Binärbildern. Dazu werden, je nach gewünschtem Verhalten, Masken (Pixel-Matrizen) definiert, die die Nachbarschaft eines Pixels im Bild abfragen und anschließend eine Farbänderung des Pixels erlauben. Die Erosion kann beim vorliegenden Ansatz dazu verwendet werden, um die schwarzen Wandpixel iterativ auszudehnen. Ein Ausschlusskriterium für die weitere Ausdehnung sind schwarze Nachbarpixel. Je nach Auflösung des Bildes sind unterschiedlich viele Iterationen zur Erosionsberechnung notwendig. Als Zwischenergebnis wird ein Binärbild erzeugt, das lediglich weiße Pixel in der Gangmitte enthält. Diese weißen Pfade repräsentieren mögliche Routing-Kanten. Nach der Invertierung des Bildes ist eine Ableitung von Vektoren aus den Rasterdaten möglich. Die so generierten Kanten und Knoten müssen manuell angepasst werden. Stehen für die Verarbeitung nur Vektordaten zur Verfügung, so kann eine rasterbasierte Verarbeitung durch die Konvertierung der Ausgangsdaten in Rasterdaten erzwungen werden.

Beim vierten Obergeschoss im HCU-Gebäude kommt die Erosion zum Einsatz. Aus dem Grundriss des Stockwerks wurde eine Rastergrafik mit einer Auflösung von 1382 x 725 Pixel erstellt. Insgesamt waren 307 Iterationen erforderlich, um das Ergebnis in Abb. 6.4 zu erzeugen. Die Positionen der Routing-Kanten sind deutlich zu erkennen. Aus dem hier gezeigten Ergebnis muss nach der manuellen Nachbesserung ein Vektorplan abgeleitet werden. Dies kann ebenfalls manuell durch Digitalisierung im CAD oder automatisiert per Linienschätzverfahren erfolgen. Zum Abschluss ist eine Rücktransformation des Pixelkoordinatensystems in das Gebäudekoordinatensystem notwendig.

Das Verfahren auf Basis der Erosion bringt gegenüber dem vektorbasierten Ansatz der Delaunay Triangulation verbesserte Ergebnisse für den automatischen Anteil. Beide Ansätze benötigen zusätzliche manuelle Anpassungen, damit ein Routing-Graph die an ihn gestellten Kriterien erfüllt.

Indoor-Mobile-Mapping

Bei Gebäuden, für die keine Gebäudepläne, Flucht- oder Rettungspläne vorliegen, ist ein Innenraumaufmaß notwendig, damit die Grundrisse erstellt werden können.

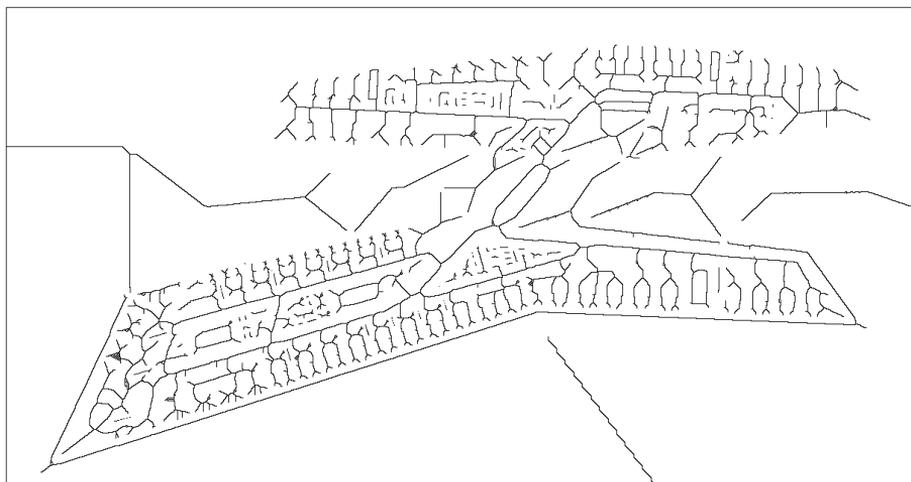


Abb. 6.4: Ergebnis einer automatischen Kantengenerierung für einen Routing-Graphen im vierten Obergeschoss der HCU auf Basis einer rasterbasierten Karte per Erosionsverfahren.

Die messtechnische Erfassung kann beispielsweise durch scannende Systeme erfolgen [Sternberg u. a., 2013]. Bestehende Systeme erlauben Zentimeter-genau eine vergleichsweise schnelle dreidimensionale Erfassung des gesamten Innenraums. Aus den so generierten dreidimensionalen Punktwolken können Grundrisse mithilfe von horizontalen Schnittberechnungen im Rasterformat abgeleitet werden. Ein Konzept zur Umsetzung kinematisch erzeugter Punktwolken zu Gebäudeplänen für die Indoor-Navigation wird in Sternberg u. a. [2013] beschrieben.

6.2 Voraussetzungen und Zusammenspiel der Komponenten

Die Funktionalität einer Indoor-Navigation hängt maßgeblich vom Zusammenspiel der notwendigen Komponenten ab. Die Komponenten können software- oder anwenderbasiert sein:

- softwarebasiert:
 - Positionsschätzung

- Routing
- Gebäudeplan und Raumattribute
- anwenderbasiert:
 - Nutzer
 - Hardware

Die softwarebasierten Elemente müssen echtzeitfähig implementiert sein und die verwendete Navigationshardware ein echtzeitfähiges Ausführen ermöglichen. Abhängig von der Laufgeschwindigkeit soll das Resultat der Positionsschätzung etwa im Sekundentakt visualisiert werden. Allerdings benötigt der Algorithmus zur Positionsschätzung eine höhere Berechnungsgeschwindigkeit, um beispielsweise Schritte und Drehungen zu verarbeiten. Das Routing findet mindestens einmal zu Beginn der Navigation statt. Für die Wegberechnung werden wenige Sekunden benötigt. Die Abweichung des Nutzers von der berechneten Route kann eine Neuberechnung während der Navigation erfordern. Der Akkuverbrauch des Smartphones steigt bei der simultanen Verwendung mehrerer Sensoren und der Darstellung auf dem Display rasant an. Prozessorauslastung und Akkuverbrauch sind entscheidende Kriterien, die bei jeder App-Entwicklung zu beobachten sind.

Für den Übergang von der Indoor- zur Outdoor-Navigation sollte die Umgebungskarte, beispielsweise von Google-Maps, eingebunden werden. Erforderlich ist dazu allerdings eine Transformation der Stockwerkspläne in das globale Koordinatensystem [Wilde, 2015]. Bei Android-basierten Anwendungen liefert die Google-API dafür die Grundlage.

Die notwendige Hardware besteht bei dem hier gewählten Ansatz aus den integrierten Sensoren, einem dreiachsigen Beschleunigungsmesser, einem dreiachsigen Gyroskop und einem Barometer, einer Smartphone-Kamera sowie dem Display. Je nach Modell unterscheiden sich die Sensoren der Smartphones hinsichtlich der Qualität und Messraten und beeinflussen so die Positionsschätzung. Der Algorithmus zur Positionsschätzung muss diese stochastisch unterschiedlichen Eingangsdaten für jedes Smartphone zu einer robusten Position unter Einhaltung der Zielgenauigkeit verarbeiten können.

Die Kamera dient der Initialisierung der relativen Positionsschätzung. Mittels eines QR-Codes (Abb. 6.5), der die eigene Lage im Raum enthält, wird die initiale Position übergeben. Ein QR-Code, der auch bei leichter Beschädigung noch decodierbar ist, kann beispielsweise an Namensschilder von Türen angebracht werden. Da er nur wenige Informationen speichern muss, sind Größen von $20 \times 20 \text{ mm}^2$ ausreichend. Zwar bedeutet die Anbringung eines QR-Codes an Bürotüren einen gewissen Eingriff in die Infrastruktur, der finanzielle Aufwand ist aber relativ unbedeutend. Ein Austausch des QR-Codes ist auch bei Änderungen an der Infrastruktur oder wechselnden Raumnutzern nicht notwendig, da im QR-Code nur Koordinaten und Orientierung vorliegen müssen.



Abb. 6.5: QR-Code zur Initialisierung der Navigation. Anbringung beispielsweise an Namensschildern von Türen. Empfohlene Größe $20 \times 20 \text{ mm}^2$.

Das Nutzerverhalten während der Navigation ist bei INS-Sensoren in Handgeräten ausschlaggebend für die Qualität der Ergebnisse. Von besonderer Bedeutung sind das Laufverhalten und das Halten der Hardware. Daher muss der Schrittzähler sich an die gemessenen Beschleunigungen möglichst automatisch anpassen und Schritte in allen Zuständen robust detektieren. Doch nur eine sachgemäße Bedienung kann negative Einflüsse weitgehend ausschalten. Die Instruktion und Anleitung einer optimalen Nutzung des Smartphones zur Navigation scheinen unumgänglich.

Während der Navigation können Unsicherheiten bei der Positionsschätzung den Nutzer verwirren. Eine gelungene Visualisierung von Umgebung und Navigationsführung ist daher geboten. Eine Visualisierung per Richtungspfeil liefert dem Nutzer weit weniger Informationen als topografische Karten. In Weninger u. a. [2011] wurde dies untersucht, und das Resultat war eindeutig: Personen, die sich verlaufen hatten, haben mithilfe von Karten den richtigen Weg deutlich schneller wiedergefunden. Zur Navigation ist daher auf dem Display ein Kartenausschnitt zu favorisieren, der sich entsprechend des Laufvorgangs kontinuierlich aktualisiert.

6.3 Anwendungen (Apps)

Im Zuge der Entwicklung der MEMS-INS-basierten Positionsschätzung sind zwei Android-basierte Apps mit dem Ziel entstanden, die Teilergebnisse möglichst anwendungsorientiert auf den Testgeräten umzusetzen. Das besondere Augenmerk lag dabei auf der Laufzeit der Komponenten, aber auch auf der Interaktion zwischen Positionsschätzung, Routing und Visualisierung per Smartphone.

MapNavi

Die in dieser Arbeit als *MapNavi* bezeichnete Applikation beinhaltet verschiedene Bausteine einer INS-basierten Navigation. Bei der Entwicklung wurden zunächst Komponenten wie der Schrittzähler und das Pedestrian Dead Reckoning (PDR) sowie eine Positionsschätzung per Kalman Filter (KF) und die Höhenbestimmung auf Basis von Luftdruckmessungen implementiert. Dabei zeigte sich, dass die einzelnen Komponenten ihre Funktionen auf den Testgeräten einwandfrei erfüllen. Besonders für die Positionsschätzung auf Basis von KF und PDR ist die Rechenzeit entscheidend, damit die Positionsschätzung innerhalb eines Schrittes durchgeführt werden kann. Zur Visualisierung wird beim KF und PDR eine Kombination mit dem Grundriss verwendet. Die Initialisierung erfolgt über eine implementierte QR-Code-Erkennung. In Abb. 6.6 wird eine Momentaufnahme der App für die Positionsschätzung auf Basis des KF angezeigt.

Das Display zeigt im Vektorformat einen Ausschnitt der Karte vom ersten Geschoss des ehemaligen D-Gebäudes der HCU. Im Zentrum ist die aktuelle Position durch einen blauen Punkt markiert. Während der Navigation verschiebt und dreht sich die Karte. So ist von der jeweiligen Position stets eine gute Sicht auf den Kartenausschnitt gegeben. Der obere Bereich des Displays stellt Informationen zum Debuggen bereit: die jeweilige Orientierung, die aktuelle Höhe sowie Initialisierungsinformationen vom zuvor fotografierten QR-Code. Der QR-Code enthält in einer Textzeile die x-, y- und z-Koordinate sowie die Orientierung des QR-Codes. Die typische Größe der benutzten QR-Codes beträgt $20 \times 20 \text{ mm}^2$. Bei der *MapNavi* wird bei jedem aufgenommenen QR-Code der KF neu initialisiert und gestartet. Der entsprechende Stockwerksgrundriss wird je nach z-Koordinate automatisch geladen.

