



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Amir Baghery

Anbindung von Bluetooth Low Energy Inertialsensoren an
Android-Systemen zur Lokalisierung in Gebäuden

Amir Baghery

Anbindung von Bluetooth Low Energy Inertialsensoren an Android-Systemen zur Lokalisierung in Gebäuden

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Technische Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer. nat. Hans H. Heitmann
Zweitgutachter : Prof. Dr.-Ing. Andreas Meisel

Eingereicht am: 16. Februar 2015

Amir Baghery

Thema der Arbeit

Anbindung von Bluetooth Low Energy Inertialsensoren an Android-Systemen zur Lokalisierung in Gebäuden

Stichworte

Lokalisierung in Gebäuden, Koppelnavigation, Kartenabgleich, Bluetooth Low Energy, Inertialsensoren, Android-Systemen

Kurzzusammenfassung

Diese Bachelorarbeit befasst sich mit der Lokalisierung anhand Inertialsensoren durch die Verbindungsart Bluetooth Low Energy. Es werden Lokalisierungsmöglichkeiten durch Inertialsensoren analysiert und die passenden Lokalisierungsalgorithmen ausgewählt.

Im Rahmen dieser Bachelorarbeit wird eine Applikation in Android System entwickelt, die zum Empfangen und Bereitstellen von Sensordaten, Implementierung der ausgewählten Algorithmen, Lokalisierung der Position und die Anpassung auf die Karte dient. Anschließend werden die einzelnen Komponenten getestet und ein Systemtest durchgeführt, um die Lokalisierung zu bewerten.

Amir Baghery

Title of the paper

Connecting Bluetooth Low Energy inertial sensors on Android systems for localization in buildings

Keywords

Localization in buildings, Indoor localization, map matching, Bluetooth Low Energy, Inertial sensors, Android systems

Abstract

This bachelor thesis is concerned with the localization in buildings based on inertial sensors connected with Bluetooth Low Energy. Therefore, inertial sensors localization's options are analyzed, and the appropriate localization algorithms are selected.

As part of this thesis an application in Android system is developed, which is used to receive and provide sensor's data, implement the selected algorithms, locate the position and map matching. The individual components are tested, and a system test is performed to evaluate the localization.

Inhaltsverzeichnis

1	Einleitung	7
1.1	Motivation	7
1.2	Zielstellung der Arbeit	8
2	Technische und theoretische Grundlagen	9
2.1	Android Betriebssystem	9
2.2	Bluetooth Low Energy (BLE)	11
2.3	Inertialsensoren	11
2.3.1	Beschleunigungssensor	12
2.3.2	Gyroskope	12
2.3.3	Magnetometer	13
2.3.4	SensorTag	14
2.4	Globale Satelliten Navigationssysteme (GNSS)	14
2.5	Lokalisierung durch Inertialsensoren	15
2.5.1	Trägheitsnavigationssystem (INS)	15
2.5.2	Koppelnavigation	15
2.6	Map Matching	17
2.7	Kalibrierung eines Lokalisierungssystems	18
3	Vergleichbare Lokalisierungsverfahren	19
3.1	Lokalisierung durch Inertialsensoren	19
3.1.1	Inertialnavigation	19
3.1.2	Koppelnavigation	21
3.2	Ortungverbesserung in Gebäuden	30

3.2.1	Filterung der Messdaten	30
3.2.2	Map Matching	31
4	Analyse des entwickelten Lokalisierungssystems	35
4.1	Verwendete Filter für die Beschleunigungswerte	36
4.2	Schritterkennung.....	37
4.3	Schrittlänge	42
4.4	Schrittrichtung.....	42
4.5	Kartenanpassung (Map Matching).....	45
5	Systemarchitektur und Implementierung.....	49
5.1	Hardware Komponente.....	49
5.1.1	SensorTag.....	49
5.1.2	Nexus.....	54
5.2	Software Architektur	55
5.2.1	Komponenten.....	55
5.2.2	Klassen.....	56
5.2.3	Grafische Benutzeroberfläche (GUI)	59
5.3	Entwicklungsumgebung	64
5.3.1	Eclipse.....	64
5.3.2	Software Development Kit (SDK)	64
6	Evaluation	65
6.1	Testaufbau.....	65
6.2	Komponententest	66
6.2.1	Schritterkennung.....	66
6.2.2	Schrittlänge	67
6.2.3	Schrittrichtung.....	68
6.2.4	Map Matching	68
6.3	Systemtest.....	70
6.3.1	Erstes Szenario	71
6.3.2	Zweites Szenario	72
7	Schluss	75

7.1	Zusammenfassung	75
7.2	Bewertung.....	76
7.3	Ausblick	77
	Abbildungsverzeichnis	79
	Diagrammverzeichnis	81
	Tabellenverzeichnis	82
	Literaturverzeichnis	83

1 Einleitung

In diesem Kapitel wird zuerst die Motivation der Arbeit vorgestellt. Weiterhin wird die Zielstellung der Arbeit beschrieben.

1.1 Motivation

Die Lokalisierung und die Navigation spielen eine große Rolle in unserem alltäglichen Leben. Der Mensch benutzt sie beim Fahren, sie wird beim Fliegen eingesetzt und auch bei der Seefahrt. Personenlokalisierung wird aber auch oft in verschiedenen Gebieten eingesetzt, beispielsweise bei der Lokalisierung in geschlossenen Bereichen, wo kein GPS-Empfang besteht und auf sie als alternative Positionsbestimmung zurückgegriffen wird. Mögliche Einsatzorte sind Einkaufszentren, Museen, Bahnhöfen oder Flughäfen. Für Marketingzwecke wird sie in Einkaufszentren eingesetzt, um die Position von Kunden analysieren zu können. In Bahnhöfen oder Flughäfen ist sie für die Besucher interessant, um den nächsten Zug oder Flugzeug nicht zu verpassen bzw. rechtzeitig zu erreichen. Besonders in der Lagerlogistik ist auch die Indoor Lokalisierung sehr gefragt. Umso größer das Lager desto schwieriger wird das Auffinden des gesuchten Artikels. Daher ist vorstellbar, dass nur durch die Eingaben der Artikelnummer in ein Smartphone der Weg zum entsprechenden Regal angezeigt und die Person dahin navigiert wird.

Aufgrund des wachsenden Interesses an der Personenlokalisierung in geschlossenen Räumen, wurde viel danach geforscht. Hierbei besteht die Problematik in der globalen Verfügbarkeit für ein betriebsfertiges System, das hierfür keine zusätzlichen Komponenten eingebaut werden müssen und der geforderten Präzision. Es wird zwischen Präzision, Zuverlässigkeit, Installationsaufwand und Kosten abgewägt, wobei durch Nutzung der Inertialsensoren zusätzlicher Installationsaufwand und Kosten eingespart werden.

Texas Instruments vereinte die Inertialsensoren mit den genannten Eigenschaften kostengünstig und durch die „Bluetooth Low Energy“ sogar Energiesparend in einem kompakten Entwicklungsbausatz, dem SensorTag. Die geringe Größe ermöglicht die Unterbringung des SensorTags an beliebigen Stellen am Körper, wodurch auch die

Bewegungsfreiheit nicht beeinträchtigt wird. Darüber hinaus bietet er die Anbindung an mobile Geräte, die für die Entwicklung des Lokalisierungssystems benötigt wird. Durch das Zusammenspiel von SensorTag und den in Android eingebetteten Lokalisierungsverfahren wird die Lokalisierung in Gebäuden ermöglicht.

1.2 Zielstellung der Arbeit

Ziel dieser Bachelorarbeit ist die Forschung und Entwicklung einer prototypischen Anwendung in Android Systemen zur Lokalisierung in Gebäuden. Es wird anhand der Anfangsposition und dem Analysieren von Sensordaten die Position ermittelt. Hierbei wird ein SensorTag, der mit Inertialsensoren ausgestattet ist, mit einem Android basierten Tablet durch die Funktechnik Bluetooth Low Energy verbunden.

Es werden Algorithmen zur Auswertung der Sensordaten und Kartenanpassung verglichen und die entsprechend ausgewählten Algorithmen werden in der Applikation implementiert und getestet. Schließlich wird die Lokalisierung durch Inertialsensoren im Gebäude anhand verschiedener Szenarien getestet.

2 Technische und theoretische Grundlagen

In diesem Kapitel werden relevante Grundlagen, die für diese Arbeit wichtig sind, dargestellt. Darunter zählen das Betriebssystem, die angewendeten Inertialsensoren, die drahtlose Kommunikation und die benutzte Lokalisierungsmöglichkeit in Gebäuden.

2.1 Android Betriebssystem

Android ist ein Open Source Betriebssystem für mobile Geräte wie zum Beispiel Smartphones, Mobiltelefone, Netbooks und Tablet-Computer. Zuerst wurde Android als Betriebssystem für Digitalkameras vorgesehen, erzählte Andy Rubin auf einer Konferenz (vgl. Pakalski, Ingo). Er gründete das Unternehmen Android im Jahr 2003. Android Betriebssystem wurde im Jahr 2005 von Google Inc. übernommen und im Jahr 2007 vorgestellt. Derzeit wird es sehr stark als Mobiles Betriebssystem eingesetzt.

Die Architektur baut sich auf dem Linux-Kernel auf und bietet die von Linux-Betriebssystemen bekannte Prozessverwaltung. Jedoch geht Android ganz andere Wege als andere Linux Versionen, die sich bei der Anpassung am Kernel bis zur Virtuellen Maschine Dalvik¹ erkennbar macht (vgl. Becker, Arno).

In Abbildung [2.1](#) wird der Aufbau des Android Betriebssystems dargestellt. Android besteht aus fünf Schichten. In der untersten Schicht, dem Linux Kernel, wird eine Hardwareabstraktionsschicht für den Android Stack zur Verfügung gestellt und bietet Speichermanagement, Prozessmanagement, Sicherheitsmodul und eine Netzwerk-Schnittstelle als Funktionalitäten.

¹ Dalvik ist eine virtuelle Ausführungsumgebung in Form einer Registermaschine, der für mobile Geräte vorgesehene wurde.

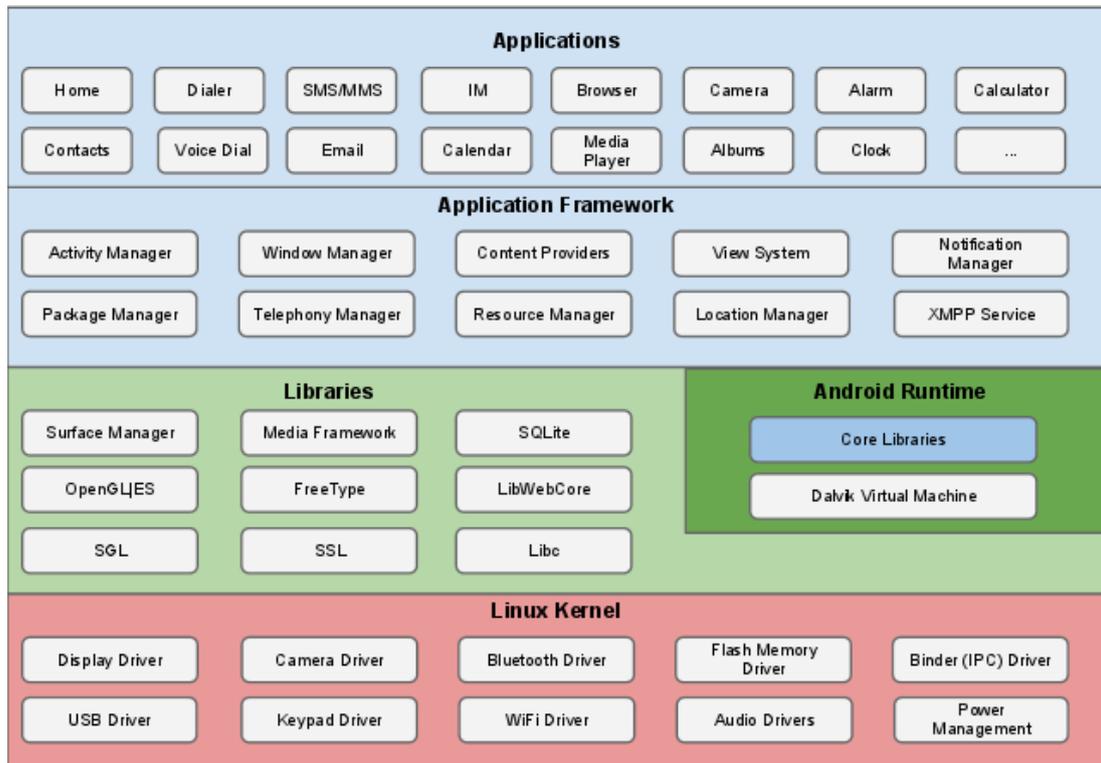


Abbildung 2.1: Die Struktur von Android (Android)

Android stellt in der Libraries Schicht verschiedene Bibliotheken zur Verfügung, die in C/C++ Programmiersprache geschrieben sind. Die Android Runtime Schicht besteht aus der Dalvik Virtual Machine (Dalvik VM) und Core Libraries. Dalvik VM trägt des Entwicklers Vorfahren Ortsnamen und ist eine Virtuelle Ausführungsumgebung auf Registerbasierter Architektur. Es handelt sich bei den Core Libraries um zahlreiche Pakete aus der Java Standard Edition (Java SE), die eine große Erleichterung für Java-Entwickler bei der Programmierung von Android-Applikationen ist.

Das Application Framework basiert auf Java und ist für Android-Entwickler die interessanteste Schicht des Systems, da hier verschiedene Programmierschnittstellen von Android bereitgestellt werden, welche zur Nutzung von Anwendungen dienen. Wie zum Beispiel der Activity Manager, der den Lebenszyklus einer Anwendung steuert.

Die Applications Schicht enthält die Android-Anwendungen, bei denen es sich um Standard- und Eigenanwendungen handelt. Die Standardanwendungen wie zum Beispiel Browser, Email, Rechner, Messaging, Bildergalerie, Telefon, etc. wurden von Google bereitgestellt. Die Eigenanwendungen werden von Android Entwickler angefertigt, die im Play Store zu finden sind.

2.2 Bluetooth Low Energy (BLE)

Bluetooth ist eine drahtlose Kommunikationstechnik, die auch Funktechnik genannt wird, womit ein Gerät innerhalb eines kleineren Abstandes vernetzt werden kann. Bluetooth Technologie wurde von Ericsson¹ im Jahr 1994 entwickelt. Es wurde anfänglich als eine drahtlose Alternative zu RS-232-Datenleitungen konzipiert (vgl. Bluetooth).

Bluetooth Low Energy, auch Bluetooth Smart genannt (kurz BLE), ist die energie sparende Version der Bluetooth Technologie. Durch ihr Design verbraucht sie nur einen Bruchteil der Leistung ihres Vorgängers und ermöglicht somit kleineren Geräten, die ihre Energie durch winzigere Batterien beziehen, die Nutzung. Die Reichweite des Bluetooths kann je nach Anwendungsbedarf optimiert werden. Die Reichweite in den meisten Bluetooth Geräten auf dem Markt ist auf ca. 10 Meter begrenzt. BLE sendet in den 2,4 GHz Bereich mit einer Übertragungsrate von maximal 1 Mbit/s. Es ist in der Integration signifikant kostengünstiger (vgl. Bluetooth Smart).

BLE wird in den Android Geräten ab Version 4.3 unterstützt. Hierdurch erhält der Nutzer die Möglichkeit die in Reichweite befindlichen anderen Geräten zu entdecken, sich mit einander zu verbinden, deren Dienstleistung abzufragen, sowie lesen und schreiben ihrer Eigenschaften.

2.3 Inertialsensoren

Inertialsensoren werden auch Mikroelektromechanische Systeme (kurz MEMS) genannt. MEMS Sensoren sind mikrosystemtechnisch aufgebaute Komponenten, die physikalische Größen in ein elektronisches Signal transformieren. Inertialsensoren bestehen im Allgemeinen aus Beschleunigungs-, Drehraten- und Magnetsensoren. Im Folgenden werden Beschleunigungssensor, Drehratensensor bzw. Gyroskop und Magnetometer näher beschrieben.

¹ Ericsson ist ein schwedisches Unternehmen mit Sitz in Stockholm. Das Kerngeschäft von Ericsson sind Mobilfunktechnologie, Internet- und Multimediakommunikation und Telekommunikation (vgl. Ericsson).

2.3.1 Beschleunigungssensor

Ein Beschleunigungssensor (engl. Accelerometer) misst seine Beschleunigung. Das erste newtonsche Gesetz besagt: „Ein Körper verharrt im Zustand der Ruhe oder der gleichförmigen Translation, sofern er nicht durch einwirkende Kräfte zur Änderung seines Zustands gezwungen wird.“ (vgl. Knežević, Karmen and Kraljević, Lidija). Also ist die Beschleunigung \vec{a} das Ergebnis der Division aus auftretender Kraft \vec{F} und der Masse m (Formel [2.1](#)) (vgl. Richard, Hans A. and Sander, Manuela).

$$\vec{a} = \frac{\vec{F}}{m} \quad (2.1)$$

Beschleunigungssensoren messen die Beschleunigung in drei Achsen (Abbildung [2.2](#)). Im Gegensatz zu einem Gyroskope, der nur Richtungsänderungen messen kann, können sie auch die Schwerkraftrichtung messen.

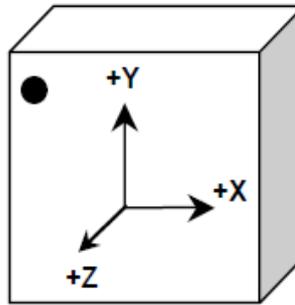


Abbildung 2.2: Drei Achsen von Beschleunigungssensor

Beschleunigungssensoren werden für mehrere Anwendungen in der Industrie und Wissenschaft eingesetzt, aber auch in Alltag. In modernen Autos ist es ein Sensor der das Auslösen eines Airbags bewirkt. Die Hochempfindlichen Beschleunigungssensoren werden in Flugzeugen und Raketen für Trägheitsnavigationssysteme benutzt. Ein interessantes Gebiet ist die Nutzung des Beschleunigungssensors bei den Lokalisierungssystemen.