6.3 Anwendungen (Apps)

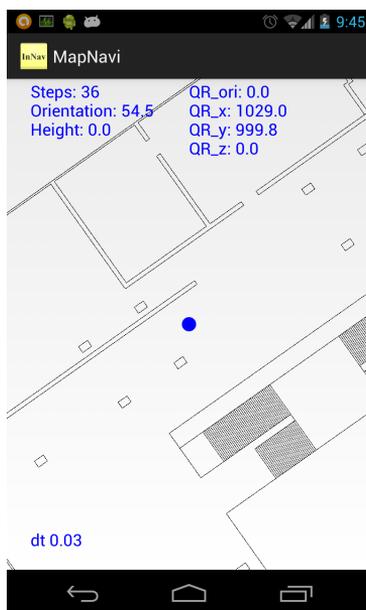


Abb. 6.6: Oberfläche der Applikation MapNavi zur Positionsschätzung auf Basis eines KF. Initialisierung mittels QR-Code. [Willemsen u. a., 2014b].

Für erste Einschätzungen zur benötigten Rechenzeit wird im unteren linken Bereich des Displays ein dt ausgegeben, das die Zeitdifferenz zwischen zwei Berechnungsdurchläufen in Sekunden wiedergibt. Für Vektordaten liegt dieser Wert beim Nexus 4 im Mittel bei 0,03 Sekunden. Bei einer Rasterkarte steigt die Rechenzeit im Mittel auf 0,3 Sekunden für einen Durchlauf. Für die Schritterkennung ist dies insofern problematisch, da der Schrittzähler eine vergleichsweise hohe Abtastrate benötigt. Die App wurde allerdings nicht mit multiplen Threads programmiert, was eine parallele Berechnung von Visualisierung und Positionsschätzung erlaubt hätte.

Unabhängig von der Positionsschätzung ist eine Kalibrieroutine implementiert, die auf Basis eines KF Korrekturen für den Beschleunigungsmesser und den Magnetfeldsensor schätzt. Da für beide Sensoren in gleicher Weise Rohdaten für die Kalibrierung gebraucht werden, kann die Schätzung parallel zur Datengenerierung erfolgen. Da Smartphones der neuesten Bauart keine Kalibrierung mehr benötigen, konnte auf weitere Implementierungen sowie Untersuchungen in dieser Richtung verzichtet werden.

NavCon

Während *MapNavi* besonders zum Testen der Nutzbarkeit der Komponenten zur Positionsschätzung dient, stellt die zweite Software-Entwicklung *NavCon* eine Kombination von Routing, Positionsschätzung und Gebäudeinformation dar. Zur Positionsschätzung nutzt die App den topologischen Ansatz auf Basis des kantenbasierten PDR für den Routing-Graphen. Eingebunden ist die API von GoogleMaps, um einen bruchlosen Übergang von der Indoor- zur GNSS-gestützten Navigation zu gewährleisten.

Abb. 6.7 gibt eine Momentaufnahme der App beim Routing wieder. Wie bei der App *MapNavi* werden im oberen Bereich des Displays Debugging-Informationen zur Orientierung, zum Luftdruck und zur Anzahl der Schritte angezeigt. Die dem Google Maps Plan überlagerten Stockwerkspläne erlauben eine Orientierung im Hinblick auf das Gebäudeumfeld. In der Momentaufnahme ist eine Route abgebildet, die vom aktuellen Stockwerk bis zum Fahrstuhl führt, wo der Stockwerksübergang stattfinden soll. Eine Initialisierung der Positionsschätzung erfolgt wie bei der App *MapNavi* über einen QR-Code.

Die mit der Applikation auf Basis des rein kantenbasierten Ansatzes errechnete Position zeigt, wie erwartet, Schwächen in offenen Bereichen. Eine Berechnung eines Routings benötigt auf dem Nexus 4 zwischen 2 und 4 Sekunden auf Basis des Dijkstra-Algorithmus. Die Kombination von Routing und Positionsschätzung ist mit Unterstützung des QR-Codes vergleichsweise einfach, da so die Startposition der Positionsschätzung und die Startposition des Routings eindeutig sind. Der Startknoten für das Routing wird aus der geringsten Entfernung zum QR-Code ermittelt. Auf jede Initialisierung mit QR-Code erfolgt eine erneute Routenberechnung. Beim aktuellen Ansatz wird auf eine Aktualisierung des Routings während der Bewegung verzichtet, da parallele Routing-Berechnungen den Prozessor belasten und den Akkuverbrauch erhöhen.

Die Aktualisierung der zuvor berechneten Route wird über einen Abgleich mit der aktuellen Position gesteuert. Überschreiten die Abweichungen zur Route einen Grenzwert, so kann das Routing neu berechnet werden. Dies ist aber gekoppelt mit einer Aufforderung zur Initialisierung beim nächsten QR-Code, was zudem Positionsunsicherheiten reduziert.



Abb. 6.7: Oberfläche der App NavCon: Zu sehen ist das Routing im vierten Obergeschoss der HCU.

Die Initialisierung der Positionsschätzung per QR-Code hat gegenüber anderen Schätzverfahren den Vorteil, für ein robustes Routing zu sorgen. Lokale Unsicherheiten in der Positionsschätzung, beispielsweise beim WLAN-Fingerprinting, könnten dazu führen, dass der falsche Startknoten ausgewählt wird. Daher ist selbst bei infrastrukturbasierten Methoden, die lokal vergleichsweise hohe Positionsunsicherheiten aufweisen können, die Nutzung von bekannten Startpositionen sinnvoll.

Struktur der favorisierten Anwendung

Auf Basis der Erfahrungen mit den ersten beiden Apps und den zur Verfügung stehenden Daten wird für die Navigations-App die folgende Struktur gewählt (Abb. 6.8). Die App basiert auf mindestens drei Threads: Routing, Visualisierung und Positionsschätzung. Die Trennung erfolgt, um die rechenintensiven Komponenten

voneinander zu trennen. Die simultane Berechnung erlaubt zudem die Verfügbarkeit der Sensoren zu jedem Zeitpunkt für die Positionsschätzung.

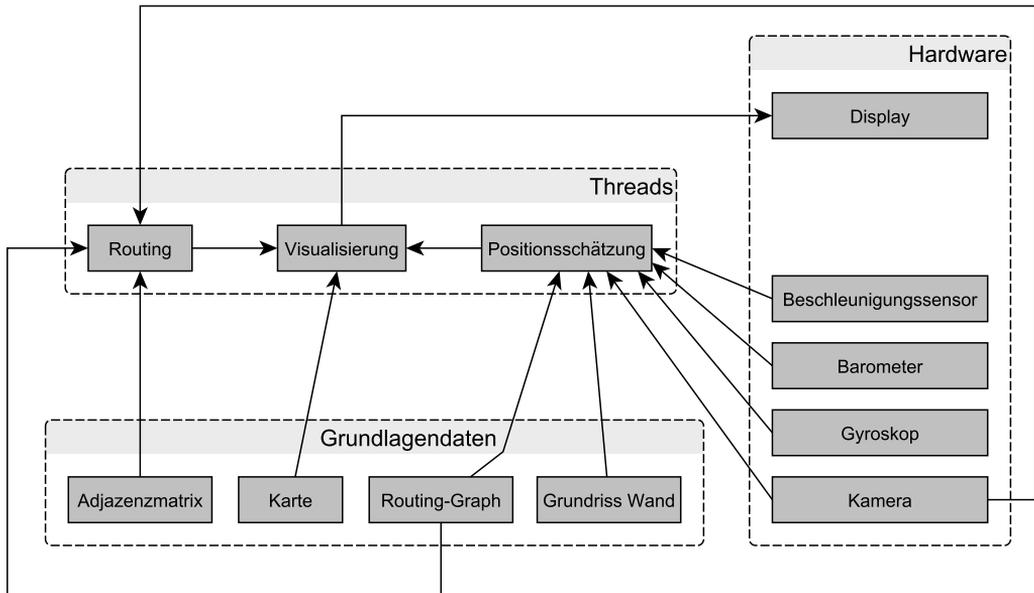


Abb. 6.8: Favorisierte Komponenten einer Navigationsanwendung und die Zuordnung zu Threads in der Programmentwicklung für eine Smartphone-App.

Die für die Navigation notwendigen Sensoren im Smartphone sind Beschleunigungsmesser, Barometer, Gyroskop, Kamera und Display. Als Datengrundlage werden eine Adjazenzmatrix mit den bekannten Knotenkombinationen aus dem Routing-Graphen, Grundrisse der Stockwerke, der geometrische Anteil des Routing-Graphen sowie Gebäudepläne zur Visualisierung benötigt. Abb. 6.8 verdeutlicht das Zusammenspiel dieser Komponenten in der App.

Das Routing kann autark zur Positionsschätzung funktionieren und ist lediglich auf einen QR-Code zur Startposition und auf einen Zielknoten angewiesen. Visualisiert werden die berechnete Route und die aktuelle Position auf der Karte. Die Darstellung auf dem Display wird dabei je nach Positionsänderung angepasst.

Bei der Entwicklung der bisherigen Anwendungen blieb die Implementierung auf verschiedene Smartphone-Modelle weitgehend unbeachtet. Von entscheidender Bedeutung aber sind neben der Verfügbarkeit der betreffenden Sensoren die Auf-

lösung des Displays und die Version des Android-Betriebssystems. Preisgünstige Smartphones verfügen nicht über einen Luftdrucksensor und sind nicht geeignet zur automatisierten Stockwerksunterscheidung. Die Applikation muss für sie eine Ersatzfunktion nutzen, die sich bei einem Smartphone ohne Barometer automatisch aktiviert und beispielsweise durch den Nutzer gesteuert wird.

Während ein Barometer recht simpel per Nutzereingriff zu kompensieren ist, lässt sich ein Gyroskop nicht ersetzen. Smartphones im Niedrigpreissektor verfügen oft nur über ein Gyroskop oder einen Magnetfeldsensor. Da die Richtungsbestimmung aus Magnetfeldsensordaten im Gebäude zu hohen Unsicherheiten führt, eignen sich diese Sensoren allein nur sehr bedingt zur Navigation. Beschleunigungssensoren jedoch, die der App zur Orientierungsbestimmung dienen, gehören heutzutage wegen der Ausrichtung des Displays zur Standardausstattung aller Smartphones.

7 Zusammenfassung und Ausblick

Zusammenfassung

Thema dieser Arbeit sind die Chancen und Grenzen einer autonomen Positionsschätzung, die sich die in modernen Smartphones verbauten MEMS-Sensoren für die Navigation in GNSS-abgeschatteten Bereichen zunutze macht. Mögliche Anwendungsgebiete sind Flughäfen, Großbahnhöfe, Museen und Messegelände sowie andere öffentlich zugängliche Großbauten und Gebäudekomplexe. In der Forschung und Entwicklung sind in den letzten Jahren national und international unterschiedlichste Ansätze dazu entwickelt, erprobt und umgesetzt worden, die sich erheblich in erreichter Positionsgenauigkeit, Kosten, Wartungs- und Implementierungsaufwand unterscheiden.

Infrastrukturbasierte Ansätze bringen in der Regel einen erheblichen Wartungs- und Implementierungsaufwand mit sich. Bildbasierte Navigationsverfahren stellen hohe Ansprüche an die räumliche Umgebung, diese muss beispielsweise gleichmäßig gut ausgeleuchtet sein und darf keine monotonen Strukturen besitzen. Ein MEMS-INS-basiertes Verfahren hat gegenüber diesen beiden Ansätzen den Vorzug, im Prinzip autonom zu arbeiten, indem es sich die im Smartphone verbauten Sensoren für die Fußgängernavigation zu Nutze macht. Aber auch dieser Ansatz weist Schwächen auf. Als problematisch haben sich die mit der Zeit sukzessiv zunehmenden Unsicherheiten der Positionsschätzung erwiesen, die sich durch die Fehlercharakteristika der Inertialsensoren ergeben. Zudem produzieren der individuelle Umgang des Nutzers und die unterschiedlichen Smartphonemodelle weitere Abhängigkeiten in der Positionsschätzung.

Für die Einschätzung einer MEMS-INS-basierten Positionsschätzung hat diese Arbeit die Qualität der Messungen und die Verwendungsweisen der Sensoren exemplarisch anhand der Testgeräte Google Nexus 4 und Samsung Galaxy Nexus untersucht. Dabei wurden folgende Sensoren, die für die Positionsschätzung in Betracht kommen, detailliert untersucht und getestet: Barometer, Beschleunigungsmesser, Gyroskop und Magnetfeldsensor.

Es zeigte sich, dass die Fähigkeit des Barometers zur relativen Luftdruckmessung ideal zur Unterscheidung von Stockwerken in Gebäuden genutzt werden kann.

Bei sämtlichen im Rahmen dieser Arbeit durchgeführten Testreihen ist mithilfe der vom Barometersensor generierten Daten die Stockwerkserkennung einwandfrei und problemlos gelungen. Die Beschleunigungssensoren von Smartphones erlauben die horizontale Ausrichtung des Smartphone-Koordinatensystems und können somit die x-y-Ebene für alle dreiachsigen Sensoren parallel zum Stockwerk ausrichten. Im Laufe der Untersuchung wurde klar, dass sich die Integration der Beschleunigungen zu Positionsveränderungen als Strapdown-Algorithmus aufgrund der hohen Messunsicherheiten der MEMS-Technologie zur Positionsschätzung nicht eignet. Daher wurde der Beschleunigungsmesser zur Schritterkennung genutzt, für den eine Kalibrierung des Sensors unnötig ist. Einzig für die Transformation in den Horizont ist die Kalibrierung des Beschleunigungsmessers entscheidend. Allerdings zeigte sich bei den Testläufen, dass die Schrittbewegungen der Nutzer großen Einfluss auf diese Transformationen haben und sie teilweise erheblich verfälschen können.

Auch das Gyroskop führt bei der Kalibrierung, so zeigte sich bei der Analyse der Rohdaten, nur zu leichten Verbesserungen. Die weitaus größere Beeinträchtigung aber resultierte aus dem sensorinhärenten Rauschen. Es ließ sich durch ein Zero Velocity Update beim Start der Anwendung reduzieren.

Der Magnetfeldsensor erwies sich für einen Einsatz zur Bestimmung der Nordrichtung im Testgebäude als sehr unzuverlässig. Auch eine Kalibrierung verbesserte die Berechnung der Nordrichtung, aufgrund der ferromagnetischen Einflüsse, nicht. Auf einen Einsatz zur Richtungsstützung wurde daher in dieser Arbeit verzichtet.

Im Vergleich der Testgeräte wies das Nexus 4 deutlich geringere Korrekturen bei der Parameterschätzung in den Kalibrierungen auf. Eine Tatsache, die sich vermutlich auf eine implementierte Systemkalibrierung für die MEMS-INS zurückführen lässt. Das könnte als Hinweis dafür gewertet werden, dass bei aktuellen Geräten die Sensoren von signifikanten Einflüssen aufgrund von Maßstab und Offset befreit sind und nur durch den instabilen Anteil des sensorinhärenten Rauschens, zufällige Fehler und Nutzerinteraktionen beeinflusst sind.

Die MEMS-INS-basierte Positionsschätzung beruht auf dem Prinzip der Koppelnavigation, dem Pedestrian Dead Reckoning, bei dem sich jede Ungenauigkeit auf

den folgenden Schritt überträgt. Daher mussten Korrekturen eingeführt werden, die sich, dem Ansatz der Arbeit entsprechend, möglichst infrastrukturunabhängig durchführen lassen sollten. Untersucht wurden Routing-Graphen und Stockwerksgrundrisse für die Positions- und Richtungskorrektur.

Stochastische Filter erlauben eine Positionsschätzung, mit der Korrekturen im Hinblick auf die zu erwartenden Unsicherheiten eingebunden werden können. Kalman Filter und Partikelfilter wurden daher als Ansätze zur Positionsschätzung weiterentwickelt und diskutiert. Im Verlauf der Untersuchung zeigte sich, dass die Unsicherheiten der Sensoren stark durch die unterschiedlichen Verwendungsweisen der Hardware beeinflusst waren, was die Anpassung der Rauschparameter in den Filtern schwierig machte. Daher wurde eine möglichst relative Positionsschätzung entwickelt, die den Routing-Graphen auf Basis von vier Zuständen einbindet. Der Vorteil stochastischer Filter besteht in der Möglichkeit, Positionen frei im Raum bestimmen zu können, was mit dem kantenbasierten Ansatz auf Basis des Routing-Graphen nicht möglich ist. Stochastische Filter erlauben in diesen Ansätzen eine freie Positionsschätzung, die bei ungünstigen Raumkonfigurationen nicht raumgenau Positionen liefert. Durch die Kopplung an Routing-Kanten verspricht dies aber der kantenbasierte Ansatz. Eine entscheidende Schwäche des kantenbasierten Ansatzes aber waren große, offene Räumlichkeiten, in denen die Nutzer von den durch den Routing-Graphen vorgesehenen Laufwegen abwichen und die Positionsschätzung unmöglich machten.

Aus diesen Gründen wurde aus den drei untersuchten Algorithmen eine kombinierte Lösung aus Partikelfilter und kantenbasiertes Pedestrian Dead Reckoning abgeleitet. Das so entwickelte Filter wird in dieser Arbeit als kantenbasiertes Partikelfilter bezeichnet. Das Ziel des Ansatzes besteht in einer relativen, möglichst durch Routing-Kanten gekoppelten Positionsschätzung, die die höheren Unsicherheiten der Sensoren durch den Kantenbezug verringert. Da es im Testgebäude Bereiche ohne Routing-Kanten gibt und die Routing-Kanten die typischen Laufwege der Nutzer nur stark generalisiert widerspiegeln, wurde in das kantenbasierte Partikelfilter das Pedestrian Dead Reckoning implementiert, das besonders in offenen Bereichen eine Entkopplung von den Routing-Kanten erlaubt. Der so entwickelte Ansatz vereint die Vorteile einer kantenbasierten relativen Positionsschätzung

mit der Möglichkeit einer freien Positionsschätzung und soll den Einfluss von fehlerhaften Schrittlängen, nicht detektierten Schritten und stark generalisierten Routing-Graphen weitgehend auffangen.

Die Positionsschätzung im Filter basiert in diesem Ansatz auf Richtungsunterschieden der jeweiligen Partikel zu den umgebenen Routing-Kanten. Die Positionsschätzung, für jedes Partikel an die Partikelrichtung gekoppelt, wurde in der Untersuchung sowohl nach dem kantenbasierten Ansatz als auch dem Pedestrian Dead Reckoning mit Wandstützung fortgeführt. Die größte Herausforderung dabei war die Kombination der topologischen Abfragen mit einem stochastischen Filter. Der hier vorgestellte Ansatz lieferte bei 29 von 42 Testläufen die angestrebte Positionsgenauigkeit. Dass aber bei 13 Datensätzen das Ziel verfehlt wurde, ist ein Beleg dafür, dass das im Zuge der Arbeit entwickelte Filter noch nicht reif für eine Implementierung ist. Das Filter kann im Prinzip durch eine manuelle Anpassung der Parameter für jeden Datensatz bessere Ergebnisse erzielen, diese Parametervariationen konnten aber aktuell keiner Nutzergruppe zugeordnet werden. Die Tests haben weiterhin gezeigt, dass besonders lange, gradlinige Strecken die Positionsschätzung stark verfälschen, da die Unsicherheiten infolge der Addition der angenommenen Schrittlängen nicht durch Kurven nach dem kantenbasierten Ansatz abgefangen werden.

Prinzipiell zeigte sich, dass der kantenbasierte Partikelfilter bei einer Positionsschätzung im Gebäude funktionieren kann, aber, je nach Route, weitere Korrekturen benötigt. Eine Verbesserung kann beispielsweise durch die Eingabe einer personenbezogenen Schrittlänge im Filter erfolgen. Ferner bieten sich die in der Arbeit diskutierten infrastrukturbasierten Verfahren an. Sie ließen sich mit geringem Aufwand implementieren, da sie nur Teilinformationen zur Korrektur im Filter bereitstellen müssen, beispielsweise die Distanzmessung zu einem Access Point. Das WLAN-Fingerprinting konnte als Verfahren ausgeschlossen werden, da dessen Unsicherheiten in offenen Bereichen zu gravierend waren und die per Grundriss und Routing-Graph korrigierte MEMS-INS-basierte Anwendung gerade hier weitere Korrekturen erforderlich gemacht hätte.

Gegenüber infrastrukturbasierten Verfahren, die Funktechniken einsetzen, müssen bei INS-basierten Verfahren die Sensoren – mithin das Smartphone – in ei-

ner stabilen Ausrichtung zum Nutzer getragen werden. Gegenüber *food-mounted* Systemen, die eine fixierte Trageweise gewährleisten, wird in dieser Arbeit die *hand-held* Trageweise bevorzugt. Dies hat sich insofern als Nachteil erwiesen, da die Fehleranfälligkeit der Positionsschätzung in Abhängigkeit vom Nutzerverhalten ansteigt. Für eine Kombination von Indoor- und Outdoor-Navigation sowie für die optimale Bedienerfreundlichkeit ist diese Trageweise allerdings unerlässlich. Eine Zustandserkennung für die Ausrichtung des Smartphones während der Bewegung könnte helfen die Bedienerfreundlichkeit und Zuverlässigkeit der Positionsschätzung weiter zu steigern.

Die Komponenten einer Navigationsanwendung sind das Routing, die Visualisierung mit Zielführung und die Positionsschätzung. Im Hinblick auf eine weitere Reduzierung des Implementierungsaufwands ist die möglichst automatische Erstellung der einzelnen Komponenten – je nach Verfügbarkeit der Basisdaten des Gebäudes – notwendig. Hierfür wurden in dieser Arbeit Ansätze zur halbautomatischen Erzeugung von Grundriss- und Routing-Daten beschrieben. Grundrisse lassen sich erfolgreich aus Flucht- und Rettungsplänen ableiten und für den Routing-Graphen ist besonders die Erosion ein hilfreiches Werkzeug zur Erzeugung von Kanten. Die Resultate beider Verfahren benötigen manuelle Nachbereitung, um brauchbare Grundrisse oder Routing-Graphen abzuleiten. Liegen keine Informationen vor, bietet sich eine dreidimensionale Erfassung der Infrastruktur auf Basis von Mobile-Mapping-Systemen an, aus deren Daten sich Grundrisse und letztendlich auch Routing-Graphen ableiten lassen.

Der Aufbau einer für jedes Smartphonemodell nutzbaren Anwendung ist mit großem Aufwand verbunden, da beispielsweise nicht verfügbare Sensoren bei der Implementierung berücksichtigt werden müssen. Zudem unterliegt die Anwendung den Besonderheiten des jeweiligen Betriebssystems. Die Struktur einer App sollte eine Trennung von Visualisierung, Routing und Positionsschätzung erlauben, damit die rechenintensiven Anteile – also das Routing und die Visualisierung – die Positionsschätzung nicht beeinträchtigen und parallel durchgeführt werden können.

Ausblick

Der kantenbasierte Partikelfilter stellt ein interessantes Werkzeug zur autonomen Positionsschätzung dar. Allerdings kann auch er die aktuellen Schwächen nicht gänzlich reduzieren. In den Ergebnissen der Stichprobe fallen besonders Unsicherheiten bei langen gradlinigen Strecken auf, die sich aus unterschiedlichen Schrittlängen ergeben. Notwendig wäre also eine Erweiterung des Filters zur besseren Anpassung der Positionsschätzung an das individuelle Verhalten der Nutzer.

Ein weiteres Feld für zukünftige Arbeiten stellt die Optimierung der Stellgrößen im Filter dar. Das Messrauschen für die Gewichtung der Korrekturen sowie der Grenzwert für die Kopplung des Partikels an Routing-Kanten können anhand einer repräsentativen Stichprobe auf breiterer Datenbasis untersucht werden. So könnten die Stellgrößen an bestimmte Nutzergruppen angepasst und die Ergebnisse insgesamt verbessert werden. Zudem sind Modifikationen des Partikelfilters denkbar – beispielsweise Variationen der Partikelanzahl, wie es beim Regularized Partikelfilter gemacht wird, um die Performance insbesondere bei schmalen Gängen zu verbessern.

Im Zuge des technischen Fortschritts sind neue und verbesserte Sensorgenerationen zu erwarten, die auch für die MEMS-basierte Positionsschätzung positive Folgen haben dürften. Es ist davon auszugehen, dass die Nutzung von Inertialsensoren in Smartphones für die Fußgängernavigation zunehmend auch von den Herstellern unterstützt wird. Dennoch wird weiterhin bei MEMS-INS eine Stützung der Positionsschätzung erforderlich sein, wie dies die vorliegende Arbeit anhand der Fehlercharakteristika eindeutig nachgewiesen hat.