2.3.2 Gyroskope

Ein Gyroskope, auch Kreiselinstrument oder Drehratensensor genannt, misst die Winkelgeschwindigkeit um eine Drehachse. Die Bezeichnung der Umdrehung um die drei Hauptachsen wird in Abbildung [2.3](#) dargestellt. Die Drehungen um die Hochachse (Y-Achse) werden Gieren bzw. Azimut (engl. yaw), um Querachse (Z-Achse) Nicken (engl. pitch) und um Längsachse (X-Achse) Rollen (engl. roll) genannt.

Bei der Nutzung des Gyroskops entsteht eine Drift der Sensordaten, welche meistens durch kleine mechanische Mangelhaftigkeiten verursacht werden. Hierbei spricht man von einem systematischen Fehler, der mit der Zeit zu einer erheblichen Abweichung führt.

Ein guter Aspekt des Gyroskops ist, dass es nicht durch die Schwerkraft der Erde beeinflusst wird. Gyroskope werden in Navigationsgeräten, sowie in Lageregelungsgeräten verwendet. In Navigationsgeräten werden durch Auswertung der Gyroskopdaten die Richtung und Bewegung ermittelt.

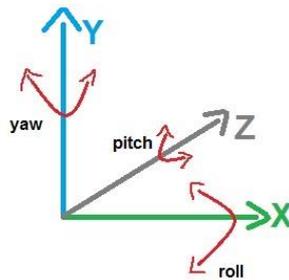


Abbildung 2.3: Umdrehung um die Hauptachsen

2.3.3 Magnetometer

Ein Magnetometer ist ein Sensor zur Messung und Kontrolle von Veränderungen der magnetischen Flussdichte, welches in der Einheit Tesla (T) gemessen wird. Dabei liegen die gemessenen Werte meistens in einem Bereich von ca. 10^{-15} T bis 10 T (vgl. Wikipedia, Magnetometer). Ein Magnetometer wird oftmals eingesetzt um das Erdmagnetfeld mit seinen Schwankungen zu messen. In der Raumfahrt haben Magnetometer eine hohe Bedeutung, da sie dort als Lagersensoren eingesetzt werden, wodurch die Lage eines Satelliten bestimmt wird. Um die Ausrichtung und die Bewegung des Smartphones erfassen zu können, werden neben anderen Sensoren auch Magnetometer eingesetzt. Ein weiterer Einsatzbereich des Magnetometers in Navigationssystemen ist die Kompassmessung als auch die Ermittlung der Bewegungsrichtung des Empfängers.

Magnetometer sind im Allgemeinen in die zwei Gruppen Scalar- und Vector-Magnetometer aufgeteilt.

Scalar-Magnetometer misst die Gesamtstärke des magnetischen Feldes, wohingegen ein Vector-Magnetometer die Komponente des magnetischen Feldes in einer bestimmten Richtung misst (vgl. Iffländer, Lukas and Gageik, Dipl.-Ing N.).

2.3.4 SensorTag

SensorTag ist ein Bluetooth-Entwicklungsbausatz für drahtlose Sensoranwendungen, der sich gut für Smartphone App-Entwickler eignet. SensorTag bietet dank des Mikrokontrollers CC2541, der on Chip mit 256 KB Flash und 8 KB RAM und Drahtlosem Bluetooth Low Energy ausgerüstet ist, die Möglichkeit einer kostengünstigen Kommunikationsart. Der SensorTag wird mit der Firmenware für die Konfiguration der Sensoren und der Kommunikation mit Bluetooth 4.0 fähigen Smartphones vorkonfiguriert. SensorTag bietet für seine Sensoren eine Frequenz in Bereichen von 0,4 Hz bis 10 Hz (vgl. Texas Instruments, SensorTag Development Kit). In der Grundeinstellung sind alle Sensoren deaktiviert und bei Bedarf kann jeder einzelne Sensor angeschaltet werden, wobei die Sensoren Werkseitig auf 1 Hz eingestellt sind. SensorTag beinhaltet sechs Sensoren für die Messung der Beschleunigung, Drehraten, Temperatur, Feuchtigkeit, Magnetfelder und Druck. Die Sensoren sind wie folgt (vgl. Texas Instruments, SensorTag):

- IR Temperature Sensor (Texas Instruments, TMP006)
- Humidity Sensor (Sensirion, SHT21)
- Pressure Sensor (EPCOS, T5400)
- Accelerometer (Kionix, KXTJ9)
- Gyroscope (Invensense, IMU-3000)
- Magnetometer (Freescale, MAG3110)

In dieser Arbeit werden der Beschleunigungssensor, das Magnetometer und das Gyroskope verwendet.

2.4 Globale Satelliten Navigationssysteme (GNSS)

Das Globale Satellitennavigationssystem (kurz GNSS engl. Global Navigation Satellite System) ist ein System zur Ortsbestimmung. Das bekannteste GNSS, das Satellite Global Positioning System (kurz GPS), wurde in den 1970er Jahren vom US Verteidigungsministerium entwickelt (vgl. Schmid, Johannes). Es wurde früher im militärischen Bereich zu Positionsbestimmung und Navigation eingesetzt. Seit den 1990er ist GPS voll funktionsfähig. Es wurde am Mai 2000 durch Abschaltung der künstlichen Signalverschlechterung auch für zivile Zwecke frei nutzbar gemacht (vgl. U.S. Government). GNSS hat eine Positionsgenauigkeit von oft besser als 10 Metern, wobei es durch wetterabhängigen Änderungen der Atmosphäre zu Fehlerquellen in der Ortsbestimmung kommen kann. Weitere Satelliten für die Nutzung des Navigationssystems sind das europäische GALILEO, das russische GLONASS und das chinesische Beidou (vgl. Mao, Guoqiang, Fidan, Barış and Anderson, Brian D.).

Im Allgemeinen eignet sich GNSS nicht, um in geschlossenen Räumen Positionsbestimmung durchzuführen, da die ausgestrahlten elektromagnetischen Signale durch Wände, Dächer

und andere Gegenständen abgeschwächt werden. Allerdings gibt es mehrere Ansätze, bei denen GNSS zusammen mit Inertialsensoren betrieben werden.

2.5 Lokalisierung durch Inertialsensoren

Eine Möglichkeit zur Ortbestimmung in geschlossenen Räumen ist die Nutzung der Inertialsensoren. Deren Hauptvorteil liegt oft in ihrer Unabhängigkeit der Ortungssignale aus der Umgebung, da sie referenzlos betrieben werden. Im Gegensatz zu anderen Lokalisierungsmöglichkeiten, wie zum Beispiel dem GPS, welches abhängig von einem Satelliten ist, haben sie einen unbegrenzten Arbeitsbereich. Auf Grund ihrer portablen und kleinen Bauform sind sie preiswert. In diesem Kapitel werden die bekannten Möglichkeiten zur Lokalisierung der Personen durch die Inertialsensoren näher beschrieben.

2.5.1 Trägheitsnavigationssystem (INS)

Das Trägheitsnavigationssystem ist ein durch Sensoren betriebenes Navigationssystem, welches die Messung der Bewegungen eines Körpers erfasst und dadurch die Position des Körpers ermittelt. Da die Erfassung der Bewegung durch die Trägheitssensoren erfolgt, wird das Verfahren Trägheitsnavigationssystem (engl. kurz INS Inertial Navigation System) genannt. Zu Ermittlung einer Positionsänderung werden Beschleunigungsmesswerte eines Körpers zweifach integriert (vgl. Ojeda, Lauro and Borenstein, Johann; Wu, Yuanxin, et al, 2005; Yun, Xiaoping, et al, 2007). Der Vorgang der zweifachen Integration von Messwerten wird in realen Navigationsanwendungen, wie zum Beispiel in Flugzeugen, Unterseebooten und Raketen, eingesetzt. Hierfür werden hochwertigere und genauere Inertialsensoren eingesetzt. Die Inertialsensoren werden in der Fußgängernavigation meistens am Fuß getragen, wodurch die Beschleunigungen der Bewegung besser messen lassen.

Ein bekanntes Verfahren in diesem Bereich ist der Strapdown-Algorithmus, bei dem die Auswertung der Inertialsensormesswerte zu der Bestimmung der Lage-, Positions- und Geschwindigkeitsänderung führen (vgl. Meister, O.)

2.5.2 Koppelnavigation

Eine stufenweise Bestimmung der Position aus Geschwindigkeit und Bewegungsrichtung, wird als Koppelnavigation oder Koppelung (engl. dead reckoning) bezeichnet. Bei diesem Verfahren wird die Position durch Zusammensetzung der Komponenten Schritterkennung, Schrittlänge und Schrittrichtung berechnet. Die Qualität der Sensoren spielen keine große Rolle bei der Koppelnavigation (vgl. Schmid, Johannes). Im Folgenden werden die für diese Arbeit relevanten Verfahren beschrieben.

Schritterkennung

Um die Schritterkennung besser nachzuvollziehen, wird zuerst die menschliche Schrittweise beschrieben. Das Gehen basiert auf einer sich wiederholenden Abfolge gleichzeitiger Bewegungen, welches in einzelne Zyklen als rhythmischer Vorgang aufgeteilt wird. Dieser basiert auf den Phasen Fuß Aufsetzphase (engl. Heel-touch-down phase) und Schwungphase (engl. Swing phase). Um Schritte zu erkennen, werden die zwei Phasen der Bewegung detektiert. Die Schwungphase wiederum besteht aus zwei weiteren Phasen. Die Abbildung [2.4](#) stellt die Bewegung eines Fußes dar.

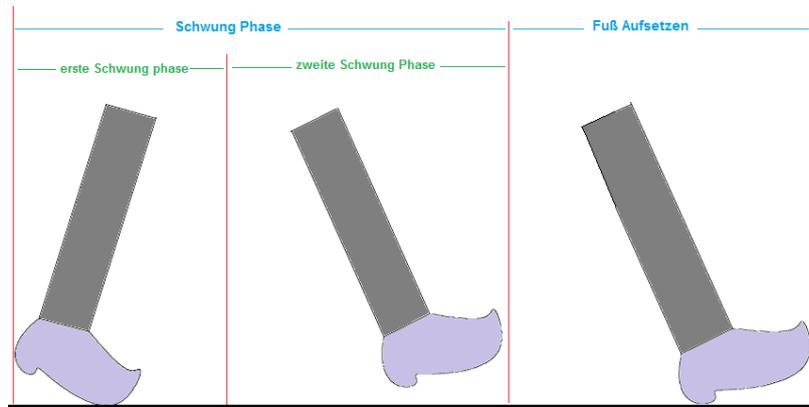


Abbildung 2.4: Einen Laufverhalten des Beines

In der ersten Schwungphase befindet sich der Fuß hinter und in der zweiten Schwungphase vor dem Schwerpunkt des menschlichen Körpers. Der Fuß beschleunigt sich während der Schwungphase. Diese Beschleunigung lässt sich mit Beschleunigungssensoren messen. Die gemessenen Beschleunigungswerte sind bei der Koppelnavigation die Eingangssignale für die Erkennung des Schrittes. Die Messwerte werden anhand definierter Schwellwerte ausgewertet, dies führt zu Erkennung der Schrittphasen (vgl. Cho, Seong Y. and Park, Chan G.).

Schrittlänge

Das Wort „Schritt“ ist in der deutschen Literatur doppeldeutig. Es bezeichnet den Abschnitt vom Abheben bis zum Aufsetzen des Fußes auf den Boden (Einzelschritt engl. step). Eine andere Bedeutung ergibt sich aus den Zeitabschnitt zwischen Aufsetzen eines Fußes bis der nochmalige Aufsetzung desselben Fußes (Doppelschritt engl. stride) (vgl. Runge, Martin and Rehfeld, Gisela).

In Abbildung 2.5 sind der Einzel- und Doppelschritt dargestellt. Um die Strecke zu berechnen, wird durch Multiplikation der Anzahl der gegangenen Schritte mit der geschätzten Länge eines Schrittes berechnet (vgl. Kim, Jeong W., et al; vgl. Jahn, Jasper, et al).

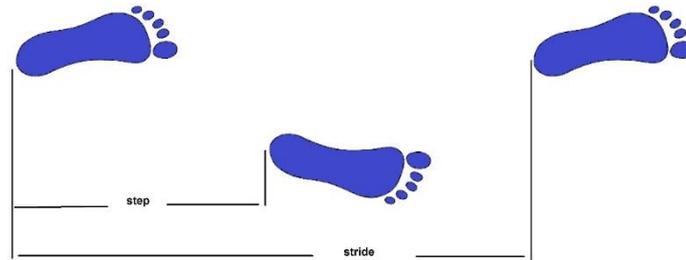


Abbildung 2.5: Schrittlänge

Schrittrichtung

Mit der Schrittrichtung ist hier die Richtung einer Person beim Gehen gemeint. Im Allgemeinen werden die Ausrichtungen in $^{\circ}$ Grad-Einheit berechnet. Die Drehraten der Gyroskope werden für die Schrittrichtung als Eingangssignal verwendet. Da sich die Richtungswinkel aus der Summe der Drehraten ergeben, kommt es nach einer Zeit zu einer großen Differenz der absoluten Ausrichtung. Grund dafür liegt an den akkumulierten Fehlern des Sensors. Die Ausrichtung wird aus einer Kombination von mehreren Sensoren ermittelt, welche häufig aus den Sensoren Gyroskop, Magnetometer und Beschleunigungssensor besteht (vgl. Abyarjoo, Fatemeh, et al; Iffländer, Lukas and Gageik, Dipl-Ing N.).

2.6 Map Matching

Map Matching, Kartenanpassung oder auch Kartenvergleich ist ein Prozess, indem die berechnete Position einer Person mit den Ortsinformationen, zum Beispiel einer digitalen Karte, abgeglichen wird (vgl. Gubi, Kurt). Diese Überprüfung wird häufig in Navigationssystemen verwendet. Da die Lokalisierung bei dem Koppelnavigationssystem auf relativen Messungen der Sensoren beruht, welche durch Verzerrungen und Drift nach längerer Zeit zu wachsenden Fehlern führt, wird mit Hilfe des Map Matching durch einen Abgleich mit einer zugrunde liegenden Karte ein besseres Resultat erreicht.

In Abbildung 2.6 wird das Map Matching anhand der Tür-Korrektur, die zur Anpassung der Trajektorien (Splines) in einem Gebäudemodell führt, dargestellt. Hierbei wird ein nicht erlaubter Raumwechsel identifiziert und entsprechend werden die Trajektoriesegmente in die nächstmögliche Position verschoben. Die roten Punkte auf der Abbildung sind jeweils berechnete Positionen einer Person. Hierbei wird der unerlaubte Raumwechsel identifiziert

und die Trajektoriestegmente werden entsprechend auf die neue Position verschoben, wobei die blauen Punkte korrigierte Trajektoriestegmente darstellen.

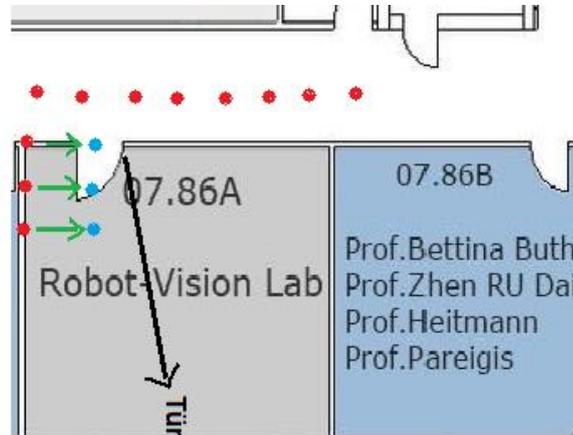


Abbildung 2.6: Eine Tür-Korrektur

2.7 Kalibrierung eines Lokalisierungssystems

Schritte bei den verschiedenen Personen sind unterschiedlich, wodurch auch unterschiedliche Sensordaten gemessen werden. Um das Lokalisierungssystem zu verbessern und präzisere Resultate zu erreichen, wird eine Kalibrierung des Systems benötigt. Um einen Schritt zu erkennen, wird meistens das Muster des Schrittes detektiert. Dazu für ist die Festlegung der Schwellwerte nötig. Genauere Schwellwerte führen zu einem präziseren Resultat. Um die Kalibrierung durchzuführen, werden in einem Zeitfenster mehrere Schritte gegangen. Dadurch werden die Sensordaten gesammelt und durch eine Analyse der gemessenen Werte werden benötigte Variablen festgelegt. Zum Beispiel werden durch die Kalibrierung die Schwellwerte eines Schrittes festgelegt, die bei einer Schritterkennung benötigt werden.

3 Vergleichbare Lokalisierungsverfahren

In folgendem Kapitel werden die relevanten Verfahren für die Lokalisierung in Gebäuden, welche allein durch die logische Auswertung der Messwerte bestehen, vorgestellt. Weiterhin werden Ortungsverbesserungsmethoden für die Verbesserung der Lokalisierung in den Innenräumen vorgestellt, welche in die zwei Bereiche Filterung der Messwerte und Map Matching aufgeteilt sind.

3.1 Lokalisierung durch Inertialsensoren

Wie bereits in dem Kapitel 2.5 vorgestellt, werden die Verfahren für die Lokalisierung durch Inertialsensoren in die zwei Bereiche Inertial- und Koppelnavigationssysteme geteilt, deren relevanten Ansätze nachfolgend vorgestellt werden.

3.1.1 Inertialnavigation

In dem Gebiet der Inertialnavigation werden durch doppelt integrierte Beschleunigungen die Positionsänderungen erfasst. Die Richtung wird durch Integration der Drehraten bestimmt. Yuanxin Wu und seine Kollegen beschreiben in ihrer Arbeit (vgl. Wu, Yuanxin, et al), dass das doppelte Integral ein höchst präzises und effizientes mathematisches Werkzeug ist, um die Rotation und die Translation der Gesamtbewegung eines Körpers gleichzeitig darzustellen. Allerdings wird hierfür ein Beschleunigungssensor und ein Gyroskop mit einer festen Frequenz von 100 Hz verwendet (vgl. Wu, Yuanxin, et al). Dieses Verfahren bezeichnet sich als Strapdown-Algorithmus. Hierzu beschreibt Johannes Schmid (vgl. Schmid, Johannes) einen Vereinfachten Strapdown-Algorithmus. Um dieses Verfahren nachvollziehen zu können, werden im Folgenden Formeln beschrieben. Formel [3.1](#) beschreibt den Zusammenhang zwischen der Beschleunigung \vec{a} , der Gravitation \vec{g} und der Position \vec{x} .

$$\vec{a} = \frac{d^2 \vec{x}}{dt^2} - \vec{g}. \quad (3.1)$$

Die Navigationsgleichung (Formel 3.2) ergibt sich durch Umformung der Formel 3.1

$$\frac{d^2 \vec{x}}{dt^2} = \vec{a} + \vec{g}. \quad (3.2)$$

Geschwindigkeit \vec{v} und Position \vec{x} lassen sich durch Integration über die Zeit berechnen. Um eine Positionsberechnung zu ermöglichen, muss diese doppelte Integration richtungs- und koordinatensystemabhängig erfolgen. Die Differentialgleichung (Formel 3.3) wird aus den Drehraten $\vec{\omega}_{bn}^b = [p, q, r]^T$ aufgestellt, um die zeitliche Änderung der Lage \dot{C}_b^n des Körperfestes Koordinatensystem (b-frame) zum Navigationskoordinatensystem (n-frame) zu ermitteln.

$$\dot{C}_b^n = C_b^n \cdot \Omega \quad \text{mit} \quad \Omega = \begin{pmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{pmatrix} \quad (3.3)$$

Die Richtungskosinusmatrix C_b^n (engl. direction cosine matrix, kurz DCM) transformiert den Körperrahmen zum Navigationsrahmen. Abbildung 3.1 stellt den Vereinfachten Strapdown-Algorithmus von Johannes Schmid dar.

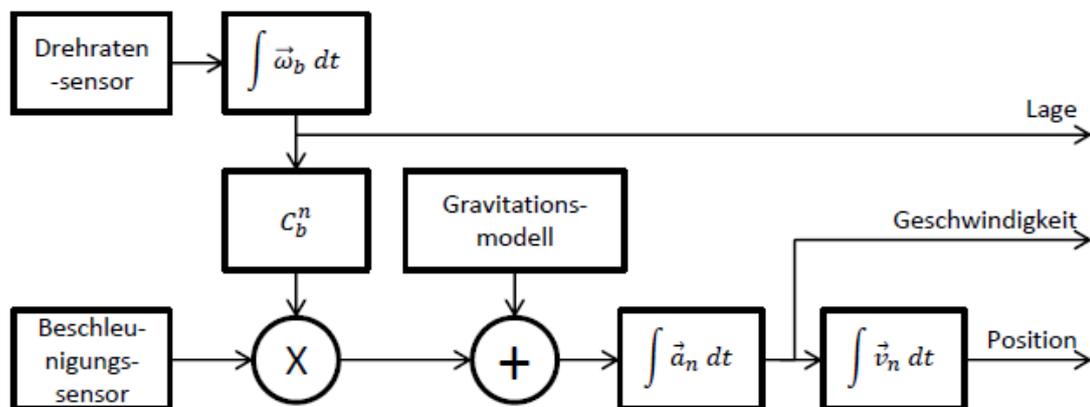


Abbildung 3.1: Vereinfachter Strapdown-Algorithmus von Johannes Schmid (vgl. Schmid, Johannes)

Aus der Integration der Drehraten $\vec{\omega}$ resultiert die Lage C_b^n , diese wird durch die Beschleunigungen \vec{a}_b vom b-frame in das n-frame transformiert. Wonach abziehen der Erdbeschleunigung \vec{g} durch doppelte Integration Geschwindigkeit \vec{v} und Position \vec{x} berechnet werden (vgl. Schmid, Johannes).

Ein weiteres Verfahren in Bereich der Inertialnavigation ist als Zero Velocity Update (kurz ZUPT) bekannt. Hierbei werden resultierende Fehler aus der doppelten Integration der Beschleunigungsdaten reduziert. ZUPT ist bei der Personenlokalisierung ausschließlich beim Tragen der Inertialsensoren am Fuß verwendbar, da ZUPT die Feststellung der Fuß Aufsetzphase benötigt um eine Null-Geschwindigkeit festzustellen (vgl. Park, Sang K. and Suh, Young S.; Beauregard, Stéphane).

3.1.2 Koppelnavigation

Der andere Bereich der Lokalisierung durch Inertialsensoren ist die Koppelnavigation. Hierbei wird die aktuelle Position der Person fortlaufend durch Berechnung von Schrittzahl, Richtung und Schrittlänge ermittelt. Im Vergleich zur Inertialnavigation ist die Güte des Sensors kein entscheidender Faktor. Zumeist wird für die Schritterkennung ein Beschleunigungssensor verwendet (vgl. Kim, Jeong W., et al; Grunau, Sina and Springer, Ing T.; Mercer, JA, et al). Es gibt auch einen anderen Ansatz, bei dem ein Gyroskop für die Schritterkennung eingesetzt wird (Park, Sang K. and Suh, Young S.). Für die Richtungsschätzung wird meistens ein Gyroskop oder ein Magnetfeldsensor verwendet (vgl. Gusenbauer, Dominik, Isert, Carsten and Krosche, J.). Im Folgenden werden die relevanten Ansätze der Komponenten des Koppelnavigationssystems vorgestellt.

Schritterkennung

Um ein Schritt zu erkennen, verwendet man oft die Beschleunigungssignalmuster (vgl. Gusenbauer, Dominik, Isert, Carsten and Krosche, J.; vgl. Cho, Seong Y. and Park, Chan G.; vgl. Kim, Jeong W., et al), welches auf der Abbildung [3.2](#) dargestellt ist.

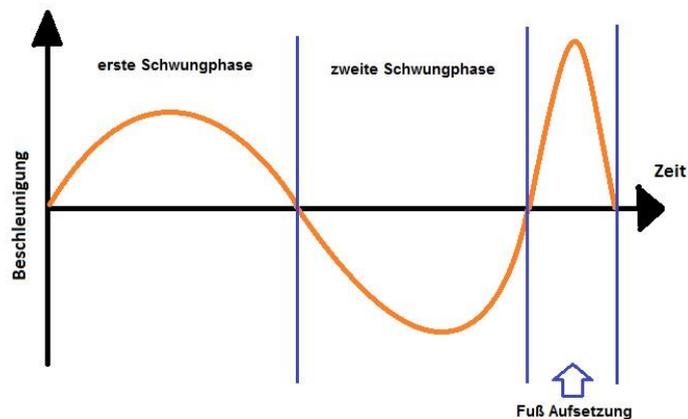


Abbildung 3.2: Beschleunigungsdaten eines Schrittes

In verschiedenen Verfahren werden Zustandsmaschinen verwendet, womit ein Schritt detektiert wird. Wanner nutzt einen nichtdeterministischen endlichen Automaten (engl. Nondeterministic finite automaton, kurz NEA). Diese Automaten versucht die Erkennung der

Zeichen in einer bestimmten Reihenfolge. Hierfür werden die Daten des Beschleunigungssensors als Eingangssignal verwendet.

Abbildung 3.3 zeigt die Phasen der Schritterkennung, die von Wanner (Wanner, Nico) dargestellt wurde. Die Amplituden der Beschleunigungs- und Verzögerungspeaks P_{acc} und P_{dec} , sowie die Zeit R_1 , R_2 und R_3 dienen für die Erkennung des Schrittes. R_1 ist die Anfangszeit, bei der das Bein nach vorne bewegt wird, womit die Schwungphase begonnen hat. R_2 ist der Zeitpunkt für die Beendigung der ersten Schwungphase und Anfang der zweiten Schwungphase. In R_3 endet die gesamte Schwungphase.

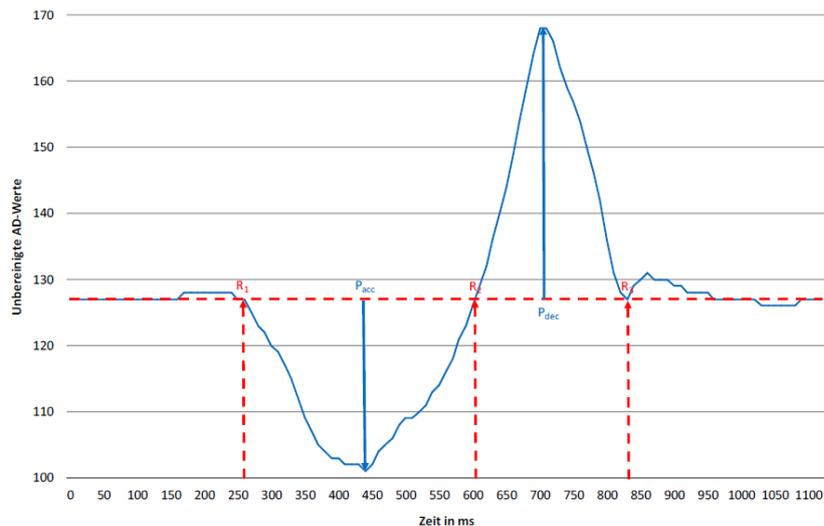


Abbildung 3.3: Die Schritterkennung Phasen (Wanner, Nico)

Abbildung 3.4 zeigt den endlichen Automaten von Wanner, welches ein Schrittmuster erkennen kann. Der Automat besteht aus einem Startzustand, der als Kreis mit doppelter Umrandung dargestellt wird, einem Endzustand, der als Kreis mit stärkerer Umrandung dargestellt wird und mehreren Folgezuständen. Jeder Zustand hat nur einen möglichen Übergang. Beim ϵ -Übergang (Epsilon) wechselt der Zustand direkt in den nächsten Zustand ohne ein Zeichen zu lesen. Nach Erkennung des jeweiligen charakteristischen Punktes, wechselt der Zustand zum Nächsten. Alle Variablen und Register werden nach Erkennung eines Schrittes zurückgesetzt. Der Automat wurde in verschiedener Gehgeschwindigkeit mit den verschiedenen Personen getestet. Dabei war zu beachten, dass bei jeder Person eine Kalibrierung durchgeführt werden musste und folglich eine neue Festsetzung der Schwellwerte (obere Wert P_{acc} untere Wert P_{acc}) notwendig war. Die Fehlerrate lag durch Verwendung des Automaten bei 20% (vgl. Wanner, Nico).

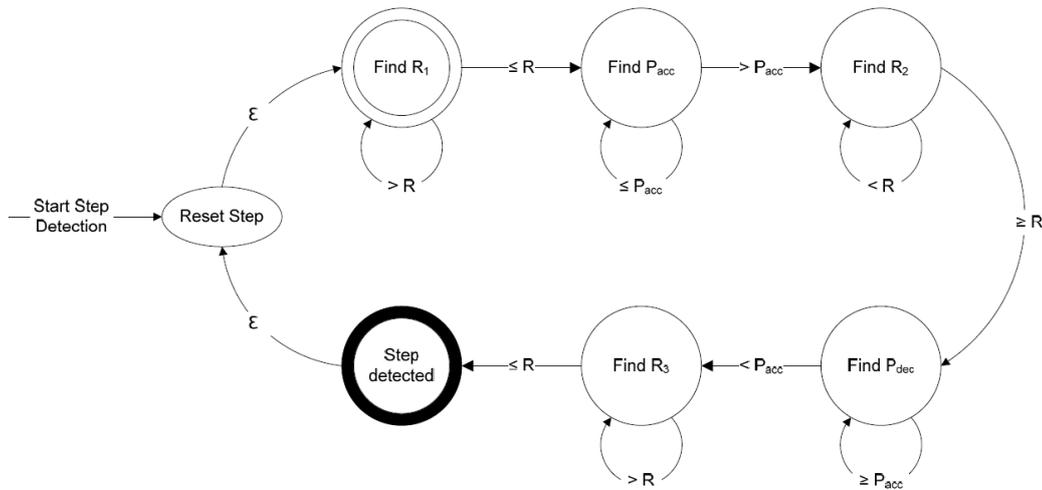


Abbildung 3.5: Ein nichtdeterministischer endlicher Automat (Wanner, Nico)

Ein ähnliches Verfahren verwendet Jeong Won Kim und seine Kollegen (vgl. Kim, Jeong W., et al) in ihrer Arbeit. Drei nacheinander erkannte Phasenmuster führen zu einer Schritterkennung. Die Phasen werden durch Messung der vertikalen und horizontalen Beschleunigung detektiert. In [Abbildung 3.5](#) wird die vertikale (h) und horizontale Beschleunigung (a) sowie die Erdbeschleunigung (g) gezeigt.

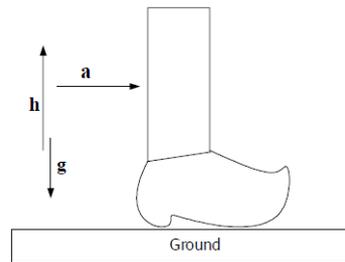


Abbildung 3.4: Horizontal- und Vertikal Beschleunigung

Die horizontale und vertikale Richtungsbeschleunigung während der Schwungphase lassen sich durch die Formel [3.4](#) berechnen, wobei $\theta(t)$ Neigungswinkel des Beins zum Zeitpunkt t ist.

$$\begin{aligned} A(t)_{H-direction} &= (h - g) \sin \theta(t) + a \cos \theta(t) \\ A(t)_{V-direction} &= (h - g) \cos \theta(t) - a \sin \theta(t) \end{aligned} \quad (3.4)$$

In vielen Untersuchungen wird festgestellt, dass ein Schritt erkannt wird, sobald $A(t)_{H-direction}$ oder $A(t)_{V-direction}$ größer als der festgelegte Schwellwert ist. Da $\theta(t)$

abhängig von den Eigenschaften des Beins einer Person ist, ist es schwierig den genauen Schwellenwert zu bestimmen. Wobei ein unkorrekter Schwellenwert zu der Erkennung eines falschen Schrittes führt. Durch die Verwendung der Signalmuster der Beschleunigung wird dieses Problem gelöst.

Es wird in Abbildung 3.6 die Bewegung des Beins während der ersten und zweiten Schwungphase dargestellt.

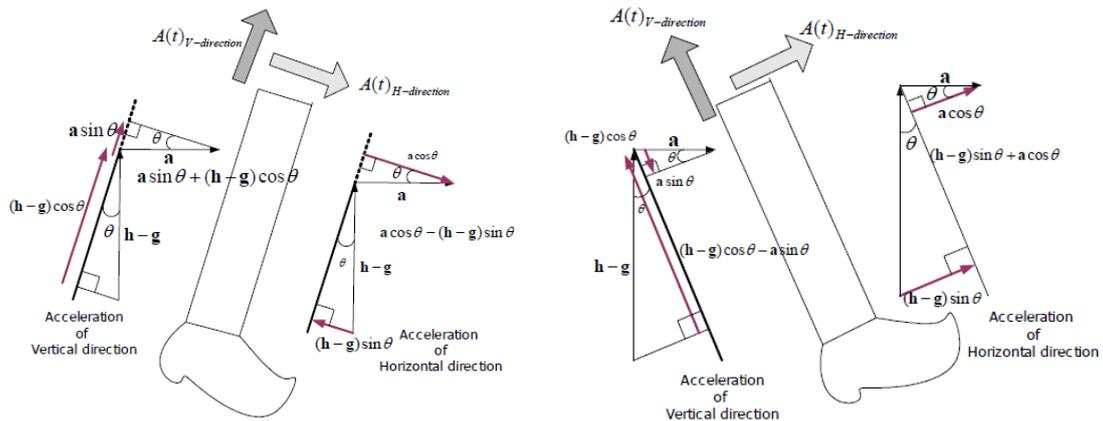


Abbildung 3.6: Erste und zweite Schwungphase (vgl. Kim, Jeong W., et al)

Dieses Verfahren ist einer der robustesten und zuverlässigsten Schritterkennung-Verfahren, da die Fehler, die durch Falscherkennung eines Schrittes durch Springen, Sitzen, Treten oder Drehen verursacht werden, reduziert werden. Erzielt wurde bei Experimenten eine Fehlerrate von ca. 1% (vgl. Kim, Jeong W., et al).

Ein anderes Verfahren für die Schritterkennung verwenden Seong Yun Cho und Chan Gook Park (vgl. Cho, Seong Y. and Park, Chan G.). Es basiert auf der Phasenerkennung eines Schrittes, bei der durch die Festsetzung des Schwellwertes und der anschließenden Überprüfung des Beschleunigungswertes anhand des Schwellwertes durchgeführt wird. Um eine mehrfache Feststellung desselben Schrittes zu vermeiden, findet die Überprüfung in einem vorgegebenen Signalfenster statt. Sowohl die Schwellwerte als auch die Fenstergröße können variabel gehandhabt werden. Abbildung 3.7 stellt das Verfahren als Flussdiagramm dar. Das Verfahren wurde durch ein Spaziergang mit 2299 Schritten durchgeführt, bei dem auch 2299 detektiert wurden. Der Test der Schrittdetektion ergab somit eine sehr gute Fehlerrate von 0% (vgl. Cho, Seong Y. and Park, Chan G.).