Aktuell werden fast ausschließlich infrastrukturbasierte Systeme für die Indoornavigation genutzt. Es ist davon auszugehen, dass sich in Zukunft der Aufwand an Implementierungsmaßnahmen für infrastrukturbasierte Anwendungen durch die Stützung von MEMS-INS verringert. Eine ausschließlich auf MEMS-INS-basierte Positionsschätzung ist zwar, wie diese Arbeit zeigt, prinzipiell möglich, der Einfluss durch den Nutzer ist allerdings bei *hand-held* Systemen schwer kontrollierbar. Zukünftige Arbeiten zur MEMS-INS-geführten Navigation sollten daher in erster Linie auf die Bedienerfreundlichkeit abzielen, ohne die Genauigkeit der Positionsschätzung zu beeinträchtigen.

Literatur

- [Aggarwal u. a. 2010] AGGARWAL, P. ; SYED, Z. ; NOURELDIN, A. ; EL-SHEIMY, N.: *MEMS-Based Integrated Navigation (GNSS Technology and Applications)*. Artech House Inc, ISBN: 978-1-60807-043-5, 2010
- [AKM 2016] AKM: *AK8963 3-axis Electronic Compass*. <http://www.akm.com/akm/en/file/datasheet/AK8963.pdf>. Version: 2016. – letzter Zugriffszeitpunkt: 22.01.2016
- [Alyafawi u. a. 2015] ALYAFAWI, I. ; NIKAEIN, N. ; BRAUN, T.: *Towards Real-Time Network-Based Positioning in LTE*. Proceedings of the 1st GI Expert Talk on Localization (RWTH Aachen - Department of Computer Science - Technical Report), 2015
- [Arduino-Guide 2015] ARDUINO-GUIDE: *Getting Started with Arduino*. <http://www.arduino.cc/en/Guide/HomePage>. Version: 2015. – letzter Zugriffszeitpunkt: 22.01.2016
- [Artese 2008] ARTESE, Trecroc A. G.: *Calibration of a low cost mems ins sensor for an integrated navigation system*. XXIst ISPRS Congress Technical Commission V, Beijing, China, 2008
- [Arulampalam u. a. 2002] ARULAMPALAM, M.S. ; MASKELL, S. ; GORDON, N. ; CLAPP, T.: *A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking*. IEEE Transactions on Signal Processing, VOL. 50, NO. 2, 2002
- [Ascher 2014] ASCHER, C.: *Modulares multisensorielles Indoor Navigationssystem*. Logos Verlag, ISBN: 978-3-8325-3669-5, 2014
- [Blankenbach 2007] BLANKENBACH, J.: *Handbuch der mobilen Geoinformation*. Wichmann Heidelberg, ISBN: 978-3-87907-455-6, 2007
- [Blankenbach 2009] BLANKENBACH, J.: *Robuster räumlicher Bogenschnitt - Ein Ansatz zur robusten Positionsberechnung in Indoor-Szenarien*. Allgemeine Vermessungsnachrichten (AVN), Heft 8-9/09, 2009
- [Blankenbach u. a. 2011] BLANKENBACH, J. ; NORRDINE, A. ; HELLMERS, H. ; GASPARIAN, E.: *A Novel Magnetic Indoor Positioning System for Indoor Loca-*

- tion Services. In: *Proceedings of 8th International Symposium on Location-Based Services*, 2011
- [Blumenthal u. a. 2007] BLUMENTHAL, J. ; GROSSMANN, R. ; GOLATOWSKI, F. ; TIMMERMANN, D.: *Weighted Centroid Localization in Zigbee-based Sensor Networks*. Conference Intelligent Signal Processing, Alcalá de Henares (Spanien), ISBN: 978-1-4244-0829-0, 2007
- [Bylemans u. a. 2009] BYLEMANS, I. ; WEYN, M. ; KLEPAL, M.: *Mobile Phone-based Displacement Estimation for Opportunistic Localisation Systems*. Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, 2009
- [Cai u. a. 2016] CAI, Y. ; ZHAO, Y. ; DING, X. ; FENELLY, J.: *Magnetometer basics for mobile phone applications*. http://www.memsic.com/userfiles/files/publications/Articles/Electronic_Products_Feb_%202012_Magnetometer.pdf. Version: 2016. – letzter Zugriffszeitpunkt: 22.01.2016
- [Dalladas u. a. 2014] DALLADAS, O. ; ROMANOVA, S. ; WEITER, J. ; DEGGIM, S.: *Umsetzung eines Routings im Neubau der HCU*. 2014. – Projektarbeit an der HafenCity Universität, unveröffentlicht, 2014
- [Ehlers u. Schiewe 2012] EHLERS, M. ; SCHIEWE, J.: *Geoinformatik*. Wissenschaftliche Buchgesellschaft Darmstadt, ISBN: 978-3-534-23526-1, 2012
- [Fu u. Retscher 2009] FU, Q. ; RETSCHER, G.: *Using RFID and INS for Indoor Positioning*. Geoinformation and Cartography, 2009. – 421–438 S.
- [GFZ 2012] GFZ, Geoforschungszentrum: *GFZ Magnetfeldrechner*. <http://www-app3.gfz-potsdam.de/Declinationcalc/declinationcalc.html>. Version: 2012. – letzter Zugriffszeitpunkt: 22.01.2016
- [GPS-Repeater 2016] GPS-REPEATER: *Aucon GPS-Repeater*. <http://gps-repeater.de/de/produkte/gps-repeater.html>. Version: 2016. – letzter Zugriffszeitpunkt: 22.01.2016
- [Gruber u. Joeckel 2004] GRUBER, F. J. ; JOECKEL, R.: *Formelsammlung für das Vermessungswesen*. B. G. Teubner Verlag, ISBN: 3-519-00486-0, 2004

-
- [Gupta u. a. 2014] GUPTA, S. K. ; BOX, S. ; WILSON, R.E.: *Low cost infrastructure free form of indoor positioning*. International Conference on Indoor Positioning and Indoor Navigation (IPIN), Busan (Südkorea), 2014
- [Hansson u. Tufvesson 2011] HANSSON, A. ; TUFVESSON, L.: *Using Sensor Equipped Smartphones to Localize WiFi Access Points*. Department of Automatic Control, Lund University, 2011
- [Haunert u. Sester 2008] HAUNERT, J.-H. ; SESTER, M.: *Area Collapse and Road Centerlines based on Straight Skeletons*. Journal GeoInformatica, 2008
- [heise.de 2016] HEISE.DE: *Datenschützer: Street-View-Autos scannen private Funknetze [Update]*. <http://www.heise.de/newsticker/meldung/Datenschuetzer-Street-View-Autos-scannen-private-Funknetze-Update-984118.html>.
Version: 2016. – letzter Zugriffszeitpunkt: 22.01.2016
- [Hellmers u. a. 2013] HELLMERS, H. ; NORRDINE, A. ; BLANKENBACH, J. ; EICHORN, A.: *An IMU/Magnetometer-based Indoor Positioning System using Kalman Filtering*. International Conference on Indoor Positioning and Indoor Navigation (IPIN), Montbeliard (Frankreich), 2013
- [Händler 2012] HÄNDLER, V.: *Konzeption eines bildbasierten Sensorsystems zur 3D-Indoorpositionierung sowie Analyse möglicher Anwendungen*. Dissertation an der TU Darmstadt, ISBN: 978-3-935631-25-9, 2012
- [IDC 2013] IDC: *International Data Corporation (IDC)*. <https://www.idc.com/getdoc.jsp?containerId=prUS23771812>. Version: 2013. – letzter Zugriffszeitpunkt: 02.05.2013
- [IIS 2015] IIS, Fraunhofer: *awiloc*. <http://www.iis.fraunhofer.de/de/ff/lok/tech/feldstaerke/rssi/tl.html>. Version: 2015. – letzter Zugriffszeitpunkt: 22.01.2016
- [iMAR 2010] iMAR, GmbH: *Configuration and Usage - Handbuch für iNAV-RQH-1003*. 2010. – unveröffentlicht
- [indoornavigation.de 2014] INDOORNAVIGATION.DE: *iBeacon*. <http://www.indoornavigation.de/ibeacon>. Version: 2014. – letzter Zugriffszeitpunkt: 22.01.2016
-

- [InvenSense 2015] INVENSENSE: *MPU-6000 and MPU-6050 Product Specification Revision 3.4*. http://store.invensense.com/datasheets/invensense/MPU-6050_DataSheet_V3%204.pdf. Version: 2015. – letzter Zugriffszeitpunkt: 14.05.2015
- [Jähne 2005] JÄHNE, B.: *Digitale Bildverarbeitung*. 6., überarbeitete und erweiterte Auflage. Springer Verlag Berlin, ISBN: 978-3540249993, 2005
- [Johnke 2012] JOHNKE, S.: *Indoor-Positionierung für mobile Endgeräte*. Akademiker Verlag Saarbrücken, ISBN: 978-3-639-45908-1, 2012
- [Kahmen 2006] KAHMEN, Heribert: *Angewandte Geodäsie - Vermessungskunde 20. Auflage*. Walter de Gruyter Verlag, 2006 (978-3-11-018464-8)
- [Kalman 1960] KALMAN, R. E.: *A New Approach to Linear Filtering and Prediction Problems*. Bd. 82. ASME Journal of Basic Engineering, 1960. – 35–45 S.
- [Keller u. Willemsen 2014] KELLER, F. ; WILLEMSSEN, T.: *Arduino in der Geodäsie? Geodäsie quo vadis - Möglichkeiten und Grenzen kostengünstiger Messverfahrens-Lösungen*, 2014. – Workshop mit Präsentation, 2014
- [Klingbeil 2006] KLINGBEIL, L.: *Entwicklung eines modularen und skalierbaren Sensorsystems zur Erfassung von Position und Orientierung bewegter Objekte*. Universitäts- und Landesbibliothek Bonn, 2006 <http://hss.ulb.uni-bonn.de/2006/0715/0715.htm>
- [Klingbeil u. a. 2010a] KLINGBEIL, L. ; REINER, R. ; ROMANOVAS, M. ; TRAECHTLER, M. ; MANOLI, Y.: *Multi-modal sensor data and information fusion for localization in indoor environments*. Positioning Navigation and Communication (WPNC), Dresden, 2010. – 187–192 S.
- [Klingbeil u. Romanovas 2012] KLINGBEIL, L. ; ROMANOVAS, M.: *Rekursive Bayesische Schätzverfahren zur Lokalisierung von Personen innerhalb von Gebäuden*. Bd. 137. Jahrgang. Zeitschrift für Vermessungswesen (zfv 6/2012), 2012. – 365–376 S.
- [Klingbeil u. a. 2010b] KLINGBEIL, L. ; ROMANOVAS, M. ; SCHNEIDER, P. ; TRAECHTLER, M. ; MANOLI, Y.: *A Modular and Mobile System for Indoor*

-
- Localization*. International Conference on Indoor Positioning and Indoor Navigation (IPIN), Zürich (Schweiz), 2010
- [Künneht 2012] KÜNNETH, T.: *Android 4: Apps entwickeln mit dem Android SDK*. Galileo Computing, ISBN: 978-3-8362-1948-8, 2012
- [Kupke u. a. 2016] KUPKE, J. ; WILLEMSSEN, T. ; KELLER, F. ; STERNBERG, H.: *Development of a step counter based on artificial neural networks*. Journal of Location Based Services, Taylor Francis Online, 2016 <http://www.tandfonline.com/doi/full/10.1080/17489725.2016.1196832>
- [Lederer 2009] LEDERER, T.: *Entwicklung eines Navigationssystems - Optimierung von Routing und Visualisierung*. VDM Verlag Dr. Müller Saarbrücken, ISBN: 978-3-639-17157-0, 2009
- [Leitz u. a. 2015] LEITZ, S. ; KELLER, F. ; WILLEMSSEN, T. ; STERNBERG, H. ; KAGERAH, S.: *Development of a WiFi-Fingerprinting for position determination by mean of probabilistic methods*. In: *FIG Working Week in Sofia (Bulgarien)*, 2015
- [LG 2016] LG: *Technische Daten Nexus 4*. <http://www.lg.com/de/handy/lg-E960-Nexus-4>. Version: 2016. – letzter Zugriffszeitpunkt: 22.01.2016
- [Liu u. a. 2010] LIU, T. ; CARLBERG, M. ; CHEN, G. ; CHEN, J. ; KUA, J. ; ZAKHOR, A.: *Indoor Localization and Visualization using a Human-Operated Backpack System*. International Conference on Indoor Positioning and Indoor Navigation (IPIN), Zürich (Schweiz), 2010. – 1–10 S.
- [Lukianto u. a. 2010] LUKIANTO, C. ; HÖNNIGER, C. ; STERNBERG, H.: *Pedestrian Smartphone-based Indoor Navigation Using Ultra Portable Sensory Equipment*. International Conference on Indoor Positioning and Indoor Navigation (IPIN), Zürich (Schweiz), 2010. – 1–5 S.
- [Lukianto u. Sternberg 2011a] LUKIANTO, C. ; STERNBERG, H.: *Overview of Current Indoor Navigation Techniques and Implementation Studies*. FIG Working Week - Bridging the Gap between Cultures, Marrakesch (Marokko), 2011
- [Lukianto u. Sternberg 2011b] LUKIANTO, C. ; STERNBERG, H.: *STEPPING - Smartphone-based Portable Pedestrian Indoor Navigation*. Proceedings of
-

- the 7th International Symposium on Mobile Mapping Technology (MMT2011), Krakau (Polen), 2011
- [Marchetti u. a. 2009] MARCHETTI, L. ; GRISSETTI, G. ; IOCCHI, L.: *A Comparative Analysis of Particle Filter Based Localization Methods*. RoboCup 2006: Robot Soccer World Cup X, ser. Lecture Notes in Computer Science, Bremen, 2009. – 442–449 S.
- [Margolis 2012] MARGOLIS, M.: *Arduino Kochbuch*. O'Reilly Verlag Köln, ISBN: 978-3-86899-353-0, 2012
- [Marouane u. Ebert 2015] MAROUANE, C. ; EBERT, A.: *Enabling Pedometers on Basis of Visual Feature Point Conversion*. Aachener Informatik-Berichte, 2015
- [Mautz 2012] MAUTZ, R.: *Indoor Positioning Technologies - Application for Venia Legendi in Positioning and Engineering Geodesy*. Habilitation der ETH Zurich, 2012
- [McComp 2013] MCCOMP, G.: *Arduino - Robot Bonanza*. McGraw-Hill Verlag, New York, ISBN: 978-0-07-178277-7, 2013
- [Merkel u. a. 2015] MERKEL, T. ; GORENFLO, T. ; SCHRÖTER, E. ; VON DELFT, C. ; WITTKKE, S.: *Entwicklung eines allgemeingültigen Kartenproduktes aus CAD Daten für die Indoornavigation*. 2015. – Projektarbeit an der HafenCity Universität, unveröffentlicht, 2015
- [Müller 2007] MÜLLER, S.: *Monte Carlo-Methoden - Angewandt in der Roboterlokalisierung*, Universität Stuttgart, Diplomarbeit, 2007. <http://www.isa.uni-stuttgart.de/LstStoch/Mueller/diplomarbeit.pdf>. – letzter Zugriffszeitpunkt: 22.01.2016
- [Mok u. a. 2012] MOK, E. ; LAU, R. ; XIA, L. ; RETSCHER, G. ; TIAN, H.: *Influential factors for decimetre level positioning using ultra wide band technology*. Survey Review, Maney Online, Volume 44, 2012
- [Neumann 2002] NEUMANN, D.: *Kalman-Filter und Partikelfilter zur Selbstlokalisierung - Ein Vergleich*. <http://www.allpsych.uni-giessen.de/dir/projects/particle.pdf>. Version: 2002. – letzter Zugriffszeitpunkt: 22.01.2016

-
- [Norrline u. a. 2013] NORRDINE, A. ; GRIMM, D. ; BLANKENBACH, J. ; WIESER, A.: *An UWB Based Indoor Compass for Accurate Heading Estimation in Buildings*. International Conference on Indoor Positioning and Indoor Navigation (IPIN), Montbeliard (Frankreich), 2013
- [Peter u. a. 2012] PETER, M. ; FRITSCH, D. ; SCHÄFER, B. ; KLEUSBERG, A. ; LINK, J.A.B. ; WEHRLE, K.: *Versatile Geo-referenced Maps for Indoor Navigation of Pedestrians*. International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sydney (Australien), 2012
- [Poor u. a. 2009] POOR, S. M. ; GREJNER-BRZEZINSKA, D. A. ; TOTH, C. K. ; RIZOS, C.: *Quality assurance / quality control analysis of dead reckoning parameters in a personal navigator*. Geoinformation and Cartography, 2009. – 333–351 S.
- [Pratama u. a. 2012] PRATAMA, A. R. ; WIDYAWAN ; HIDAYAT, R.: *Smartphone-based Pedestrian Dead Reckoning as an Indoor Positioning System*. International Conference on System Engineering and Technology (ICSET), Bandung (Indonesien), 2012
- [PTB 2007] PTB, Physikalisch Technische B.: *PTB gravity field*. <http://www.ptb.de/de/org/1/11/115/doc/tabelle.pdf>. Version: 2007. – letzter Zugriffszeitpunkt: 22.01.2016
- [Qian u. a. 2013] QIAN, J. ; MA, J. ; YING, R. ; LIU, P. ; PEI, L.: *An Improved Indoor Localization Method Using Smartphone Inertial Sensors*. International Conference on Indoor Positioning and Indoor Navigation (IPIN), Montbeliard (Frankreich), 2013
- [Real Ehrlich u. Blankenbach 2014] REAL EHRlich, C. ; BLANKENBACH, J.: *Innenraumpositionierung für Fußgänger unter Verwendung eines Smartphones*. Geoinformationssysteme 2014 - Münchener GI-Runde, 2014
- [Renaudin u. a. 2013] RENAUDIN, V. ; DEMEULE, V. ; ORTIZ, M.: *Adaptive pedestrian displacement estimation with a smartphone*. International Conference on Indoor Positioning and Indoor Navigation (IPIN), Montbeliard (Frankreich), 2013
- [Romanovas u. a. 2013] ROMANOVAS, M. ; GORIDKO, V. ; KLINGBEIL, L. ; BOURUAH, M. ; AL-JAWAD, A. ; TRAECHTLER, M. ; MANOLI, Y.: *Pedestrian*
-

Indoor Localization Using Foot Mounted Inertial Sensors in Combination with a Magnetometer, a Barometer and RFID. Location Based Services in Geoinformation and Cartography, Springer Verlag, Berlin, 2013. – 151–172 S.

[von Rosenberg 2006] VON ROSENBERG, H.: *Sensorfusion zur Navigation eines Fahrzeugs mit low-cost Inertialsensorik*, Universität Stuttgart, Diplomarbeit, 2006

[Samsung 2011] SAMSUNG: *Samsung technische Daten Galaxy Nexus.* <http://www.samsung.com/de/consumer/mobile-device/mobilephones/archive-mobile-phones/GT-I9250TSADBT-spec>. Version: 2011. – letzter Zugriffszeitpunkt: 02.08.2013

[Schmidt 2012] SCHMIDT, M.: *Arduino - Ein schneller Einstieg in die Microcontroller-Entwicklung.* dpunkt.verlag Heidelberg, 2012

[Schwarzlmüller 2007] SCHWARZLMÜLLER, C.: *Analyse der kamera-basierten SLAM - Bibliothek SceneLib und Erweiterung für die Roboterlokalisierung bei einer unbeschränkten Anzahl von visuellen Features*, Alpen-Adria-Universität Klagenfurt, Diplomarbeit, 2007. <http://vi.uni-klu.ac.at/publications/master/master-schwarzlmueeller.pdf>. – letzter Zugriffszeitpunkt: 22.01.2016

[Sensortec 2016] SENSORTEC, Bosch: *Data sheet BMP180 Digital pressure sensor.* <http://ae-bst.resource.bosch.com/media/products/dokumente/bmp180/BST-BMP180-DS000-12~1.pdf>. Version: 2016. – letzter Zugriffszeitpunkt: 22.01.2016

[Sternberg 2000] STERNBERG, H.: *Zur Bestimmung der Trajektorie von Landfahrzeugen mit einem hybriden Messsystem.* Schriftenreihe der Universität der Bundeswehr München, Neubiberg, 2000

[Sternberg u. a. 2009] STERNBERG, H. ; FESSELE, M. ; HÖNNIGER, C.: *Indoor Navigation Without Infrastructure-based Local Positioning System.* Mobile Mapping Technology (MMT2009), Sao Paulo (Brasilien), 2009

[Sternberg u. a. 2010] STERNBERG, H. ; HÖNNIGER, C. ; FESSELE, M.: *Indoor Navigation ohne Verwendung von Infrastruktursensor.* 16. Internationaler Ingenieurvermessungskurs, München, 2010

-
- [Sternberg u. a. 2012] STERNBERG, H. ; KELLER, F. ; WILLEMSSEN, T.: *Calibration of Smartphones for the use in indoor navigation*. International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sydney (Australien), 2012
- [Sternberg u. a. 2013] STERNBERG, H. ; KELLER, F. ; WILLEMSSEN, T.: *Precise indoor mapping as a basis for coarse indoor navigation*. Bd. 7-4. Journal of Applied Geodesy, de Gruyter Verlag, 2013. – 227–308 S.
- [Sternberg u. Schwalm 2007] STERNBERG, H. ; SCHWALM, C.: *Qualification Process for MEMS Gyroscopes for the Use in Navigation Systems*. Mobile Mapping Technologie (MMT07), Padua (Italien), 2007
- [Sternberg u. a. 2015] STERNBERG, H. ; WILLEMSSEN, T. ; KELLER, F.: Einsatz von MEMS Inertial-Sensoren für die Orientierung in Gebäuden. In: *145. DVW-Seminar Interdisziplinäre Messaufgaben im Bauwesen*, 2015
- [Turau 2009] TURAU, V.: *Algorithmische Graphentheorie*. 3. Auflage, Oldenbourg Verlag München, 2009
- [Wang 2012] WANG, Q.: *Entwicklung eines MEMS Drucksensor-Prozessmoduls für die Post-CMOS Integration*. Dissertation an der Universität Duisburg-Essen, 2012
- [Wendel 2007] WENDEL, J.: *Integrierte Navigationssysteme - Sensordatenfusion, GPS und Inertiale Navigation*. Oldenbourg Verlag München, ISBN: 978-3-486-58160-7, 2007
- [Wendel 2011] WENDEL, J.: *Integrierte Navigationssysteme - Sensordatenfusion, GPS und Inertiale Navigation*. 2. Auflage, Oldenbourg Verlag München, ISBN: 978-3-486-70439-6, 2011
- [Weninger u. a. 2011] WENINGER, B. ; BOUCHAIN, J. ; LUKIANTO, C. ; STERNBERG, H.: *Smartphone-Based Portable Pedestrian Indoor Navigation: Experiments on Visualisation and Positioning*. Gartner, G. und Ortog, F. (eds.) Digital Proceedings of the 8th International Symposium on Location-Based Services (LBS), Wien, 2011
- [Widyawan u. a. 2008] WIDYAWAN ; KLEPAL, M. ; BEAUREGARD, S.: *A Backtracking Particle Filter for fusing building plans with PDR displacement estimates*. Positioning, Navigation and Communication (WPNC), Hannover, 2008. – 207 – 212 S.
-

- [Wild-Pfeiffer u. Schäfer 2011] WILD-PFEIFFER, F. ; SCHÄFER, B.: *MEMS-Sensoren, auch für die Geodäsie*. Bd. 1. Zeitschrift für Vermessungswesen (zfv 1/2011), 2011. – 30–39 S.
- [Wilde 2015] WILDE, K.: *Untersuchungen zur Umsetzung einer autonomen Smartphone basierten Innenraumnavigation*. 2015. – Masterarbeit an der HafenCity Universität, unveröffentlicht, 2015
- [Wilde u. a. 2014] WILDE, K. ; SCHLEGEL, I. ; LIN, C. W. ; NIPPGEN, P.: *Umsetzung eines Routings im HCU-Gebäude*. 2014. – Projektarbeit an der HafenCity Universität, unveröffentlicht, 2014
- [Willemsen u. a. 2013] WILLEMSSEN, T. ; KELLER, F. ; STERNBERG, H.: *Untersuchungen zur Indoornavigation mittels aktueller Smartphonetechnologie*. Beiträge der Oldenburger 3D-Tage, Wichmann Verlag, 2013
- [Willemsen u. a. 2014a] WILLEMSSEN, T. ; KELLER, F. ; STERNBERG, H.: *Concept for building a MEMS based Indoor Localization system*. International Conference on Indoor Positioning and Indoor Navigation (IPIN), Busan (Südkorea), 2014
- [Willemsen u. a. 2014b] WILLEMSSEN, T. ; KELLER, F. ; STERNBERG, H.: *Concept for building a Smartphone based Indoor Localization System*. In: *Fusion2014, Salamanca (Spanien)*, 2014
- [Willemsen u. a. 2014c] WILLEMSSEN, T. ; KELLER, F. ; STERNBERG, H.: *Fusionierung von im Smartphone verbauten Sensoren zur Innenraumpositionierung - Ein Ansatz mit Kalman Filter*. Beiträge der Oldenburger 3D-Tage, Wichmann Verlag, 2014
- [Willemsen u. a. 2015a] WILLEMSSEN, T. ; KELLER, F. ; STERNBERG, H.: *Kartengestützte MEMS-basierte Indoorpositionierung mittels Partikelfilter*. Beiträge der Oldenburger 3D-Tage, Wichmann Verlag, 2015
- [Willemsen u. a. 2015b] WILLEMSSEN, T. ; KELLER, F. ; STERNBERG, H.: *A Topological Approach with MEMS in Smartphones based on Routing-Graph*. International Conference on Indoor Positioning and Indoor Navigation (IPIN), Banff (Kanada), 2015