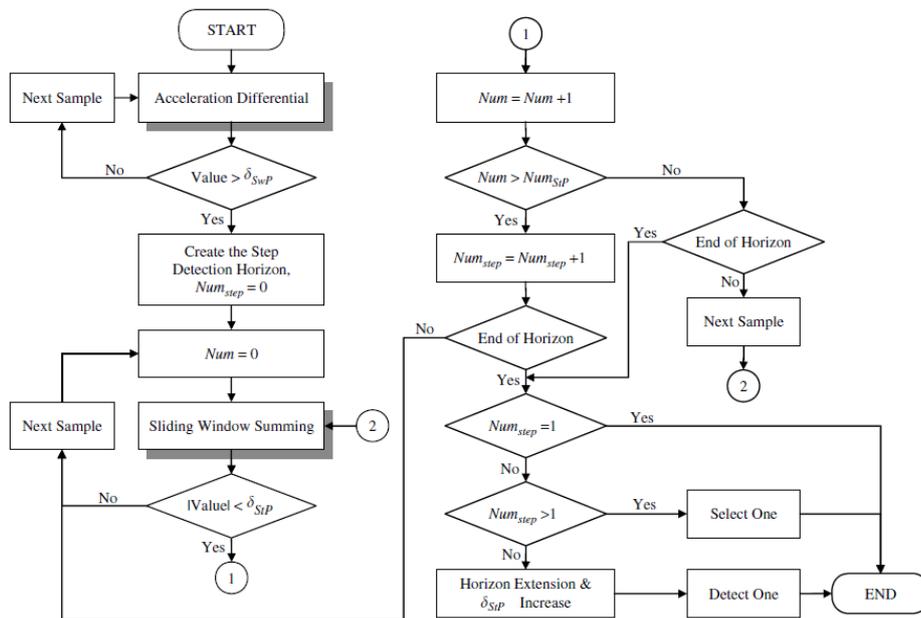


Abbildung 3.7: Flussdiagramm des Schritterkennungs-Algorithmus

(vgl. Cho, Seong Y. and Park, Chan G.)

Schrittlänge

Die Schrittlänge wird durch Abschätzung einer Schrittlänge oder durch die Berechnung der Geschwindigkeit über einen längeren Zeitraum ermittelt (vgl. Jahn, Jasper, et al; vgl. Kim, Jeong W., et al). Generell wird bei der Schrittlängenabschätzung der Beschleunigungssensor eingesetzt.

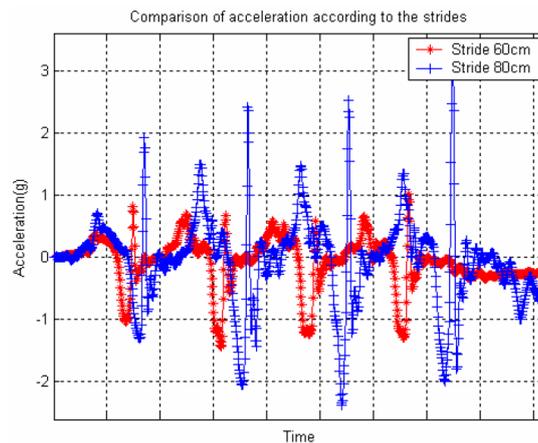


Abbildung 3.8: Das Beschleunigungssignal von 60 cm und 80 cm Schritt (vgl. Kim, Jeong W., et al)

Eine Berechnungsformel für die Schrittlängenabschätzung haben Jeong Won Kim u. a. (vgl. Kim, Jeong W., et al) durch mehreren Tests herausgefunden. Abbildung 3.8 zeigt die Testergebnisse von zwei Personen mit unterschiedlichen Schrittlängen.

In der Abbildung 3.8 wird die Beziehung zwischen der Beschleunigung und der Schrittzahl dargestellt. Hierbei haben die Personen jeweils eine Schrittlänge von 80 cm und 60 cm. Deren Ergebnisse werden tabellarisch in 3.1 und 3.2 dargestellt, welche die Beziehung zwischen der Beschleunigung und der Zeit eines Schrittes zeigen. Deutlich zu erkennen ist, dass die Größe der Beschleunigung eine direkte Auswirkung auf die Länge des Schrittes hat.

Stride	Mean value (g)
60cm	0.2882
80cm	0.5549

Tabelle 3.1: Der Mittelwert des absoluten Beschleunigungswertes (vgl. Kim, Jeong W., et

Stride	Mean of time (sec.)
60cm	0.675
80cm	0.662

Tabelle 3.2: Der Zeitraum eines Schrittes (vgl. Kim, Jeong W., et al)

Die Formel 3.5 (vgl. Kim, Jeong W., et al) wurde aus Resultaten mehrerer Experimente extrahiert. Sie bezieht sich auf die gemessenen Beschleunigungen während des gesamten Gehvorganges. Hierbei wird die Länge eines Schrittes (engl. *stride*) berechnet. Die Fehlerrate liegt bei der Schrittlängenabschätzung durch Verwendung dieser Formel bei unter 5% (vgl. Kim, Jeong W., et al).

$$Stride(m) = 0.98 \times \sqrt[3]{\frac{\sum_{k=1}^N |A_k|}{N}} \quad (3.5)$$

Ähnlich dazu hat Weinberg in seinem Verfahren der geometrischen Überlegung eine Verhältnismäßigkeit zwischen der vierten Wurzel, der Beschleunigungsamplitude während eines Schrittes und der Schrittlänge ausgenutzt (vgl. Weinberg, Harvey). Die zurückgelegte Strecke wird mit der Formel 3.6 berechnet, hierbei sind A_{max} und A_{min} die minimale und maximale Beschleunigung während eines Schrittes. n ist die Anzahl der gegangenen Schritte und K ist die Konstante zur Einheitenumrechnung (z.B. Meter). Durch mehrere Tests mit den verschiedenen Personen wird eine Fehlerrate von ca. 8% festgestellt (vgl. Weinberg, Harvey).

$$Distance \approx \sqrt[4]{A_{max} - A_{min}} \times n \times K \quad (3.6)$$

Schrittrichtung

In den meisten Verfahren zur Fußgängernavigation wird ein Magnetfeldsensor eingesetzt, um die Richtung der Person festzustellen (vgl. Abyarjoo, Fatemeh, et al; vgl. Cho, Seong Y. and Park, Chan G.). Aber die Problematik bei der Nutzung der Magnetfeldsensoren ist die auftretende Störung in Innenräumen. Ein alternatives Verfahren für die Richtungserkennung ist die Bewegungserkennung durch einen an der Hüfte getragenen Beschleunigungssensor. Bei dem von Kourogi und Kurata vorgestellten Verfahren, wird die Bewegungsrichtung durch Analyse der Beschleunigungswerte abgeleitet (vgl. Kourogi, Masakatsu and Kurata, Takeshi).

Ein alternativer Ansatz wäre die Einsetzung eines Gyroskops, womit ein akzeptables Resultat erreicht werden kann. Borenstein u. a. (Borenstein, Johann, Ojeda, Lauro and Kwanmuang, Surat) beschreiben in ihrer Arbeit einen Algorithmus, wodurch mit der Einsetzung des Gyroskops die Richtung berechnet wird, dabei verwenden sie den genannten Ansatz in ihrem eigenen Lokalisierungssystem, das "Personal Dead Reckoning" (kurz PDR) genannt wird. Hierbei wird die Ausrichtung durch die Einsetzung von „Heuristic Drift Reduction“ (kurz HDR) verbessert. Da die HDR-Verbesserung mit der Richtungsberechnung zusammenhängt, wird sie an diese Stelle als eine Einheit vorgestellt.

Das HDR-Verfahren bezieht sich auf die Verringerung von Messfehler des Gyroskops, dabei werden die Daten von der Z-Achse eines Gyroskops als Eingangssignal benutzt. Die gemessenen Drehraten ω_{raw} werden durch die Formel 3.7 dargestellt, wobei ω_{raw} die gemessenen Drehratenrohdaten sind. ω_{true} sind die genauen Drehraten. ε_0 ist der statische Messfehler, der unmittelbar vor einem Spaziergang gemessen und festgelegt wird. ε_d sind die dynamischen Messfehler.

$$\omega_{raw} = \omega_{true} + \varepsilon_0 + \varepsilon_d \quad (3.7)$$

Demnach werden bei jedem Intervall bzw. jedem Schritt die gemessene Abweichung ε_0 von ω_{raw} abgezogen (Formel 3.8).

$$\omega'_{raw} = \omega_{raw} - \varepsilon_0 = \omega_{true} + \varepsilon_d \quad (3.8)$$

Die Azimut bzw. die geänderte Richtung wird durch die Formel 3.9 berechnet.

$$\psi_i = \psi_{i-1} + \omega'_{raw,i} T_i \quad (3.9)$$

Hierbei ist ψ_i die berechnete Richtungsänderung bei jedem Schritt in der Einheit Grad°. T_i ist die Länge des Zeitintervalls in der Einheit Sekunde, welche die Zeit zwischen dem vorherigen Schritt (i-1) und dem aktuellen Schritt (i) zeigt. Abbildung 3.9 zeigt ein Blockdiagramm des HDR-Verfahrens.

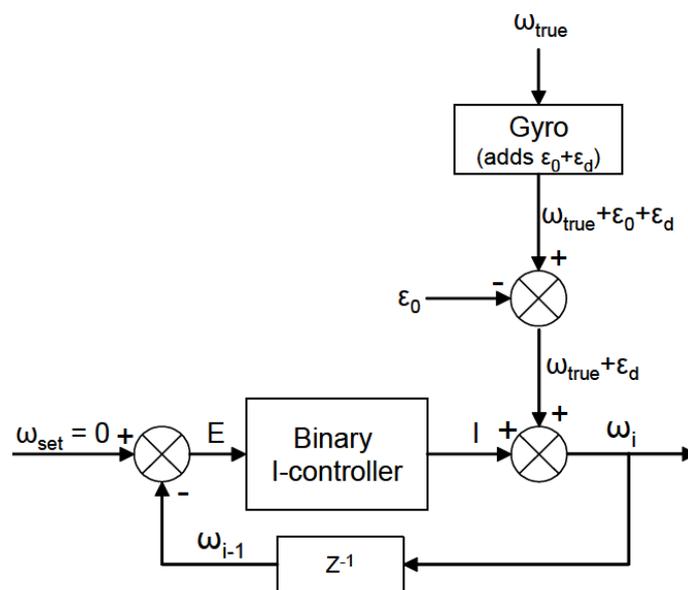


Abbildung 3.9: Der HDR-Algorithmus (Borenstein, Johann, Ojeda, Lauro and Kwanmuang, Surat)

Die grundlegenden Funktionen des HDR-Algorithmus sind wie ein geschlossenes System aufgebaut. Dies unterscheidet es von den meisten anderen Messsystemen, bei denen die Signale des Sensors ungefiltert ausgegeben werden (vgl. Borenstein, Johann, Ojeda, Lauro and Kwanmuang, Surat).

Es wird angenommen, dass die zu lokalisierende Person für eine Zeitlang geradeaus geht, dann soll der Richtungswinkel in dieser Zeit immer unverändert bleiben und ω_{true} ist null. Daher wird der Messfehler ε_d das einzige Ausgangssignal des Gyroskops sein (Formel 3.9). Dies ist für das Regelsystem eine Störung.

Diese Störung wird durch einen Proportional-Integral-Regler (PI-Regler) oder einen Integral-Regler (I-Regler) gegen Null konvergieren. Unter idealen Bedingungen gilt $I \cong -\varepsilon_d$. Da der Sollwert ω_{set} dauerhaft auf null gesetzt ist, gilt folgendes (Formel 3.10):

$$I_i = \begin{cases} I_{i-1} - i_c & \text{for } \omega_{i-1} > 0 \\ I_{i-1} + i_c & \text{for } \omega_{i-1} < 0 \end{cases} \quad (3.10)$$

Hierbei ist i_c die konstante Zahl in der Einheit $^\circ$ Grad. Der Ausgang des Reglers I_i wird mit den Rohmessdaten addiert. Das Resultat ist die korrigierte Drehung ω_i in der Einheit $^\circ$ Grad (Formel 3.11).

$$\omega_i = \omega_{true} + \varepsilon_d + I_i \quad (3.11)$$

Der vorgeschlagene Ansatz (HDR) zeigte nach verschiedenen Experimenten gute Resultate (vgl. Borenstein, Johann, Ojeda, Lauro and Kwanmuang, Surat). Abbildung 3.10 zeigt das Resultat des HDR-Algorithmus in einem Test. Wobei die Teststrecke 1.800 m ist. Der Test dauerte im Durchschnitt 20 Minuten. Hierbei stellt die blaue Linie die durch GPS ermittelten Positionen dar. Die rote Linie zeigt die nicht korrigierten Positionen. Die grüne Linie stellt die durch HDR-Verfahren korrigierten Positionen dar, was eine erhebliche Verbesserung im Vergleich zu einem nicht korrigierten PDR-System zeigt.

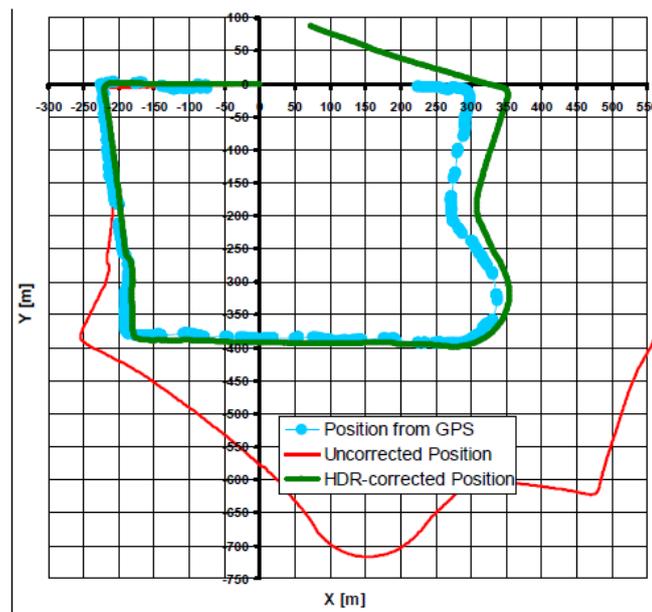


Abbildung 3.10: Test des HDR-Verfahrens (Borenstein, Johann, Ojeda, Lauro and Kwanmuang, Surat)

3.2 Ortungsverbesserung in Gebäuden

Eine hohe Genauigkeit bei der Lokalisierung in Gebäuden ist eine Herausforderung. Die Ansätze, die sich nur auf die Realisierung der Sensordaten verlassen, zeigen kein präzises Resultat. Um die Resultate der Ortung genauer bestimmen zu können, bedarf es hochpräziser Sensoren, die hohe Kosten verursachen. Es wird auch an der Verbesserung bestehender Ortungsansätzen geforscht, wodurch ein präziseres und günstigeres Verfahren entwickelt werden kann. Zwei Entwicklungsrichtungen haben sich innerhalb dieses Forschungsgebietes etabliert. Zum einen liegt der Ansatz darin zusätzliche Hardware zu verwenden, zum anderen liegt der Focus auf die logische Auswertung (vgl. Meyer, Johannes). Bei der Nutzung zusätzlicher Hardware wird meistens ein unabhängiges Lokalisierungssystem als Referenz benutzt. Im Folgenden werden die Verbesserungsverfahren mittels logischer Auswertung in zwei Bereichen vorgestellt. Einerseits ist es die Filterung von Sensordaten, die sich auf die alleinige algorithmische Messwertverbesserung spezialisiert und andererseits konzentriert sich die Verbesserung der Lokalisierung durch Anpassung mit einer Karte (Map-Matching).

3.2.1 Filterung der Messdaten

Die Messdaten spielen eine große Rolle bei der Positionsbestimmung mittels Sensoren. Je genauer die Messdaten sind, desto präziser können Positionen bestimmt werden. Daher wird die Filterung der Messdaten in den meisten Koppelnavigationssystemen angewendet. Voskuhl stellt in seiner Masterarbeit eine Plausibilitätsprüfung vor (vgl. Voskuhl, Sören), bei der die auszuwertenden Daten, zum Beispiel die Rohdaten, die aus dem Beschleunigungssensor und dem Gyroskope gemessen werden, im Vorfeld bearbeitet werden. Dadurch sollen nur die relevanten Daten das System beeinflussen. Die Erkennung von sogenannten Ausreißern, die weit von allen übrigen Werten liegen, ist schwierig (vgl. Voskuhl, Sören). Basierend auf dem arithmetische Mittel (Formel [3.12](#)) wird ein Filter für die Trajektoriestrassen realisiert, welche einen Mittelwert über die aktuelle Position und ihrem direkten Nachbarn bildet.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (3.12)$$

Das mathematische Fachgebiet der Statistik bietet mehrere Ansätzen an, wodurch die Werte als Ausreißer betrachtet werden. Einige davon werden in den Gebieten der Lage- und Streuungsmaße dargestellt. Um eine Maßzahl einer Beobachtungsreihe zu bestimmen, wird der Median verwendet (Formel [3.13](#)) (vgl. Voskuhl, Sören).

$$\bar{x}_{med} = \begin{cases} x_{\frac{n+1}{2}} & \text{falls } n \text{ ungerade} \\ \frac{1}{2} (x_{\frac{n}{2}} + x_{\frac{n}{2}+1}) & \text{falls } n \text{ gerade} \end{cases} \quad (3.13)$$

Allerdings reicht diese Lagemaße nicht, um die Ausreißer zuerkennen. Daher wird die Streuungsmaße mittels der Formel [3.14](#) bestimmt, damit wird die mittlere absolute Abweichung zum berechneten Zentrum beschrieben. Hierdurch ist die Identifizierung der Ausreißer mittels eines Filters möglich.

Einen anderen Ansatz für die Plausibilitätsprüfung der Messwerte verwendet Grubbs in seiner Arbeit (Grubbs, Frank E.). Hierbei werden alle Messwerte eines festgelegten Abstands als Ausreißer erklärt. Es kann aber auch durch Filterung der Sensordaten eine Ortungsgenauigkeit erreicht werden, indem der Messfehler, ein Rauschen bzw. eine Abweichung, im Vorfeld bereinigt wird.

$$d = \frac{1}{n} \sum_{i=1}^n |x_i - \bar{x}_{med}| \quad (3.14)$$

3.2.2 Map Matching

In den meisten Koppelnavigationssystemen wird die Karte als Hilfsmittel verwendet (siehe Kapitel 2.6). Map Matching Verfahren werden schon seit Jahren für die Fahrzeugnavigation verwendet. Ähnliche Verfahren wurden auch für die Personennavigation entwickelt. Durch die Nutzung der Ortsinformationen können präzisere Positionen in den Innen- und Außenorten ermittelt werden. Allerdings ist die Fußgängernavigation im Vergleich zu der Fahrzeugnavigation weitaus komplexer, da sich die Person im Raum frei bewegen kann und die Abstraktion auf ein Knoten- und Kantenmodell ohne Einschränkungen nicht ermöglicht wird.