- [Willert u. a. 2011] WILLERT, V.E. ; WILLERT, V.V. ; GERING, S. ; RASS, S. ; ETZEL, J.: *Automated extraction of image coordinates for Optical Indoor Positioning*. International Conference on Indoor Positioning and Indoor Navigation (IPIN), Guimaraes (Portugal), 2011
- [www.ifixit.com 2015] WWW.IFIXIT.COM: *Teardown*. <https://www.ifixit.com/Teardown/Nexus+4+Teardown/11781><https://www.ifixit.com/Teardown/Samsung+Galaxy+Nexus+Teardown/7182>. Version: 2015. – letzter Zugriffszeitpunkt: 14.05.2015
- [Xsens 2013] XSENS: *MTi-G User Manual and Technical Documentation*. [//www.xsens.com/wp-content/uploads/2013/11/MTi-G_User_Manual_and_Technical_Documentation.pdf](http://www.xsens.com/wp-content/uploads/2013/11/MTi-G_User_Manual_and_Technical_Documentation.pdf). Version: 2013. – letzter Zugriffszeitpunkt: 22.01.2016
- [Yamaha 2016] YAMAHA: *YAS530 MS-3E Magnetic Field Sensor Type 3E*. http://download.yamaha.com/api/asset/file/?language=ja&site=jp.yamaha.com&asset_id=46264. Version: 2016. – letzter Zugriffszeitpunkt: 22.01.2016
- [Zekavat u. Buehrer 2012] ZEKAVAT, S. A. ; BUEHRER, R. M.: *Handbook of Position Location: Theory, Practice and Advances*. JohnWiley & Sons, New Jersey, ISBN: 978-0-470-94342-7, 2012

Abbildungsverzeichnis

1.1	Konzept zur Umsetzung eines Indoor-Navigationssystems [Willem- sen u. a., 2014b]	6
2.1	Low Cost Inertial Messsystem MTiG von Xsens [Xsens, 2013].	11
2.2	Platine mit MEMS-Sensoren auf roten Platinen [Lukianto u. Stern- berg, 2011b]	11
2.3	Feder-Masse-Prinzip eines Beschleunigungsmessers entlang einer Achse (abgeleitet nach: http://www.elektronik-kompodium. de/sites/bau/1503041.htm)	12
2.4	Feder-Masse-Prinzip um die Z-Achse für ein Gyroskop (ab- geleitet nach: http://www.hsg-imit.de/fileadmin/gfx/ pdfs/Aif-_Praesentationen_2007/ok_2007_03_22_AiF-Tag_ RefGyro-final_Jan_Dehnert.pdf)	14
2.5	Funktionsweise einer Achse des Magnetfeldsensors auf Basis des Hall-Effekts (abgeleitet nach: [Cai u. a., 2016]).	15
2.6	Funktionsprinzip für einen MEMS-Barometer als Querschnitt durch einen Sensor (abgeleitet nach: http://sus.ziti. uni-heidelberg.de/Lehre/Seminar03/07.pdf).	16
2.7	Prinzip des Resamplings im Bootstrap Partikelfilter (abgeleitet nach: [Aggarwal u. a., 2010] und [Wendel, 2011])	25
2.8	Ausschnitt des vierten Obergeschosses aus dem Gebäudeplan der HCU mit Routing-Graph (blau gestrichelt: Kanten des Routing- Netz, schwarz: Wände und Türen)	32
2.9	Beispiel-Graph als Grundlage für einen Routingalgorithmus. Kos- ten sind Entfernungen, Stockwerksübergänge werden durch Kom- mazahlen repräsentiert [Willemssen u. a., 2014a].	35
2.10	Berechnungsschritte eines Dijkstra Algorithmus angewendet auf den Beispiel-Graph aus Abb. 2.9 [Willemssen u. a., 2014a].	35
2.11	Testgerät Samsung Galaxy Nexus, Marktverfüg- barkeit 2011 (http://www.pcmweb.nl/nieuws/ samsung-galaxy-nexus-met-ice-cream-sandwich.html).	37
2.12	Testgerät LG E960 verfügbar als Google Nexus 4, Markteinführung 2012 [LG, 2016].	39
2.13	Koordinatensystem von Android Smartphones (http:// developer.android.com/images/axis_device.png).	40

2.14	Sensorenübersicht der Hauptplatine im Google Nexus 4. Die markierten Rechtecke fokussieren verschiedene Sensoren, beispielsweise in Gelb die der MPU-6050 (Quelle: https://www.ifixit.com/Teardown/Nexus+4+Teardown/11781).	41
2.15	Drehraten des Gyroskops im Samsung Galaxy Nexus, sichtbar sind die ersten Sekunden nach dem Start (Achsenzuordnung: RGB = XYZ).	42
2.16	Drehraten des Gyroskops im Google Nexus 4, sichtbar sind die ersten Sekunden nach dem Start (Achsenzuordnung: RGB = XYZ).	42
2.17	Rohdaten der x-Achsen der Beschleunigungsmesser in beiden Testgeräten einer gleichzeitigen Messung über einen Zeitraum von 800 Sekunden (schwarz: Nexus 4; grün: Galaxy Nexus).	44
2.18	Relative Höhen abgeleitet aus Luftdruckmessungen für eine Datenaufzeichnung von 800 Sekunden Dauer (schwarz: Nexus 4; grün: Galaxy Nexus).	45
2.19	Rohdaten der x-Achsen beider Magnetfeldsensoren der Testgeräte über einen Zeitraum von 800 Sekunden (schwarz: Nexus 4; grün: Galaxy Nexus).	46
2.20	Magnetisch Nord, berechnet aus den Rohdaten in Abb. 2.19 (schwarz: Nexus 4; grün: Galaxy Nexus).	47
2.21	Arduino Mikroprozessoren: Arduino Mega oben; Arduino Uno links; Arduino Nano rechts. (http://arduino.cc/en/Main/Products)	49
2.22	Versuchsaufbau für die im Nexus 4 verbauten MEMS-Sensoren basierend auf einem Arduino Nano (von links: SD-Karten-Shield, Arduino Nano, MPU-6050, BMP180, 9V Blockbatterie).	50
2.23	Rohdaten des Gyroskops aus dem MPU-6050 im Arduino gesteuerten Versuchsgerät (RGB = XYZ).	52
2.24	Vergleich der z-Achse des Arduino Moduls mit der z-Achse des Beschleunigungsmessers aus dem Testgerät Nexus 4. Die Daten entstammen einem Testlauf im HCU-Gebäude bei dem beide Sensoren zusammen in einer Hand getragen wurden (schwarz: Arduino; blau: Nexus 4).	52

3.1	Relative Höhen des Treppenhauses im ehemaligen HCU-Gebäude aus Luftdruckmessungen des Galaxy Nexus (schwarz: relative Höhen, grün: geglättet auf Basis der Messungen der vorherigen Sekunde, blau: Referenzstockwerkshöhen).	55
3.2	Relative Höhen des Treppenhauses im HCU-Gebäude aus Luftdruckmessungen des Nexus 4 (schwarz: relative Höhen, grün: geglättet auf Basis der Messungen der vorherigen Sekunde, blau: Referenzstockwerkshöhen).	55
3.3	Langzeitmessung des Luftdrucksensors im Galaxy Nexus im Vergleich zum Referenzgerät (Setra B470T).	58
3.4	Temperatur während der Langzeitbeobachtung des Luftdrucksensors im Galaxy Nexus aus Abb. 3.3 (schwarz: Gehäusetemperatur; blau: Umgebungstemperatur).	58
3.5	Langzeitmessung des Luftdrucksensors im Nexus 4 im Vergleich zum Referenzgerät (Setra B470T).	59
3.6	Temperatur während der Langzeitbeobachtung des Luftdrucksensors im Nexus 4 aus Abb. 3.5 (schwarz: Gehäusetemperatur; blau: Umgebungstemperatur).	59
3.7	Modifizierte Totalstation Leica TCRP 1105+ zur Kalibrierung von Gyroskopen im Smartphones [Sternberg u. a., 2012].	60
3.8	Versuchsaufbau für die Kalibrierung der Drehgeschwindigkeiten des Leica TCRP 1105+ mit einem Lasertracker Leica AT901-MR [Sternberg u. a., 2012].	62
3.9	Drehraten der z-Achse des Gyroskops im Galaxy Nexus, erzeugt durch Drehung der Totalstation (blau: Drehraten Gyroskop; schwarze horizontale Linien: Referenzgeschwindigkeiten).	63
3.10	Detailansicht aus Abb. 3.9 für die Drehratenmessung der z-Achse des Gyroskop im Galaxy Nexus (blau: Drehraten Gyroskop; schwarze horizontale Linien: Referenzgeschwindigkeiten).	64
3.11	Detailansicht der Drehratenmessung auf der Totalstation der z-Achse des Gyroskops im Nexus 4 (blau: Drehraten Gyroskop; schwarze horizontale Linien: Referenzgeschwindigkeiten).	64
3.12	Detailansicht der Drehraten während einer fest eingestellten Drehgeschwindigkeit an der Totalstation für eine Kalibriermessung. Periodische Schwingung, hier bei 0,2 rad/s, aufgrund von Motorregelung.	65

3.13	Verbesserungen nach der Ausgleichung der Drehgeschwindigkeiten des Gyroskops im Galaxy Nexus.	67
3.14	Verbesserungen nach der Ausgleichung der Drehgeschwindigkeiten des Gyroskops im Nexus 4.	67
3.15	Drehraten der Gyroskopkalibrierung im Galaxy Nexus für eine Kalibrierungsmessung mit zuvor definierten Winkelschritten (RGB = XYZ).	68
3.16	Integrierte Drehraten aus der Kalibrierungsfahrt des Gyroskops in Abb. 3.15 mit Winkelschritten (RGB = XYZ; rot gestrichelt: um Drift bereinigte x-Achse).	69
3.17	Verwendeter Ablauf einer Kalibrierungsmessung für den Magnetfeldsensor (räumlich) sowie für den Beschleunigungsmesser im Smartphone. Darstellung der Displayposition zum genäherten Horizont.	74
3.18	Ergebnisse der Kalibrierung des Beschleunigungsmessers im Galaxy Nexus mit Kalman Filter. Vergleich der Magnituden (Zielgröße: $g = 9.81 \text{ m/s}^2$) vor und nach der Kalibrierung (rot: Magnitude fehlerbehaftet, schwarz: korrigierte Magnitude).	74
3.19	Ergebnisse der Kalibrierung des Beschleunigungsmessers im Nexus 4 mit Kalman Filter. Vergleich der Magnituden (Zielgröße: $g = 9.81 \text{ m/s}^2$) vor und nach der Kalibrierung (rot: Magnitude fehlerbehaftet, schwarz: korrigierte Magnitude).	75
3.20	Beschleunigungen einer Achse aus dem Testgerät Nexus 4 einer beliebigen Einzelmessung in Ruhelage zu Geschwindigkeiten (gestrichelte Linie) und zu Distanzen (blau) integriert.	78
3.21	Vergleich der Magnitude vor der Kalibrierung und nach der Korrektur aus den drei Achsen des Magnetfeldsensors im Galaxy Nexus (blau: aus Rohdaten; schwarz: korrigiert; Referenzwert für Hamburg: $49,627 \mu T$).	80
3.22	Vergleich der Magnitude vor der Kalibrierung und nach der Korrektur aus den drei Achsen des Magnetfeldsensors im Nexus 4 (blau: aus Rohdaten; schwarz: korrigiert; Referenzwert für Hamburg: $49,627 \mu T$).	80
3.23	Horizontierte Magnetfelddaten der x- und y-Achsen für das Galaxy Nexus (links) und das Nexus 4 (rechts) – vor und nach Berücksichtigung von Korrekturen einer Kalibrierung auf Basis des 2D-Pythagoras (blau: aus Rohdaten; schwarz: korrigiert; rot: Sollkreis).	82

3.24	Vergleich der Richtungen aus magnetisch Nord, berechnet durch die verebneten Magnetfeldsensordaten, und der Drehwinkel, berechnet aus verebneten Gyroskopdaten (blau: Gyroskop; schwarz: Magnetfeldsensor).	84
3.25	Versuchsaufbau am Drehtisch zur Überprüfung der Parallelität der Sensorachsen von Beschleunigungsmesser und Gyroskop im Testgerät Nexus 4.	89
3.26	Versuchsaufbau zur Überprüfung der Uhrendrift im Smartphone (links: Koordinatensystem INS, auf dem das Smartphone aufgesetzt wird, rechts: Versuchsaufbau).	91
3.27	Rohdaten beider z-Achsen der Beschleunigungssensoren aus der Versuchsdurchführung zur Uhrendrift. Die Referenzdaten (rot) und die Daten aus dem Smartphone (schwarz) nach Korrektur des Offsets.	92
3.28	Berechnung der Drift mittels ausgleichender Gerade (oben: Rohdaten mit ausgleichender Gerade (rot); unten: Verbesserungen).	93
4.1	Prinzip der Schritterkennung aus Beschleunigungen der z-Achse (schwarz) nach Verebnung des Smartphone-Koordinatensystems und Reduzierung um g (rot: Schritt erkannt; blau: Offset abhängig von in Ruhelage berechnetem Mittel der Beschleunigungen) [Willemsen u. a., 2015b].	98
4.2	Pedestrian Dead Reckoning mit Sensordaten des Galaxy Nexus im zweiten Obergeschoss des ehemaligen HCU Gebäudes (grün: mit ZUPT-Korrektur; blau: ohne ZUPT-Korrektur) [Willemsen u. a., 2015a].	106
4.3	Verwendeter Drehwinkel r_z aus den Testdaten für das Dead Reckoning in Abb. 4.2 (blau: ohne Korrektur des Drehratenoffsets; schwarz: mit Korrektur).	107
4.4	Genauigkeiten (Helmertscher Punktfehler mit 95 % Konfidenzniveau) der Trajektorie aus Varianz-Kovarianzfortpflanzung (rot: gesamt; grün: nur Anteil Schrittlänge; blau: nur Anteil Gyroskop-Orientierung, cyan: mit ZUPT Korrektur).	109
4.5	Höhen aus relativen Luftdruckmessungen abgeleitet während eines Testlaufs im zweiten Obergeschoss des ehemaligen HCU-Gebäudes.	110
4.6	Positionsbestimmung nach dem Prinzip der Annäherung und Cell of Origin für funkbasierte Verfahren (abgeleitet nach [Blankenbach, 2007]).	112

4.7	Funktionsprinzip des Fingerprintings (rechts: Generierung einer Referenzdatenbank durch Beobachtungen der verfügbaren Signale an bekannten Positionen; links: Positionsschätzung auf Basis eines Vergleichs mit den zuvor gesammelten Referenzdaten) [Willemsen u. a., 2014b].	114
4.8	Positionsschätzung nach dem Prinzip der Trilateration am Beispiel von Innenraumanwendungen. Ableitung der Distanzen zwischen Sender und Empfänger und anschließende Schätzung mit dem Pythagoras als funktionalem Zusammenhang [Willemsen u. a., 2014b].	115
4.9	Prinzip der Triangulation für die Lagebestimmung (abgeleitet nach [Blankenbach, 2007]).	117
4.10	Prinzip des Polaren Anhängens für die Lagebestimmung (abgeleitet nach [Blankenbach, 2007]).	117
4.11	Euklidische Distanz einer Positionsschätzung mittels Fingerprinting durch einen Vergleich zu jeder Position in der vorliegenden Datenbank. Die minimale euklidische Distanz ist die wahrscheinlichste aktuelle Position.	123
4.12	Berechnete Wahrscheinlichkeiten einer Fingerprinting Positionsschätzung auf Basis eines probabilistischen Ansatzes. Der maximale Wert repräsentiert die wahrscheinlichste Position.	124
4.13	Trajektorien im OG5 (schwarz) und OG4 (grau) berechnet nach der Euklidischen Distanz (grün) und dem Occupancy Grid (blau; Referenztrajektorie in rot).	128
4.14	Schematischer Aufbau der verfügbaren Komponenten für eine Indoor-Navigation mittels Smartphonesensoren.	131
4.15	Gebäudeplan des vierten Obergeschosses im HCU-Gebäude mit manuell erzeugtem Routing-Graph (blau).	134
4.16	Beispiel einer Kollisionsberechnung mittels Geradenschnitt aus aktueller Position und Richtung auf Basis einer Vektorkarte [Willemsen u. a., 2014a].	134
4.17	Geometrischer Zusammenhang für die Berechnung von Stützinformationen aus Karten- und Routingdaten (rot: aktuelle Position; blau: Routing-Kanten; schwarz: Wände).	136
4.18	Favorisierte Sensoren und deren Kombinationsmöglichkeit für die Positionsschätzung mittels Smartphone (Spalte 1: Sensoren, Spalte 2: Teilergebnis für die Fusion).	138

5.1	Vergleich der Orientierung aus Magnetfeldsensor (rot) zur Orientierung des Gyroskops (grün) [Willemsen u. a., 2013].	143
5.2	Grafische Darstellung der Verwendung der Kanten des Routingnetzes zur Positionskorrektur im KF (blau: aktuelle Position; grau: neue Schätzung; grün: Fußpunkt auf favorisierter Routing-Kante) [Willemsen u. a., 2015b].	144
5.3	Positionsschätzung mit Kalman Filter im 4. OG der HCU (rot: geschätzte Trajektorie; grün: Fußpunkte auf den Routing-Kanten, die zur Positionskorrektur im Filter eingesetzt wurden) [Willemsen u. a., 2015b].	149
5.4	Treppenerkennung mit Barometerdaten für einen Datensatz im Gebäude D des ehemaligen HCU-Gebäudes (cyan: Schrittlänge; grün: StdAbw. S_0 aus Daten über einen Zeitraum von 1 s).	151
5.5	Höhen aus Luftdruckmessungen für den zweiten repräsentativen Datensatz aus der KF-Schätzung (blau: z-Koordinate; grün: Stockwerksübergang aus Referenzmessung).	153
5.6	KF-Trajektorie (rot) des räumlichen Testdatensatz für das vierte Obergeschoss. Start und Endpunkt an gleicher Position (blau: Routing-Kanten; schwarz: Grundriss).	154
5.7	Trajektorie (rot) des räumlichen Testdatensatz für das zweite Obergeschoss auf Basis des KF (blau: Routing-Kanten; schwarz: Grundriss).	155
5.8	Trajektorie (rot) des räumlichen Testdatensatz für das EG auf Basis des KF (blau: Routing-Kanten; schwarz: Grundriss).	156
5.9	Streuung von Partikeln auf Basis von Schrittlänge und Richtung sowie die anschließende Gewichte-Berechnung durch Wandinformationen des Grundrisses. (blau: aktuelle wahrscheinlichste Position; rot: Partikel mit minimalem Gewicht, da hinter der Wand liegend; Graustufen: Färbung nach Höhe des Gewichts, berechnet in Abhängigkeit der Richtungsdivergenz zwischen Wand und Partikellinie) [Willemsen u. a., 2015b].	159
5.10	Prinzip einer Gewichte-Berechnung der Partikel im PF auf Basis von Routing-Kanten (blauer Punkt: aktuelle Position; grau: gestreute Partikel; cyan: orthogonale Abstände zu ausgewählter Routing-Kante) [Willemsen u. a., 2015b].	160

5.11	Trajektorie für das vierte Obergeschoss aus einer Positionsschätzung auf Basis von PF mit Stützung durch Routing- und Kartendaten (blau: Partikel, rot: gewichtetes Mittel jedes Berechnungsschrittes im PF; magenta: Pedestrian Dead Reckoning ohne Korrekturen) [Willemsen u. a., 2015b].	161
5.12	Anfang und Ende der Trajektorie des zweiten Testdatensatzes im vierten Obergeschoss auf Basis von PF mit Positionskorrektur durch Routing- und Kartendaten (rot: gewichtetes Mittel jedes Berechnungsschrittes im PF; magenta: PDR).	163
5.13	Geschätzte Trajektorie des zweiten Testdatensatzes im zweiten Obergeschoss auf Basis von PF mit Positionskorrektur durch Routing- und Kartendaten (rot: gewichtetes Mittel jedes Berechnungsschrittes im PF; magenta: PDR).	164
5.14	Erdgeschoss der Trajektorie des zweiten Test-Datensatzes auf Basis von PF mit Positionskorrektur durch Routing- und Kartendaten (rot: gewichtetes Mittel jedes Berechnungsschrittes im PF; magenta: PDR).	165
5.15	Ablaufdiagramm der Positionsschätzung nach dem topologischen Ansatz auf dem Routing-Graphen, abgeleitet nach [Willemsen u. a., 2015b].	167
5.16	Grafisch visualisierte Positionsberechnung der jeweiligen Zustände des topologischen Ansatzes zur Positionsschätzung [Willemsen u. a., 2015b].	169
5.17	Trajektorie des ersten Datensatzes auf Basis des topologische Ansatzes im vierten Obergeschoss des HCU-Gebäudes (rot: Trajektorie; cyan: verwendete Routing-Kanten) [Willemsen u. a., 2015b].	171
5.18	Vergleich der Orientierung zwischen Gyroskopdaten (schwarz) und ausgewählten Routing-Kanten (rot) des ersten Datensatzes im vierten Obergeschoss [Willemsen u. a., 2015b].	172
5.19	Detail der Trajektorie aus Abb. 5.17 berechnet nach dem topologischen Ansatz: Hier Zustand 4 aktiv, globale Suche nach neuer Bezugskante [Willemsen u. a., 2015b].	173
5.20	Trajektorie (rot) des zweiten räumlichen Testdatensatzes, Positionsschätzung basierend auf den topologischen Ansatz im vierten Obergeschoss.	174

5.21	Trajektorie (rot) des zweiten räumlichen Testdatensatzes für das zweite Obergeschoss.	174
5.22	Trajektorie (rot) des zweiten räumlichen Testdatensatzes für das Erdgeschoss.	175
5.23	Differenzen zur Referenztrajektorie für alle drei verwendeten Algorithmen in jedem Schritt (rot: Kalman Filter; grün: Partikelfilter; blau: kantenbasiertes PDR).	178
5.24	Auswirkungen der Positionskorrektur basierend auf Wandabfragen (oben: Anwendung eines Unsicherheitsbandes, um Ecken überqueren zu können; unten: zwei Zustände, rot: Stauchung durch Wandstützung, gelb: Abbruch der Berechnung, wenn der Filter den Durchbruch nicht ermöglicht).	182
5.25	Grafische Darstellung der Positionsschätzung auf Basis der Fusionierung des topologischen Ansatzes mit einem PF (grün: Partikelpositionen und -richtungen; rot: Partikelgewicht = 0).	186
5.26	Konzeptioneller Ablauf des kantenbasierten Richtungspartikelfilters.	189
5.27	Trajektorie im vierten Obergeschoss des HCU-Gebäudes basierend auf dem kantenbasierten Richtungspartikelfilter mit favorisierter Route, ausgewählt auf Basis der einer Punktgruppe zugeordneten Anzahl von Partikeln (magenta: favorisierte Route, grün: Partikel, rot: PDR).	191
5.28	Trajektorie im vierten Obergeschoss des HCU-Gebäudes basierend auf dem kantenbasierten Richtungspartikelfilter mit favorisierter Route, ausgewählt in Abhängigkeit der Gewichtssumme der einer Punktgruppe zugeordneten Partikel (blau: favorisierte Route, grün: Partikel, rot: PDR).	192
5.29	Details einer Trajektorie des ersten Testdatensatzes im vierten Obergeschoss, berechnet nach dem kantenbasierten PF aus Abb. 5.27 (links: Übergang offener Bereich mit Wandstützung, Mitte: Entkoppeln von Kanten, rechts: mehrere Routen).	193
5.30	Differenzen zur Referenztrajektorie für in jedem Schritt für den PF- und topologischem Ansatz sowie der Fusion, dem kantenbasierten PF. (grau: kantenbasierter PF ausgewählt nach Anz. der Routenpartikel; grün: Partikelfilter; blau: kantenbasiertes PDR).	194