Einige Verfahren basieren auf Abgleich von Soll- und Ist-Trajektorien (vgl. Link, JA B., et al). Solche Verfahren werden meistens auch für die Fahrzeugnavigation verwendet. Hierbei werden die aufeinanderfolgenden Trajektorien in eine zuvor festgelegte Route angepasst. Dafür sind im Voraus die Festlegung des Start- und Zielpunktes notwendig. Daher ist die Korrektur einer freien Bewegung im Gebäude nicht möglich. Dies führt zu einer erheblichen Einschränkung der Nutzungsmöglichkeit der Lokalisierung.

Häufig wird in den Ortungssystemen mindestens ein Anfangspunkt benötigt, da die aktuelle Position auf einem definierten Anfangspunkt basiert, welches manuell oder durch zusätzliche absolute Ortungsverfahren festgelegt wird. Bei dem zusätzlichen Ortungsverfahren wird im Allgemeinen ein unabhängiges Ortungssystem verwendet, welches meistens auf Funksystemen, wie zum Beispiel WLAN-Fingerprinting, basiert. Spassov entwickelte einen Ansatz zur Anpassung der Trajektorien an ein Gebäudemodell (vgl. Spassov, Ivan), wodurch die Abhängigkeit zu anderen Systemen minimiert und die Autonomie der Anfangspositionsbestimmung ermöglicht wird.

Ein anderer Ansatz wird im Computer Aided Disaster Management System (CADMS) Framework der Technischen Universität Graz vorgestellt (vgl. CADMS TEAM). Das System basiert auf Fusionierung des Beschleunigungssensors und dem Abgleich der Kartendaten. Die Abbildung [3.11](#) stellt die Funktionsweise des Verfahrens da.

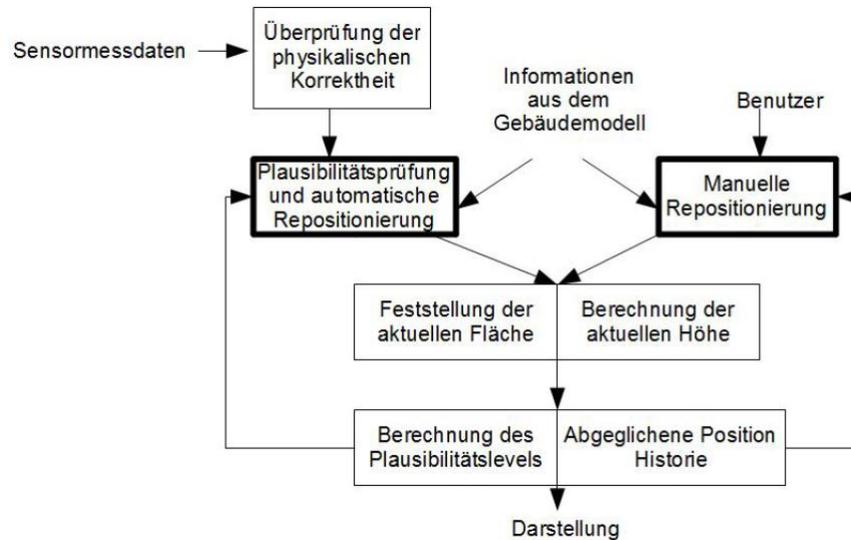


Abbildung 3.11: Ablaufdiagramm des CADMS Frameworks (vgl. CADMS TEAM)

Hierbei werden zuerst die Sensordaten auf die physikalische Richtigkeit kontrolliert und in Koordinaten übertragen. Danach wird durch die Plausibilitätsprüfung der Raumwechsel kontrolliert, wobei versucht wird einen unerlaubten Raumwechsel zu erkennen. Hierfür werden die Gebäudeinformationen, wie beispielsweise die Raumübergänge und die Wände, in eine XML-Datei (Extensible Markup Language) umgewandelt. In der generierten XML-Datei werden die Information als 2D-Polygone dargestellt. Der Plausibilitätsprüfung entsprechend wird dem Positionspunkt eine logische Fläche zugeschrieben, damit erhält man einen logischen Bezug zur aktuellen Position in den Innenräumen. Es besteht auch die Möglichkeit die aktuelle Position der Person Manuell mittels grafischer Benutzeroberfläche auszuwählen, falls die automatisierte Neupositionierung fehlschlägt. Im Fehlerfall geschieht die Korrektur der Trajektorien in zwei verschiedenen Arten. Einerseits wird eine Trajektorie durch den Winkel zur Wand unter Umständen korrigiert. Zum Beispiel wenn eine Trajektorie auf der Wand liegt, dann wird diese so korrigiert, dass sie entlang der Wand ausgerichtet wird. Andererseits kann man auf Grund der vorherigen Daten und naheliegenden Durchgänge darauf Rückschlüsse ziehen, ob ein Raumwechsel stattfand. Wobei bei einem unerlaubten Raumwechsel entsprechend eine Korrektur vorgenommen wird. Bei der Durchgangskorrektur wird das letzte Segment der Trajektorie des Pfades vor dem Raumwechsel und darauffolgende Segmenten neu ausgerichtet (Abbildung [3.12](#)).

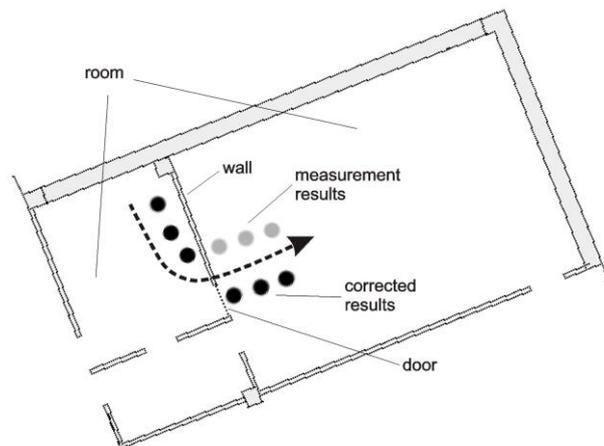


Abbildung 3.12: Durchgangskorrektur-CADMS (vgl. CADMS TEAM)

In der Abbildung [3.13](#) werden durch das CADMS-Frameworks vor und nach Korrekturen der Trajektorien dargestellt. Im ersten Teil der Abbildung [3.13](#) wird die IMU Trajektorie ohne die Korrektur durch das CADMS Verfahren gezeigt, wodurch die Trajektorien quer durch die Räume verlaufen. Die Korrektur der Trajektorien durch die Plausibilitätsprüfung wird im zweiten Teil der Abbildung [3.13](#) gezeigt. Hierbei wird das jeweils letzte Segment der Trajektorie unmittelbar nach einer Wandabschnittsüberschneidung korrigiert.

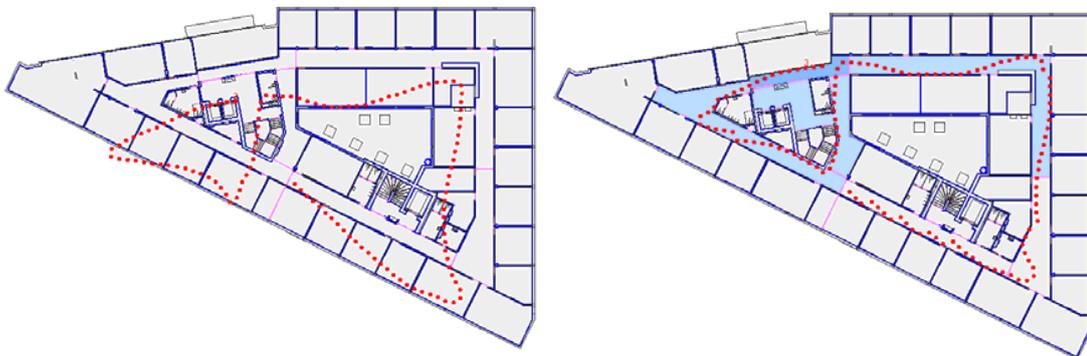


Abbildung 3.13: IMU Pfad vor und nach Map Matching (vgl. CADMS TEAM)

In einem anderen Gebiet der Ortungsverbesserung existieren Ansätze, welche durch den IMU-basierten Partikelfilter zur Positionsverbesserung führen (vgl. Ascher, C., et al; Woodman, Oliver J. and Harle, Robert). In seiner Masterarbeit beschreibt Meyer, dass hierfür ein rekursiver Bayes-Filter implementiert wird, welches auf sequenzielle Monte-Carlo Methoden basiert (vgl. Meyer, Johannes). Hierbei werden die gewichteten

Aufenthaltspunkten, die durch die Wahrscheinlichkeitsbeschreibung des Ortungssystems beschränkt werden, generiert. Im Vorfeld werden diesbezüglich für die Erkennung eines Schrittes mehrere Partikel berechnet, die auf dem vorherigen Schritt durch mehrere Folgezustände und deren Wahrscheinlichkeit basieren.

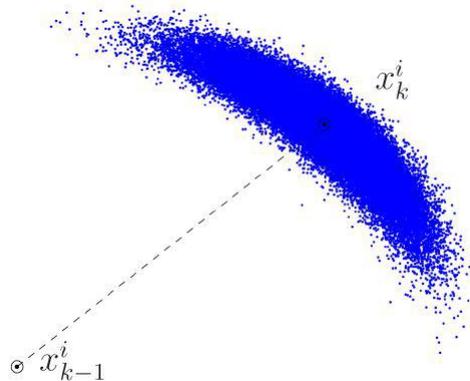


Abbildung 3.14: Partikelwolke für den nächsten Schritt (vgl. Meyer, Johannes)

In der Abbildung [3.14](#) wird die berechnete Partikelwolke für den nächsten Schritt gezeigt. Dabei ist x_{k-1}^i der vorherige Schritt und beschreibt den Ausgangspunkt des i -ten Partikels. x_k^i ist das i -te Partikel des k -ten Schrittes. Ein entscheidender Faktor für eine Verbreiterung der Partikelwolke ist die Bewegungsrichtung. Hingegen sorgt die Schrittlänge für eine Unschärfe bei der Berechnung der Partikelwolke. In diesem Fall können die Partikel noch nicht bewertet werden, da keine Informationen bzw. Einschränkungen des Gebäudemodells vorliegen. Aus der Kombination eines Gebäudemodells und der berechneten Partikel können genauere Aussagen über den Aufenthaltsort getroffen werden, wodurch die Falscherkennung der Trajektorien verringert werden kann und dies führt zu einer Verbesserung des Lokalisierungssystems.

Die erzielbare Genauigkeit bei einem IMU-basierten Partikelfilter wurde von Woodman untersucht (vgl. Woodman, Oliver J. and Harle, Robert). Dessen Testläufe wurden in einem 8725 m² großen Gebäude durchgeführt. In 95% der Fälle zeigten die Tests eine Genauigkeit von 0,73 m (vgl. Woodman, Oliver J. and Harle, Robert), was einer Verbesserung der Lokalisierungsgenauigkeit gegenüber eines nicht durch Partikelfilter korrigierten Systems entspricht.

4 Analyse des entwickelten Lokalisierungssystems

Dieses Kapitel befasst sich mit der Analyse der verwendeten Algorithmen, welche in dem entwickelten Lokalisierungssystem verwendet werden. Die Algorithmen sind nach bestimmten Kriterien ausgewählt. Die Datenmenge bzw. Datenrate der Sensoren ist ein Kriterium, das eine entscheidende Rolle bei der Auswahl spielt. Ein bekannter Ansatz für die Inertialnavigation ist der Strapdown-Algorithmus (Kapitel 3.1.1). Hierfür wird ein präziser Sensor verwendet. Zum Beispiel werden in der Arbeit von Yuanxin Wu u. a. ein Beschleunigungssensor und ein Gyroskop mit einer festen Frequenz von 100 Hz betrieben (vgl. Wu, Yuanxin, et al). Wobei die verwendeten Sensoren in dieser Arbeit höchstens auf eine Frequenz von 10 Hz eingestellt werden können (Kapitel 2.3.4). Aufgrund dessen fällt die Entscheidung auf ein Koppellokalisierungssystem, das auch mit niedrigerer Frequenz betrieben werden kann (Kapitel 2.5.2). Bei einem Koppelnavigationssystem wird die Position schrittweise berechnet. Das System wird meistens durch die Zusammenarbeit mehrerer Komponenten betrieben, wodurch die Ermittlung der aktuellen Position ermöglicht wird. Im Folgenden wird ein Überblick der bestehenden Komponenten des Lokalisierungssystems gegeben.

Abbildung [4.1](#) zeigt eine Übersicht des Systems. Die gemessenen Sensordaten werden durch die BLE (Kapitel 2.2) an die Komponenten gesendet. Nach dem Empfang werden die Daten durch entsprechende Algorithmen ausgewertet. Die Resultate werden an eine weitere Komponente weitergegeben, wobei durch die Zusammensetzung der Resultate die aktuelle Position ermittelt wird. Allerdings wird hierfür auch die Anfangsposition benötigt. Sie wird durch eine Benutzeroberfläche manuell eingestellt. Um ein besseres Resultat zu erreichen, wird die Position anhand einer zusätzlichen Karte angepasst.

Im Folgenden werden die Komponenten einzeln realisiert. Darüber hinaus werden die verwendeten Verfahren anhand der vorgestellten Ansätze (Kapitel 3) erläutert.

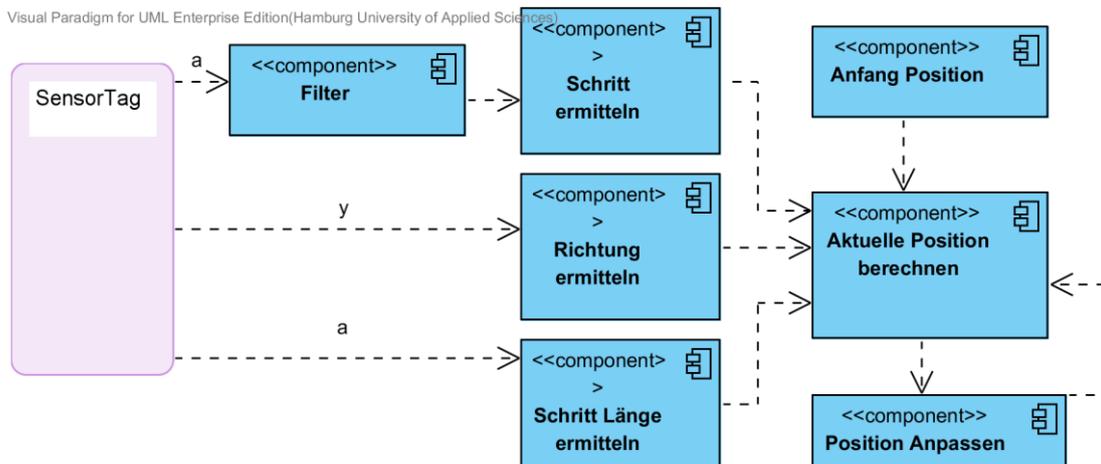
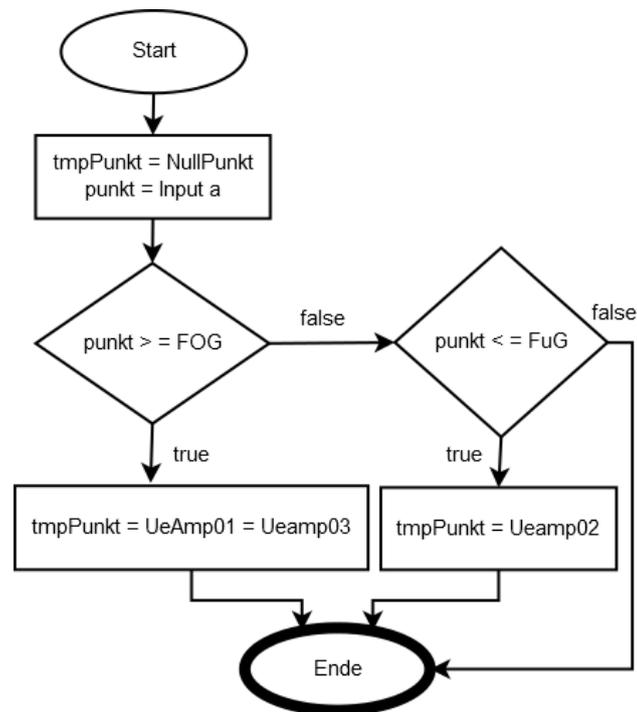


Abbildung 4.1: Übersicht des Systems

4.1 Verwendete Filter für die Beschleunigungswerte

Das entwickelte Lokalisierungssystem bekommt für die Schritterkennung Daten des Beschleunigungssensors als Eingangssignal. Um einen Schritt zu erkennen, werden drei aufeinander folgende Muster identifiziert. Ein Filter wird benötigt, um das Schritterkennungsverfahren effizienter zu betreiben. Bei jedem Schritt werden unterschiedliche Sensordaten gemessen. Zum Vorbereiten der Messwerte wird ein Filter eingesetzt, wodurch die Effizienz des Schritterkennungsmechanismus erhöht wird. Der Mechanismus wird dadurch vereinfacht und die Überprüfungsdauer verkürzt.

Das Ablaufdiagramm [4.1](#) zeigt den verwendeten Filter, hierbei enthält die Variable „punkt“ den aktuellen Beschleunigungswert. „FOG“ und „FUG“ beinhalten den oberen bzw. unteren Filtergrenzwert, die durch kalibrieren des Lokalisierungssystems festgelegt werden (siehe Kapitel 2.7). Nach der Überprüfung werden die Beschleunigungswerte durch einen der drei konstanten „UeAmp01“, „UeAmp02“ und „UeAmp03“ (Über die Amplitude kurz UeAmp) ersetzt. Wenn die Beschleunigungswerte nicht zu den Bereichen gehören, werden sie durch „NullPunkt“ ersetzt. „NullPunkt“ sind die Beschleunigungswerte, die zwischen den Amplituden 01 und 02 liegen.