5.31	Trajektorie des zweiten Testdatensatzes für das vierte Obergeschoss auf Basis des kantenbasierten PF (rot: PDR, grün: Partikel, magenta: favorisierte Route nach Anz. Partikel, blau: favorisierte Route nach Summe Gewicht; magenta und blau überlagern sich).	196
5.32	Trajektorie des zweiten Testdatensatzes für das zweite Obergeschoss auf Basis des kantenbasierten PF (rot: PDR, grün: Partikel, magenta: favorisierte Route nach Anz. Partikel, blau: favorisierte Route nach Summe Gewicht; magenta und blau sind überlagert dargestellt).	197
5.33	Trajektorie des zweiten Testdatensatzes für das Erdgeschoss auf Basis des kantenbasierten PF (rot: PDR, grün: Partikel, magenta: favorisierte Route nach Anz. Partikel, blau: favorisierte Route nach Summe Gewicht; magenta und blau sind überlagert dargestellt).	198
5.34	Trajektorien von 20 Nutzern berechnet mit Pedestrian Dead Reckoning für eine Route im dritten Obergeschoss zwischen Raum 3.110 und Instrumentenausgabe (schwarz-cyan: jeweils eine Route bzw. Nutzer).	203
5.35	Trajektorien von 20 Nutzern berechnet mittels kantenbasiertem Partikelfilter. Auswahl der favorisierten Route nach Anzahl zugeordneten Partikeln der jeweiligen Routing-Kanten (schwarz-cyan: jeweils eine Route bzw. Nutzer).	204
5.36	Trajektorien von 20 Nutzern berechnet mittels kantenbasiertem Partikelfilteransatz. Auswahl der favorisierten Route nach der Gewichtssumme der zugeordneten Partikel der jeweiligen Routing-Kanten (schwarz-cyan: jeweils eine Route bzw. Nutzer).	205
5.37	Trajektorien im dritten Obergeschoss berechnet auf Basis von PDR für die 22 verwendeten Datensätze der räumlichen Untersuchung (schwarz-cyan: jeweils eine Route bzw. Nutzer; farblich skalierte Routen können abbildungsübergreifend zugeordnet werden).	207
5.38	Trajektorien im Erdgeschoss berechnet auf Basis von PDR für die 22 verwendeten Datensätze der räumlichen Untersuchung (schwarz-cyan: jeweils eine Route bzw. Nutzer; farblich skalierte Routen können abbildungsübergreifend zugeordnet werden).	207

5.39	Trajektorien im dritten Obergeschoss berechnet auf Basis des kantenbasierten PF für die 22 verwendeten Datensätze der räumlichen Untersuchung. Favorisierte Route durch die Partikelanzahl bestimmt. (schwarz-cyan: jeweils eine Route bzw. Nutzer; farblich skalierte Routen können abbildungsübergreifend zugeordnet werden).	208
5.40	Trajektorien im Erdgeschoss berechnet auf Basis des kantenbasierten PF für die 22 verwendeten Datensätze der räumlichen Untersuchung. Favorisierte Route durch die Partikelanzahl bestimmt. (schwarz-cyan: jeweils eine Route bzw. Nutzer; farblich skalierte Routen können abbildungsübergreifend zugeordnet werden)	209
5.41	Trajektorien im dritten Obergeschoss berechnet auf Basis des kantenbasierten PF für die 22 verwendeten Datensätze der räumlichen Untersuchung. Favorisierte Route durch die Summe der Partikelgewichte bestimmt. (schwarz-cyan: jeweils eine Route bzw. Nutzer; farblich skalierte Routen können abbildungsübergreifend zugeordnet werden).	210
5.42	Trajektorien im Erdgeschoss berechnet auf Basis des kantenbasierten PF für die 22 verwendeten Datensätze der räumlichen Untersuchung. Favorisierte Route durch die Summe der Partikelgewichte bestimmt. (schwarz-cyan: jeweils eine Route bzw. Nutzer; farblich skalierte Routen können abbildungsübergreifend zugeordnet werden).	211
6.1	Fotografie eines Flucht- und Rettungsplans für die Nordseite des vierten Obergeschosses im HCU-Gebäude.	218
6.2	Visualisierung der Teilergebnisse einer Berechnung des vektorbasierten Grundrisses aus einem Bild des Flucht- und Rettungsplans der Nordseite des vierten Obergeschosses der HCU (oben: Helligkeit und Farbe angepasst; Mitte: Farben reduziert und Binärbild erzeugt, unten: Vektordaten aus Binärbild).	220
6.3	Ergebnis einer automatischen Kantengenerierung für einen Routing-Graphen aus einem vektorbasierten Grundriss des zweiten Obergeschosses der HCU (grün: berechnete Kanten, rot: Grundriss) [Merkel u. a., 2015].	222
6.4	Ergebnis einer automatischen Kantengenerierung für einen Routing-Graphen im vierten Obergeschoss der HCU auf Basis einer rasterbasierten Karte per Erosionsverfahren.	224

6.5	QR-Code zur Initialisierung der Navigation. Anbringung beispielsweise an Namensschildern von Türen. Empfohlene Größe $20 \times 20 \text{ mm}^2$.	226
6.6	Oberfläche der Applikation MapNavi zur Positionsschätzung auf Basis eines KF. Initialisierung mittels QR-Code. [Willemsen u. a., 2014b].	228
6.7	Oberfläche der App NavCon: Zu sehen ist das Routing im vierten Obergeschoss der HCU.	230
6.8	Favorisierte Komponenten einer Navigationsanwendung und die Zuordnung zu Threads in der Programmentwicklung für eine Smartphone-App.	231

Tabellenverzeichnis

2.1	Vergleich von Partikelfilter und Kalman Filter nach Basiskriterien (+ = gut, 0 = neutral, - = schlecht [Schwarzlmüller, 2007]	28
2.2	Für die Positionsschätzung interessante MEMS-Sensoren im Smartphone Samsung Galaxy Nexus	38
2.3	Für die Positionsschätzung interessante MEMS-Sensoren im Google Nexus 4 / LG E960	39
2.4	Auflösung und Arbeitsbereich von Gyroskop und Beschleunigungssensor im MPU-6050 ([InvenSense, 2015]).	51
3.1	Vergleich der berechneten Stockwerkhöhen aus den Luftdruckmessungen beider Testgeräte zu Referenzhöhen.	57
3.2	Vergleich der Drehraten aus Lasertrackerdaten mit den eingestellten Drehraten an der Totalstation [Sternberg u. a., 2012].	63
3.3	Ergebnisse der Ausgleichung aus den Kalibriermessungen für die z-Achsen der Gyroskope in beiden Testgeräten.	66
3.4	Soll-Ist-Vergleich für berechnete Winkel aus der Kalibrierung des Gyroskops auf Basis von Winkelmessungen mithilfe der Totalstation; ausgewählte Winkel, ermittelt aus den Daten in Abb. 3.16. . .	70
3.5	Berechnete Korrekturparameter mit Standardabweichungen für die Kalibrierung des Beschleunigungsmessers beider Testgeräte.	76
3.6	Berechnete Korrekturparameter aus zwei Kalibriermessungen jeweils für beide Magnetfeldsensoren in den Testgeräten auf Basis des räumlichen funktionalen Zusammenhangs.	81
3.7	Berechnete Korrekturparameter nach dem 2D-Ansatz zweier Kalibriermessungen für die Magnetfeldsensoren beider Testgeräte. . . .	83
3.8	Ergebnisse für Offset, Maßstab und Orthogonalität des zweiten Testdatensatzes zur Kalibrierung der Beschleunigungen unter Hinzunahme der Orthogonalitätsparameter für die Testgeräte Galaxy Nexus und Nexus 4.	87
3.9	Einflussgröße der Korrektur durch Orthogonalität am Beispiel des Beschleunigungsmessers der Testgeräte Galaxy Nexus und Nexus 4.	88
3.10	Orthogonalitätsparameter als Kriterium für die Parallelität der Sensorachsen des Gyroskops zum Beschleunigungsmesser des Nexus 4.	90

4.1	Vergleich der Positionsschätzung im Fingerprinting nach dem deterministischen und dem probabilistischen Ansatz gegen Referenzpositionen. Der Wert 0.0 repräsentiert eine richtige Zuordnung. Die Abweichungen sind der räumliche Abstand zwischen Ist- und Soll-Koordinaten.	126
4.2	Vergleich der einzelnen Positionen für die Berechnung der Trajektorie nach dem probabilistischen und dem deterministischen Ansatz. Der Wert 0.0 repräsentiert eine richtige Zuordnung. Die Abweichungen sind der räumliche Abstand zwischen Ist- und Sollkoordinaten.	129
5.1	Differenzen zu Referenzpositionen einzelner Positionen auf den Trajektorien berechnet mit KF, PF und topologischem Ansatz des ersten Testdatensatzes im vierten Obergeschoss, ausgewählt in Abhängigkeit zur Länge der Laufstrecke [Willemsen u. a., 2015b] . . .	177
5.2	Koordinatendifferenzen vor dem Stockwerksübergang zu den Korrekturpositionen für KF, PF und kantenbasierten PDR. Mögliche Orte des Stockwerksübergangs bzw. für ein CUPT sind fettgedruckt. Sie ergeben sich aus der Treppenerkennung und Luftdruckveränderungen. Fettgedruckte Differenzen repräsentieren die geringste Koordinatendifferenz und damit den ausgewählten Stockwerksübergang.	180
5.3	Koordinatendifferenzen vor dem Stockwerksübergang zu den Korrekturpositionen für das kantenbasierten PF sowie PF und kantenbasierten PDR. Mögliche Orte des Stockwerksübergangs bzw. für ein CUPT werden fett gedruckt dargestellt und ergeben sich aus der Treppenerkennung und Luftdruckveränderungen. Fettgedruckte Differenzen repräsentieren die geringste Koordinatendifferenz und damit den ausgewählten Stockwerksübergang.	199
5.4	Vergleich der Differenzen zwischen Start- und Endpunkt der insgesamt 20 verwendeten 2D-Trajektorien im dritten Obergeschoss nach den unterschiedlichen Berechnungsmethoden. Zuordnung in Genauigkeitsklassen.	206
5.5	Vergleich der Differenzen zwischen Start- und Endpunkt der insgesamt 22 verwendeten räumlichen Trajektorien nach den unterschiedlichen Berechnungsmethoden. Zuordnung in Genauigkeitsklassen.	212

6.1 Vergleich der Rechenzeit für fünf Testrouten mit dem Floyd-Warshall und dem Dijkstra Algorithmus (abgeleitet nach [Dalladas u. a., 2014])	215
---	-----

Stichwort- und Variablenverzeichnis

a_P	Abszisse
AP	Access Point
BMP	Bosch Motion Pressure
CUPT	Coordinate update
DR	Dead Reckoning
GNSS	Global Navigation Satellite System
GSM	Global System for Mobile Communications
HCU	HafenCity Universität
INS	Inertial Navigationssystem
KF	Kalman Filter
KNN	Künstliches Neuronales Netz
LTE	Long Term Evolution
MdkQ	Methode der kleinsten Quadrate
MEMS	Micro Electronical Mechanical System
MPE	Maximum Permissible Error
MPU	Motion Processing Unit
NEMS	Nano Electronical Mechanical System
o_P	Ordinate
PDR	Pedestrian Dead Reckoning
PF	Partikelfilter
QR-Code	Quick Response Code
Q_x	Kovarianzmatrix des Unbekanntenvektors x
RFID	Radion Frequency Identification
S_0	Standardabweichung einer Beobachtung vom Gewicht 1
UWB	Ultra Wide Band
v	Messwertrauschen
w	Systemrauschen
WDF	Wahrscheinlichkeitsdichtefunktion
x	Zustandsvektor
y	Messwertvektor
ZUPT	Zero velocity update