Ablaufdiagramm 4.1: Verwendete Filter

4.2 Schritterkennung

Das Schritterkennungsverfahren ist ein Bestandteil des Koppelnavigationssystems. In dem Kapitel 3.1.2 wurden drei Ansätze in diesem Bereich vorgestellt. In dem Ansatz von Seong Yun Cho und Chan Gook Park werden durch eine Überprüfung des festgelegten Schwellwertes während des vordefinierten Zeitfensters Schritte detektiert (vgl. Cho, Seong Y. and Park, Chan G.). Einen ähnlichen Ansatz verfolgt Wanner, in dem die Detektion in einem nichtdeterministischen endlichen Automaten stattfindet (vgl. Wanner, Nico). Hierbei werden die entstehenden Amplituden der Beschleunigungs- und Verzögerungspeaks sowie deren Zeitpunkte detektiert. Diese aufeinander folgenden Überprüfungen führen zu einer Erkennung eines Schrittes. In dem dritten Ansatz von Won Kim und seine Kollegen führt die Detektion der Schwungphasen eines Schrittes zu der Schritterkennung (vgl. Kim, Jeong W., et al).

In dieser Arbeit wird ein Ansatz entwickelt, welcher den Ansätzen von Won Kim u. a. (vgl. Kim, Jeong W., et al) und Wanner (vgl. Wanner, Nico) ähnelt. Durch selbst geführte

Experimente wird festgestellt, dass jeder Schritt aus einem ähnlichen Muster besteht. Solche Muster werden anhand der Beschleunigungswerte fünf aufeinander folgender Schritte in Abbildung 4.2 dargestellt.

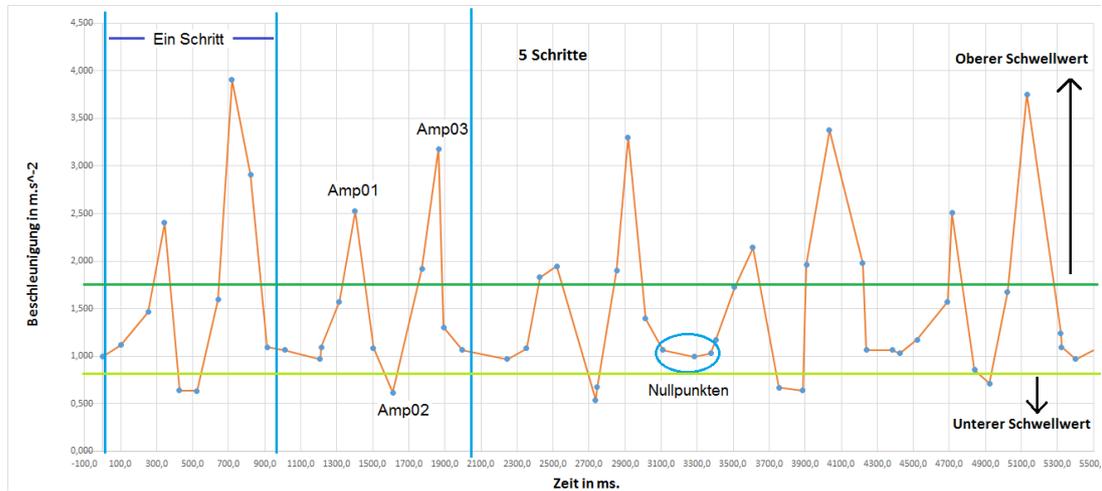
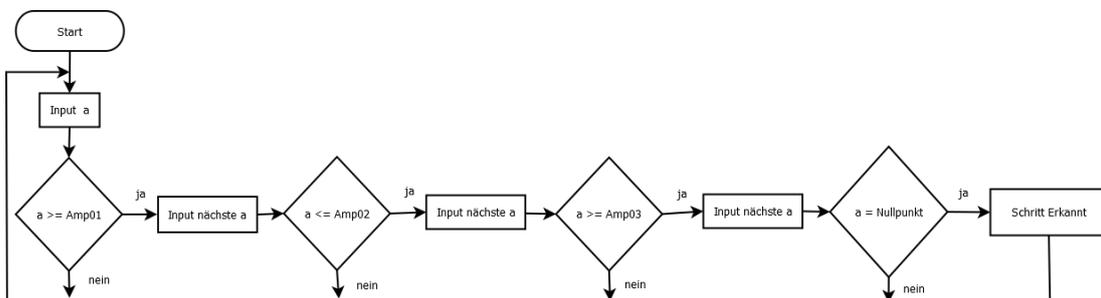


Abbildung 4.2: Schrittmuster

Das Ablaufdiagramm 4.2 zeigt die Grundform des entwickelten Erkennungsalgorithmus eines Schrittes.



Ablaufdiagramm 4.2: Schritterkennung

Um ein besseres Resultat zu erreichen, werden zuerst die gemessenen Sensordaten durch ein Filter (siehe Kapitel 4.1) vorbearbeitet. Nach der Bearbeitung wird jeder gemessene Punkt durch eine Konstante, die entweder der Nullpunkt oder eine der Amplituden ist, ersetzt. Abbildung 4.3 zeigt die gefilterten Beschleunigungswerte fünf aufeinander folgender Schritte.

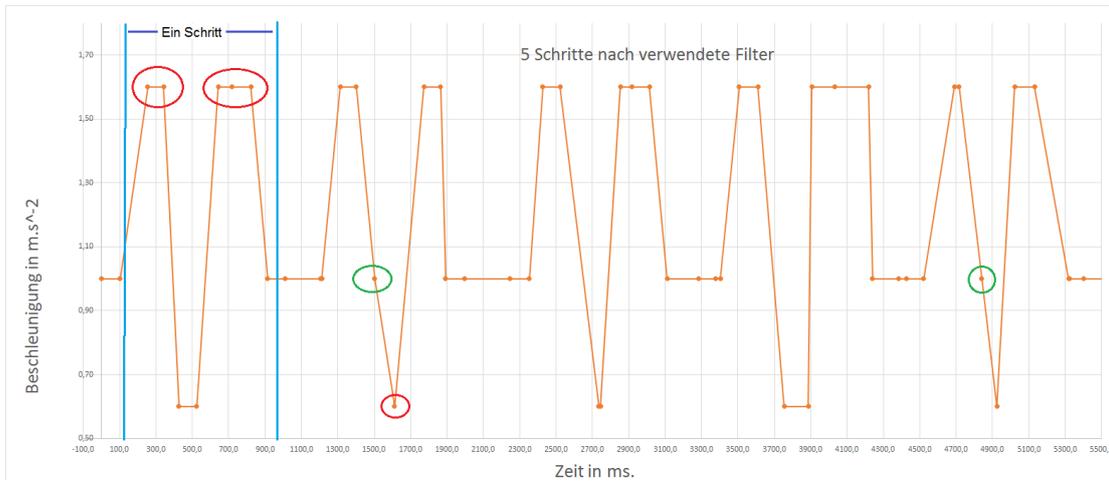


Abbildung 4.3: Schrittmuster nach Filterung

Nach Verwendung des Filters ist es jedoch notwendig Änderungen an dem dargestellten Algorithmus (4.1) vorzunehmen. Dabei entstehen mindesten ein bis höchstens drei Punkte, bei den Amplituden (rote Kreise in Abbildung 4.3). Zwischen der ersten und der zweiten, sowie zwischen der zweiten und der dritten Amplitude kann höchstens ein Nullpunkt (grüne Kreise in Abbildung 4.3) entstehen. Nun wird ein Zustandsautomat für die Schritterkennung entwickelt, bei dem die Amplituden, Start- und Endbereich als Zustände deklariert werden.

Die Zustände während eines Schritts werden in Abbildung 4.4 dargestellt.

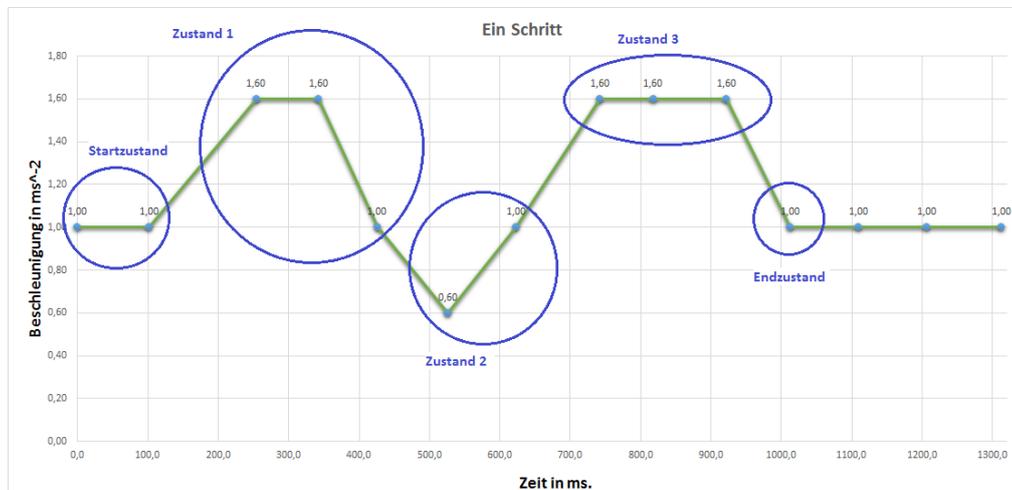
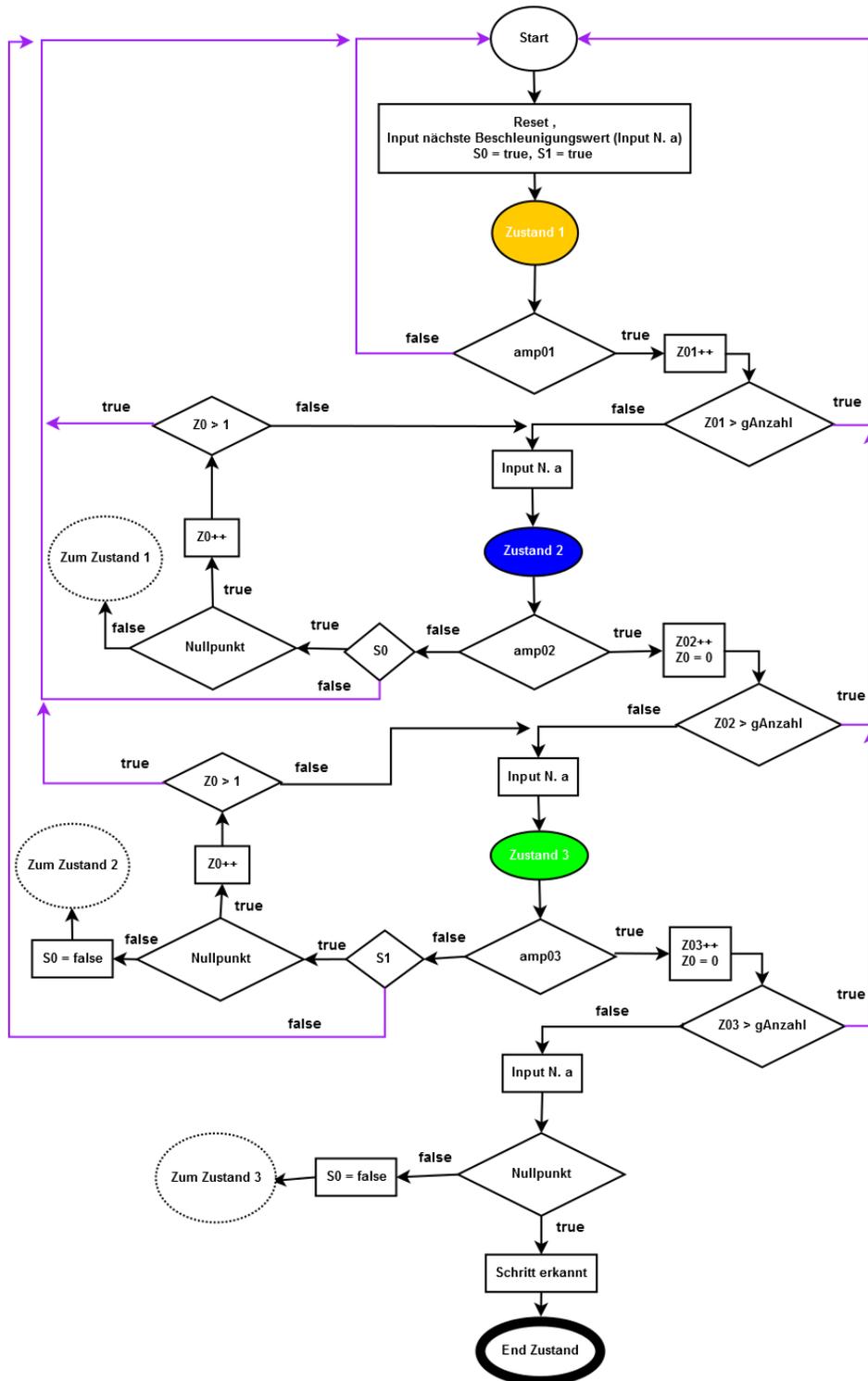


Abbildung 4.4: Zustandsbereiche

Um den Schritterkennungsalgorithmus anzupassen, werden an jedem Übergang zum nächsten Zustand noch zwei Bedingungen geknüpft. Die erste Bedingung prüft nach einem

entstehenden Nullpunkt zwischen zwei Phasen. Die Nullpunkte, die nach der Filterung einem Wert von 1 entsprechen, werden als letzte Punkte in den markierten Zuständen 1 und 2 in der Abbildung [4.4](#) dargestellt. Die Experimente haben gezeigt, dass dieses Ereignis maximal einmal zwischen den Zuständen 1 und 2 bzw. 2 und 3 vorkommt. Es wird sichergestellt, dass dieses Ergebnis mit Hilfe der booleschen Variablen $S0$ und $S1$ höchstens einmal vorkommt (Ablaufdiagramm [4.3](#)). Die zweite Bedingung dient der Erkennung wiederholter Punkte (rote Kreise in Abbildung [4.3](#)), welche durch die Verwendung der Filter entstehen. Allerdings hat die Anzahl der vorkommenden gleichen Punkte eine Beziehung zur Schrittgeschwindigkeit der Person. Je langsamer eine Person geht bzw. je langsamer die Geschwindigkeit des Schrittes ist, desto mehr entstehen solche Punkte bei Zustand 1 und 2. Das Ablaufdiagramm [4.3](#) stellt den verbesserten Schritterkennungsalgorithmus dar. Hierbei ist $gAnzahl$ die Konstante für die wiederholten Punkte in einem Zustand. $Z01$, $Z02$ und $Z03$ sind Zähler für die Wiederholung der Zustände. $Z0$ ist ein Nullpunkte-Zähler.

Um ein genaueres Resultat zu erzielen, ist es notwendig eine Kalibrierung durchzuführen und die Grenzen für die Bestimmung der Amplituden, der benutzten Filter und der maximalen Anzahl wiederholter Punkte in einem Zustand festzulegen.



Ablaufdiagramm 4.3: Verbesserter Schritterkennungsalgorithmus

4.3 Schrittlänge

In dem Kapitel 3.1.2 wurden zwei Möglichkeiten vorgestellt, um die Länge eines Schrittes zu berechnen. Zum einen kann die Strecke berechnet werden, indem über einen längeren Zeitraum die Geschwindigkeit durch einen hochwertigen Beschleunigungssensor gemessen wird (Formel 3.6). Zum anderen kann jede Schrittlänge einzeln berechnet werden (Formel 3.5), wobei nach dem gesamten Gehvorgang die Länge eines Schrittes geschätzt wird. Auf Grund des geringeren Implementierungsaufwandes und der ähnlichen Fehlerrate der vorgestellten Formeln, fällt die Entscheidung auf die Formel 3.6. Durch eine Modifikation der bestehende Formel 3.6 resultiert die Formel 4.1, wodurch die Länge eines Schrittes statt der Strecke berechnet wird. Bei jedem Schritt wird die vierte Wurzel der Differenz zwischen dem Minimum und dem Maximum des Beschleunigungswertes berechnet. Diese wird mit einer Konstanten zur Einheitenumrechnung (zum Beispiel in Meter) multipliziert.

$$\text{Schrittlänge}_{(pixel)} = \sqrt[4]{a_{max} - a_{min}} * P_{Faktor} \quad (4.1)$$

$$\text{Schrittlänge}_{(Meter)} = \sqrt[4]{a_{max} - a_{min}} * M_{Faktor} \quad (4.2)$$

Die Formel 4.1 berechnet die Länge eines Schrittes in Echtzeit. Hierbei sind a_{max} und a_{min} die Maximal- und Minimalbeschleunigungswerte während eines Schrittes und M_{Faktor} eine Konstante zur Einheitenumrechnung in Meter. Die Formel 4.2 ist für die Schrittlängenberechnung in der Einheit Pixel. Die Schrittlänge wird auch in der Einheit Pixel berechnet, da dies für die Darstellung auf Android Geräten notwendig ist.

4.4 Schrittrichtung

Eine Möglichkeit für die Bestimmung der Schrittrichtung ist die Einsetzung eines Magnetfeldsensors. Der Vorteil liegt in der höheren Genauigkeit, wohingegen viele Gegenstände im Innenraum für eine Störung des Magnetfeldsensors sorgen können (siehe Kapitel 3.1.2). Was einen großen Nachteil darstellt. Auf Grund dessen wird von der Nutzung des Magnetometers abgesehen. Eine andere Alternative für die Richtungsbestimmung ergibt sich durch die Nutzung des Gyroskops. Hierbei wird der Winkel kontinuierlich berechnet, indem er auf einander aufsummiert wird. In dieser Arbeit wird der Gyroskope für die Richtungserkennung verwendet. Allerdings wird keine Berechnung des Winkels stattfinden, was natürlich eine erhebliche Einschränkung ist. Stattdessen wird die Umdrehung um die y-Achse in einem bestimmten Schwellwert identifiziert. Der Schwellwert entspricht einer Umdrehung um 90°, da die Richtungsänderung meistens aus einer rechtwinkligen Drehung

in einem Gebäude besteht. Der Grund für solche Drehungen sind die Strukturen der Gebäude, wobei sie fast immer aus geraden Gängen und viereckigen Räumen bestehen. Daher sind die Verbindungen, also die Korridore, fast immer rechteckig. Resultierend aus Experimenten werden ein oberer und unterer Schwellwert festgelegt, wodurch jeweils bei der Überschreitung dieser Schwellwerte eine Umdrehung um 90° erkannt wird. Je nachdem ob die Überschreitung bei dem unteren oder dem oberen Schwellwert passiert, wird die Umdrehung als rechts oder links identifiziert. Eine solche Umdrehung dauert meistens unter einer Sekunde. Die Gyroskope-Werte einer Umdrehung werden mehrmals über den festgelegten Schwellwerten gemessen. Dies führt zu einer falschen Erkennung der Richtung. Um diesen Fehler zu vermeiden, findet die Überprüfung der Werte in einem Zeitfenster statt. [Abbildung 4.5](#) zeigt die gemessenen Drehraten nach links und nach rechts. Hierbei sind die Amplituden in den blauen Kreisen jeweils eine Drehung nach links und die Amplituden in den grünen Kreisen jeweils eine Drehung nach rechts.

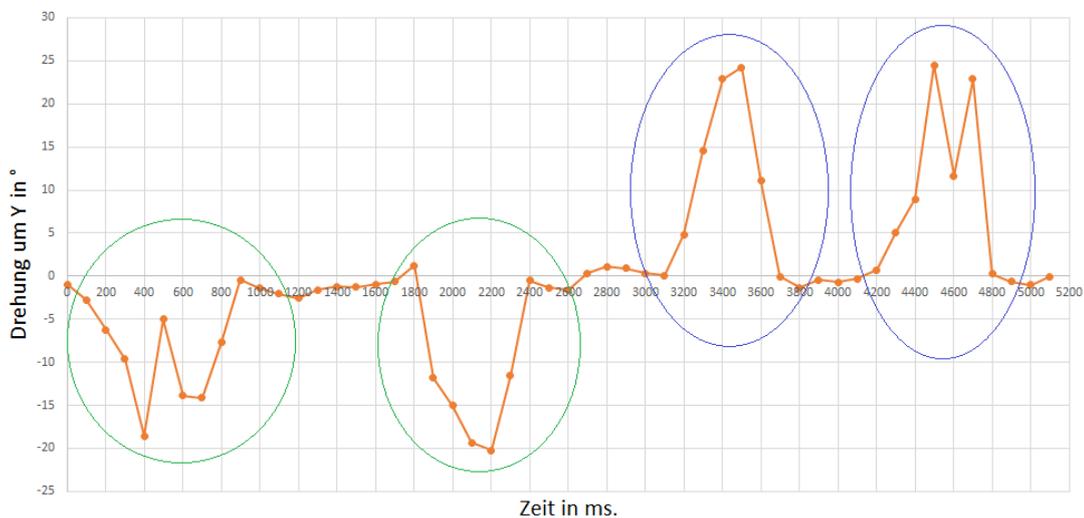
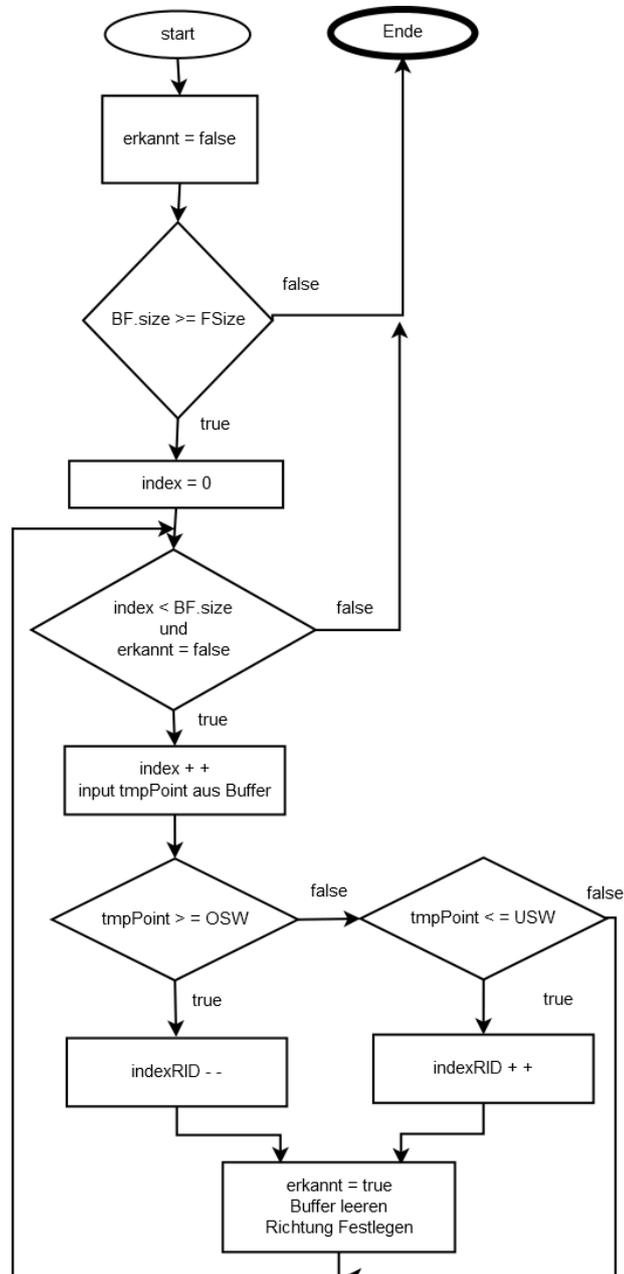


Abbildung 4.5: Drehung um einen rechten Winkel

In Ablaufdiagramm [4.4](#) wird der Richtungserkennungsalgorithmus dargestellt. Hierfür ist die manuelle Bestimmung der Anfangsrichtung notwendig, was durch den Benutzer vor Start festgelegt wird. Die Daten des Sensors werden in einem Buffer *BF* gesammelt, bis die Menge der Zeitfenster *FSize* erreicht wird. Die Schwellwerte *OSW* und *USW* werden durch Kalibrierung festgelegt (Kapitel 2.7). Die *indexRID* ist die aktuelle ID der Richtung. Die IDs der Richtungen sind jeweils ID 0 für NORD, ID 1 für OST, ID 2 für SUED und ID 3 für WEST. Wird zum Beispiel die Startrichtung als Ost (ID 1) festgelegt, so sollte nach einer Umdrehung nach rechts die Richtung Nord identifiziert werden.

Demzufolge wird die gemessene Umdrehung (*tmpPoint*) größer sein als der oberer Schwellwert (*OSW*), was zur Erkennung der Umdrehung nach rechts führt. Durch das Inkrementieren der *indexRID* ergibt sich die *indexRID* mit dem Wert Zahl null (Nord).



Ablaufdiagramm 4.4: Richtungserkennung Algorithmus

4.5 Kartenanpassung (Map Matching)

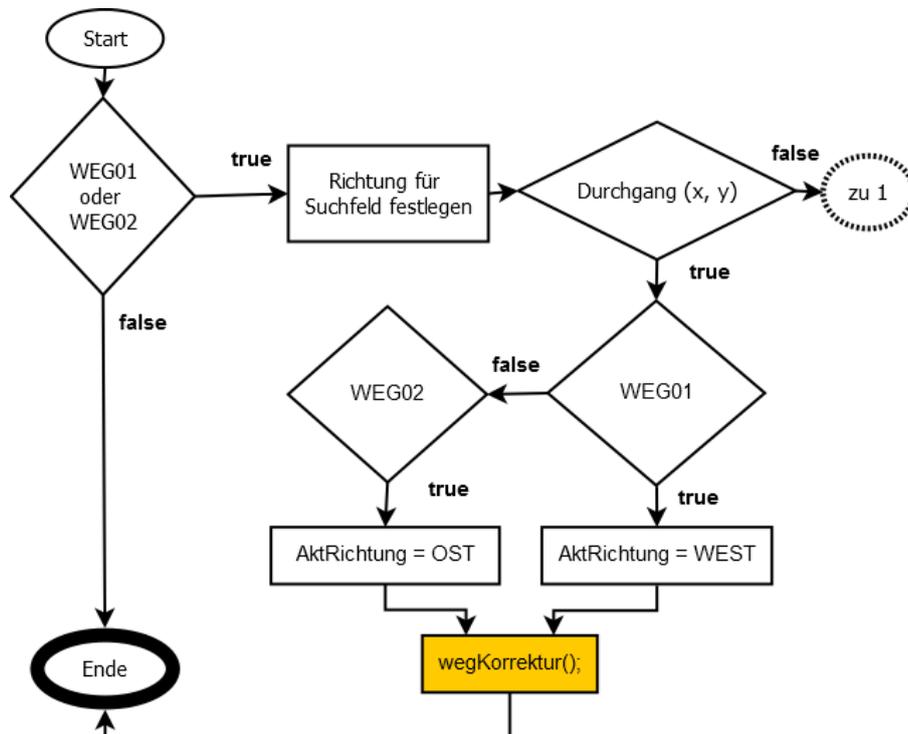
Für die Verbesserung der Lokalisierung im Gebäude wird eine Karte als zusätzliches Hilfsmittel verwendet. Hierbei werden durch die Informationen der Karte (zum Beispiel Türen und Wänden) die Einschränkungen für die mögliche Bewegungsrichtung festgelegt, wodurch eine Korrektur der aktuellen Position ermöglicht wird. Wie schon in dem Kapitel 3.2.2 vorgestellt, existieren mehrere Algorithmen in diesem Gebiet. In einem Ansatz wird die Karteninformation auf ein Knoten- und Kantenmodell abstrahieren, wobei die aktuelle Position mit einem vorher festgelegten Navigationspfad verglichen und entsprechend korrigiert wird. Hierfür werden ein Start- sowie ein Endpunkt benötigt. Die Bewegung wird im Gebäude sehr stark eingeschränkt, da bei diesem Ansatz ein Start- sowie ein Zielpunkt vorausgesetzt wird. Daher wird dieser nicht bei dem entwickelten Lokalisierungssystem verwendet. Bei einem anderen Ansatz wird durch die Berechnung der Partikel die nächste Position ermittelt. Diese Partikeln werden aus der Bewegungsrichtung und den Ortsinformationen berechnet. Da sich die Schrittrichtung in dieser Arbeit nur aus den festen Richtungen Nord, Süd, West und Ost bestehen, wird die Nutzung eines Partikelfilters nicht zu eine Verbesserung bei der Ermittlung der Position führen. Daher wird eine Plausibilitätsüberprüfung (kurz PÜ) auf Basis eines einfachen Bewegungsmodells verwendet, wodurch bei jeder durch den Schritterkennungsalgorithmus ermittelte Position mit der Information der Karte auf Plausibilität geprüft wird. Die gültigen Positionen werden nur zugelassen, wenn sie auf den definierten Bereichen der Karte liegen.

Zusätzlich wird durch ein Verfahren der aufgezeichnete Bewegungspfad an einen Gebäudegrundriss angepasst. Raumwechsel werden nur durch die Durchgänge bzw. die Türen erlaubt und bei einem nicht erlaubten Raumwechsel wird die Trajektorie anhand der Informationen der Karte geprüft und gegebenenfalls angepasst bzw. korrigiert. Bei CADMS-Framework wurde schon ein solcher Ansatz entwickelt. Dabei werden die Raumwechsel nur durch deklarierte Eingänge ermöglicht. Bei einem nicht erlaubten Raumwechsel wird die Trajektorie entsprechend korrigiert. Bei der Korrektur wird die Trajektorie durch die Plausibilitätsprüfung (PÜ) auf der Karte angepasst.

Um die Informationen der Karte benutzen zu können, wird die verwendete Karte manuell digitalisiert. Dabei werden die Flure und Eingänge gemessen, wodurch anhand der Auflösung des verwendeten Gerätes die konstanten Punkte deklariert werden. Der entwickelte Mechanismus wird auf den Ablaufdiagrammen [4.5](#) und [4.6](#) dargestellt.

Der entwickelte Mechanismus für die Kartenanpassung unterstützt nur einen Teil der Anpassung der Position auf der Karte. Im Falle einer nicht plausiblen Bewegung an den genannten Stellen, zum Beispiel beim einer nicht deklarierten Position innerhalb der Karte, wird die Korrektur vorgenommen und der Mechanismus wird gestartet. Nach dem Start des Algorithmus wird zuerst überprüft, ob die zu korrigierenden Punkte auf den definierten Bereichen der Karte liegen.

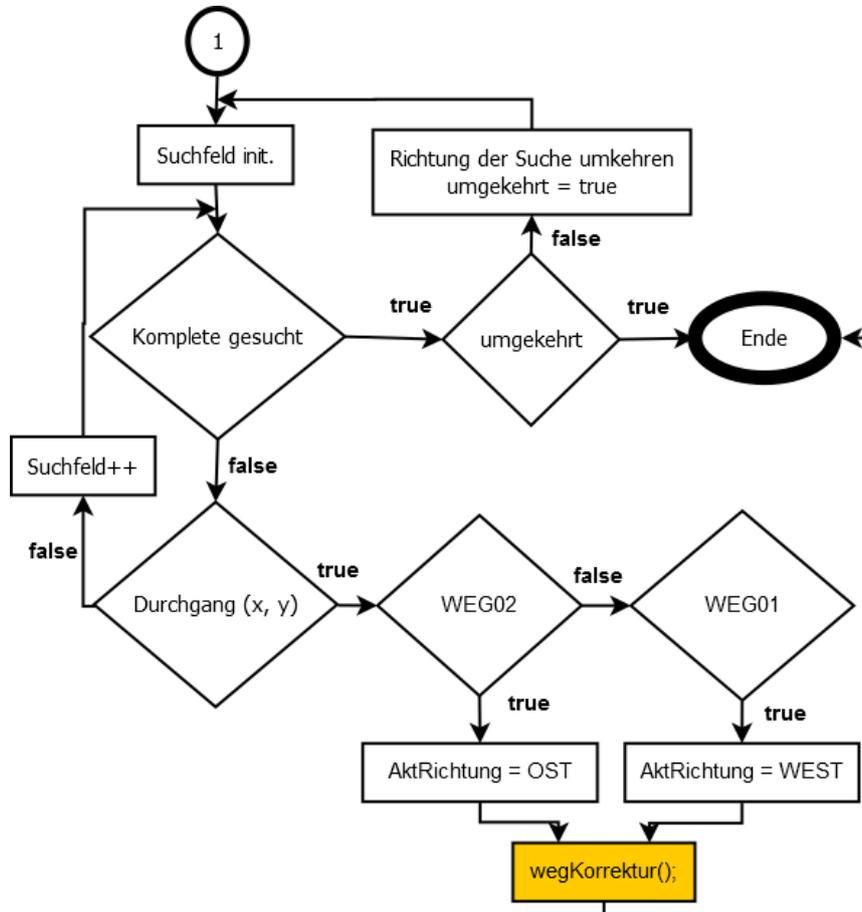
Der Algorithmus wird in zwei aneinander gekoppelte Bereiche unterteilt. Im Ablaufdiagramm [4.5](#) wird der erste Bereich dargestellt. Zuerst wird überprüft, ob die zu korrigierenden Positionen bzw. Trajektorie auf den für die Korrektur vorgesehenen Bereichen liegen. Nach der Bestätigung wird die Suchfeldrichtung für den zweiten Bereich des Algorithmus festgelegt. Weiterhin wird geprüft, ob die zu korrigierende Position direkt an einem Durchgang liegt, wodurch ein Raumwechsel möglich wäre. Wenn dies der Fall ist, wird, je nachdem wie die letzte bekannte Position ist, die Richtung ausgewählt. Schließlich wird entsprechend der Trajektorie korrigiert. Bei der „WegKorrektur“ Methode werden die Schritte ab dem Zeitpunkt, wo der Algorithmus gestartet wird, bis zum letzten erkannten Segment der Trajektorie berechnet und ermittelt.



Ablaufdiagramm 4.5: Kartenanpassungsmechanismus (Teil 01)

Falls der Überprüfung „Durchgang“ fehlschlägt, wird im zweiten Bereich des Algorithmus nach dem nächstmöglichen Bereich, indem ein Raumwechsel möglich ist, gesucht und entsprechend korrigiert. Hierbei wird zuerst das Suchfeld anhand der Informationen der Bewegungsrichtung, die schon im Vorfeld im ersten Bereich des Algorithmus gespeichert wurde, definiert. Fehler bei dem Schritterkennungsalgorithmus können sein, dass zu viele bzw. zu weniger Schritte erkannt werden. Da die Wahrscheinlichkeit fehlender Schritte weitaus höher ist, wird der Algorithmus zunächst in Laufrichtung mögliche Durchgänge bzw.

Türen absuchen. Andernfalls wird er in entgegengesetzter Laufrichtung suchen. Nach Festlegung des Suchfeldbereiches wird in kleinen Abständen nach dem nächstmöglichen Durchgang in der vorgemerkten Richtung gesucht. Nach finden des Durchgangs wird entsprechend die aktuelle Richtung festgelegt und die Trajektorie wird ermittelt. In Abbildung 4.6 wird der zweite Bereich des Algorithmus dargestellt.



Ablaufdiagramm 4.6: Kartenanpassungsmechanismus (Teil 02)

Hierbei wird die gesamte Anpassung der Positionen auf der Karte, bei einer falschen Erkennung der Richtung bzw. Schritte, zu einer Verbesserung des Lokalisierungssystems führen. Bei einer Fehlerkennung der Richtung wird eine Korrektur vorgenommen, indem die neue Richtung festgelegt und die gegangenen Schritte nachgebildet werden (erster Bereich des Algorithmus). Würde eine Person z.B. am Ende des Korridors nach links abbiegen und die Änderung der Richtung würde nicht erkannt werden, dann würde durch den Karteneinpassungsalgorithmus die Richtung und gegebenenfalls auch die Trajektorie korrigiert werden.

Bei einer falschen Erkennung der Schrittzahl im Falle eines Raumwechsels, würde zuerst nach den nächstmöglichen Durchgang gesucht und entsprechend korrigiert werden. Würde eine Person den Raum bei einem nicht definierten Durchgang wechseln, was möglicherweise durch eine Fehlerkennung des Schrittes passieren würde, würde die Position und die Trajektorie neu berechnet und auf der Karte angepasst werden (zweiter Bereich des Algorithmus).

5 Systemarchitektur und Implementierung

In diesem Kapitel wird die Hard- und Software des entwickelten Lokalisierungssystems näher erläutert. Dazu wird die verwendete Hardwarekomponente, das Sensor Kit vorgestellt. Weiterhin wird das verwendete Android Gerät veranschaulicht. Die entwickelte Applikation wird anhand von Komponenten- und Klassendiagrammen näher beschrieben. Anschließend wird die Entwicklungsumgebung des Lokalisierungssystems vorgestellt.

5.1 Hardware Komponente

Die verwendeten Sensoren für das entwickelte Lokalisierungssystem bestehen aus einem Sensor Kit und einem Android basierten Tablet. Die oben genannte Hardware wird im Folgenden näher im Detail beschrieben.

5.1.1 SensorTag

Die grundlegende Erklärung des SensorTag wurde im Kapitel 2.3.4 vorgestellt. Die detaillierte Beschreibung der verwendeten Sensoren des SensorTag wird im Folgenden näher erklärt. Die Architektur des SensorTag wird in der Abbildung [5.1](#) mit Hilfe eines Blockdiagramms dargestellt.

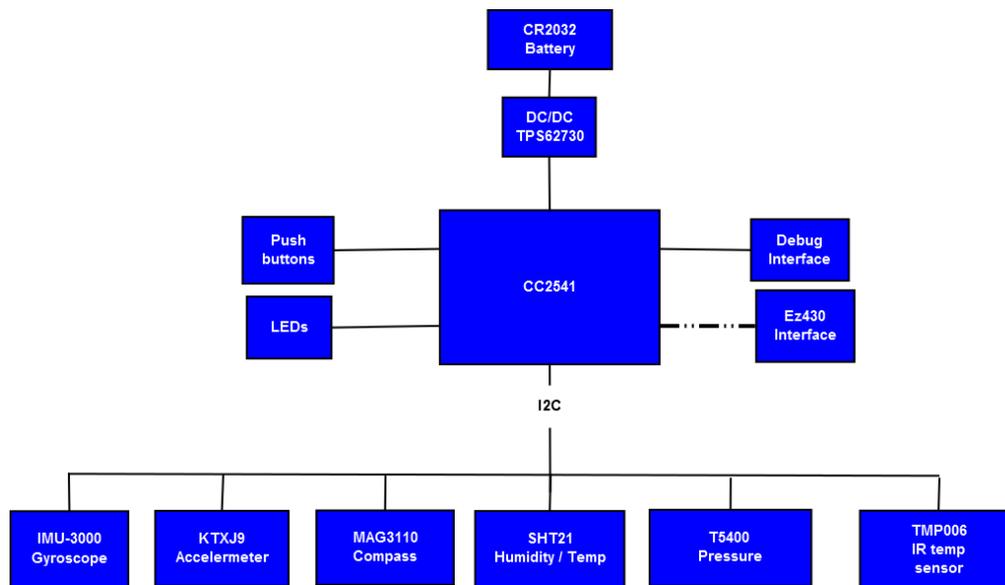


Abbildung 5.1: Blockdiagramm des SensorTag (vgl. Texas Instruments, block d.)

Die oben gelisteten Sensoren verwenden die I2C-Schnittstelle. Um den Stromverbrauch zu minimieren, sind alle Sensoren standardmäßig deaktiviert. Daher ist eine gezielte Aktivierung der gewünschten Sensoren notwendig. Im Folgenden werden die verwendeten Sensoren und Komponenten näher erläutert.

LEDs

Die LED Leuchten D1 und D2 zeigen den Status des SensorTag an. Bei einem Neustart signalisiert die blinkende D1-LED ihre Koppelungsbereitschaft für einen Verbindungsaufbau. Konnte dieser nicht korrekt ausgeführt werden, wird dies durch das Blinken der D2-LED gekennzeichnet.

Beschleunigungssensor (KXTJ9)

Der Beschleunigungssensor misst die Beschleunigung in drei Achsen mit programmierbarer Auflösung mit bis zu 14 Bit. Im Gegensatz zu einem Gyroskop, kann dieser auch die Richtung der Schwerkraft bestimmen. Der Beschleunigungssensor verfügt über einen internen Spannungsregler und einen I²C-Bus für die digitale Kommunikation. Der Sensor, der eine Betriebstemperaturbereich von -40 ° C bis + 85 ° C hat, ist in einem 3 x 3 x 0,9 mm LGA-Gehäuse untergebracht. Eine besondere Eigenschaft des KXTJ9 ist der niedrige Stromverbrauch. Dieser liegt im Standbybetrieb sowie bei niedriger Auflösung 10 µA und bei hoher Auflösung 325 µA. Außerdem kann benutzerdefiniert zwischen ± 2g, ± 4g oder ± 8g gewählt werden. Die Abbildung [5.2](#) stellt die Pin Anordnung und ein paar verfügbare Eigenschaften des Beschleunigungssensors (KXTJ9) dar.

SensorTag attribute table					
TI Base UUID: F000XXXX-0451-4000-B000-000000000000. 128-bit UUIDs are typed 'bold blue'					
Handle (hex)	Handle (dec)	Type (hex)	Type (text)	GATT Server Permissions	Description/Value (text)
0x2E	46	0x2800	GATT Primary Service Declaration	R	Accelerometer Service
0x2F	47	0x2803	GATT Characteristic Declaration	R	Accelerometer Data
0x30	48	0xAA11	Accelerometer Data	RN	X : Y : Z Coordinates
0x31	49	0x2902	Client Characteristic	RW	Write "01:00" to enable notifications, "00:00" to disable
			Configuration		
0x32	50	0x2901	Characteristic User Description	R	Accel. Data
0x33	51	0x2803	GATT Characteristic Declaration	R	Accelerometer Config
0x34	52	0xAA12	Accelerometer Config	RW	Write "01" to select range 2G, "02" for 4G, "03" for 8G, "00" disable sensor
0x35	53	0x2901	Characteristic User Description	R	Accel. Conf.
0x36	54	0x2803	GATT Characteristic Declaration	R	Accelerometer Period
0x37	55	0xAA13	Accelerometer Period	RW	Period = [Input*10] ms, (lower limit 100ms), default 1000 ms
0x38	56	0x2901	Characteristic User Description	R	Accel. Period

Abbildung 5.2: SensorTag-Attributtabelle des Beschleunigungssensors (vgl. Texas Instruments, GATT S.)

Gyroskop (IMU-3000)

Das Gyroskop misst die Rotationsgeschwindigkeit in 3 Achsen (X, Y, Z) mit einer Genauigkeit von bis zu 16-Bit. Der programmierbare Bereich liegt in ± 250 , ± 500 , ± 1000 und ± 2000 °/s (dps). Sein Rauschverhalten beträgt 0,01 dps / $\sqrt{\text{Hz}}$ (Industriestandard). Andere Eigenschaften des IMU-3000 sind programmierbare digitale Filter, eine Präzision mit ca. 1% Abweichung in einem Temperaturbereich zwischen -40 ° C bis 85 ° C und einem niedrigen Stromverbrauch, der im Bereich von 12 mW liegt. Seine Betriebsspannungsversorgung liegt zwischen 2,1 V bis 3,6 V. In Abbildung 5.3 wird die Pin-Anordnung und einige verfügbare Eigenschaften des Gyroskops (IMU-3000) dargestellt.

SensorTag attribute table					
TI Base UUID: F000XXXX-0451-4000-B000-000000000000. 128-bit UUIDs are typed 'bold blue'					
Handle (hex)	Handle (dec)	Type (hex)	Type (text)	GATT Server Permissions	Description/Value (text)
0x5E	94	0x2800	GATT Primary Service Declaration	R	Gyroscope Service
0x5F	95	0x2803	GATT Characteristic Declaration	R	Gyroscope Data
0x60	96	0xAA51	Gyroscope Data	RN	XLSB:XMSB:YLSB:YMSB: ZLSB:ZMSB
0x61	97	0x2902	Client Characteristic Configuration	RW	Write "01:00" to enable notifications, "00:00" to disable
0x62	98	0x2901	Characteristic User Description	R	Gyro Data
0x63	99	0x2803	GATT Characteristic Declaration	R	Gyroscope Config
0x64	100	0xAA52	Gyroscope Config	RW	Write 0 to turn off gyroscope, 1 to enable X axis only, 2 to enable Y axis only, 3 = X and Y, 4 = Z only, 5 = X and Z, 6 = Y and Z, 7 = X, Y and Z
0x65	101	0x2901	Characteristic User Description	R	Gyro Conf.
0x66	102	0x2803	GATT Characteristic Declaration	R	Gyroscope Period
0x67	103	0xAA53	Gyroscope Period	RW	Period = [Input*10]ms (lower limit 100ms), default 1000ms
0x68	104	0x2901	Characteristic User Description	R	Gyro Period

Abbildung 5.3: SensorTag-Attributtabelle des Gyroskops (vgl. Texas Instruments, GATT S.)

Die IMU-3000 ist im SensorTag Horizontal eingebaut, so dass die y-Achse nach oben, die z-Achse nach vorne und die x-Achse seitlich liegen. In Abbildung 5.4 werden die Achsenrichtungen des Gyroskops dargestellt.

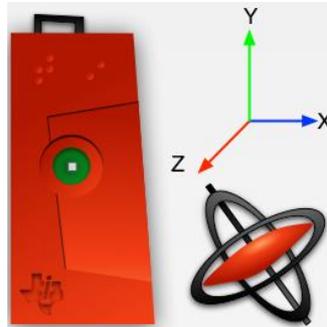


Abbildung 5.4: Achsenrichtungen des Gyroskops
(Texas Instruments, Gyroscope, 2013)

Magnetometer (MAG3110)

Das MAG3110 ist ein kleines und energiesparendes Magnetometer, welches die Magnetfelder in drei Hauptachsen misst, was der Eigenschaft eines Vector-Magnetometers (Kapitel 2.3.3) entspricht. Wobei die genauen Positionsinformationen in Verbindung mit einem Beschleunigungsmesser realisiert werden können.

Das MAG3110 verfügt über eine serielle I²C-Standardschnittstelle und intelligente eingebettete Funktionen, eine Datenrate mit bis zu 80Hz und kann in einem Temperaturbereich von -40 ° C bis + 85 ° C betrieben werden (vgl. Freescale, MAG3110). Weitere Eigenschaften des Sensors sind:

- 2 mm x 2 mm x 0,85 mm, 0,4 mm Pitch, 10-Pin-Gehäuse
- Vollskalenbereich $\pm 1000 \mu\text{T}$
- Empfindlichkeit von $0,10 \mu\text{T}$
- Rauschen bis zu $0,25 \mu\text{T rms}$ (Abkürzung für Root Mean Square, Quadratisches Mittel)

Der seitliche Knopf

Der Knopf dient zum An- und Ausschalten, sowie zum Zurücksetzen des SensorTag. Wird der Knopf während einer aktiven Verbindung gedrückt gehalten, wird die Verbindung getrennt und die Sensoren werden auf Werkseinstellung zurückgesetzt. Hierbei werden alle Datenwerte auf den Reset-Zustand zurückgesetzt. In Abbildung 5.5 wird der SensorTag mit und ohne Ummantelung dargestellt, hierbei zeigen die blauen Kreise dem seitlichen Knopf.



Abbildung 5.5: Seitlicher Knopf des SensorTag

5.1.2 Nexus

Für die Implementierung und Inbetriebnahme des entwickelten Lokalisierungssystems wird ein Android-Tablet „ASUS Nexus 7“ verwendet. Dieser wird mit einem QUALCOMM Snapdragon S4 Pro 8064 (1,5GHz - Quad Core) Prozessor betrieben und verfügt über 2 GB Arbeitsspeicher. Der entscheidende Punkt für die Auswahl des Nexus 7 ist, dass es mit einem Betriebssystem höher als Android 4.3 betrieben wird und Bluetooth 4.0 unterstützt, welche Systemvoraussetzungen zur Nutzung des SensorTags sind.

5.2 Software Architektur

Die Android Applikation dient zum Empfangen der Sensordaten, welche dann von den Lokalisierungsalgorithmen ausgewertet werden. Die Applikation wird „Indoor“ genannt. Nachfolgend wird die Applikation, die das Lokalisierungssystem umfasst, anhand von Komponenten- und Klassendiagrammen detailliert beschrieben.

5.2.1 Komponenten

In diesem Kapitel werden die Komponenten des Lokalisierungssystems näher beschrieben. Die Hauptkomponente *Map*, empfängt die Sensordaten vom SensorTag und stellt diese den anderen Komponenten zur Verfügung. Des Weiteren besteht *Map* aus vier inneren Komponenten. In einer dieser Komponenten wird die Anfangsposition mit Hilfe der GUI festgelegt. Die nächste Komponente berechnet die aktuelle Position anhand der Schrittzahl, Schrittlänge, Schrittrichtung und Anfangsposition. Eine weitere innere Komponente stellt die Schrittlängenberechnung bereit. Die Komponente Receiver ist für das empfangen der Sensorwerte verantwortlich, von diesen werden dann die wichtigen Werte extrahiert und den Komponenten bereitgestellt. Das Lokalisierungssystem benötigt die Anfangsposition und die Laufrichtung, diese werden durch eine andere Komponente bereitgestellt. Das Aktualisieren der GUI übernimmt eine andere Komponente, hierbei wird die Trajektorie mit der aktuellen Position der Person auf der Oberfläche aktualisiert. Die Komponente *Schritts* ist für die Auswertung des Beschleunigungswertes zuständig, womit sie die Muster eines Schrittes und dadurch die Schritte erkennt. In der Komponente *RichtungErkennung* wird die aktuelle Richtung durch Auswertung des Gyroskops erkannt. In der Komponente *KartenAnpassung* wird der vorgestellte Map Matching Algorithmus bereitgestellt, hierbei werden die Positionen auf der Karte angepasst. Die Segmente der Trajektorie, die aus Positionsanfang bis zur aktuellen Position bestehen, werden in der Komponente *Figur* als Trajektorie dargestellt. In der Abbildung [5.6](#) wird das Komponentendiagramm des Lokalisierungssystems dargestellt.

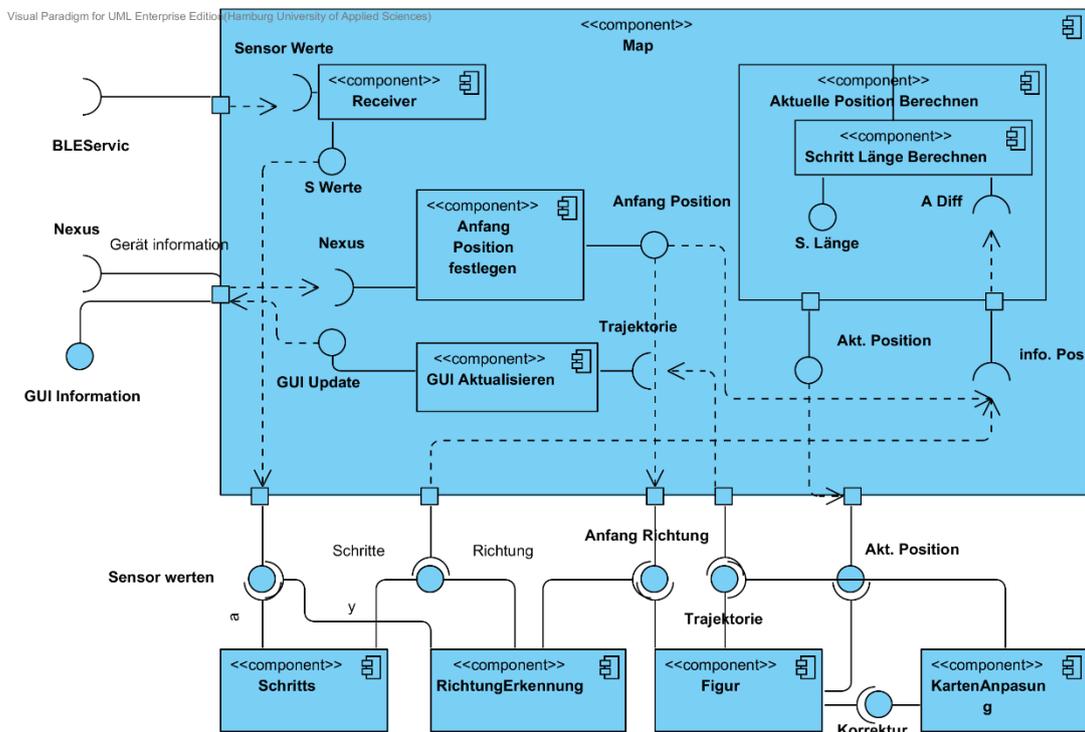


Abbildung 5.6: Komponentendiagramm des Lokalisierungssystems

5.2.2 Klassen

Die Applikation besteht aus zwei Bereichen. Im ersten Bereich wird die Verbindung zum SensorTag aufgebaut. Nach dem Verbindungsaufbau wird auch die Möglichkeit geboten die Sensoren des SensorTag einzustellen. Nach einem erfolgreichen Verbindungsaufbau werden die Beschleunigungs- und Gyroskop-Daten jeweils als Graph dargestellt. Dazu werden die Werte der Beschleunigung, des Gyroskops, des Magnetometers und des Drucksensors angezeigt. Zum Zwecke der Datenanalyse werden die Daten des Gyroskops, des Beschleunigungssensors und des Magnetometer jeweils in eine Textdatei aufgezeichnet und in die Speicherkarte des Geräts gespeichert.

Die Abbildung [5.7](#) stellt das Klassendiagramm des ersten Bereiches dar. Die *MainActivity* bildet die Hauptklasse, sie sorgt für eine reibungslose Zusammenarbeit aller anderen Klassen des ersten Bereiches. In ihr werden auch die Oberflächen bzw. GUI verwaltet. Weiterhin übernimmt sie auch das Einstellungsmenü der Applikation, wodurch die Änderung der Einstellung der einzelnen Sensoren mit Hilfe der Klasse *BLEService* ermöglicht wird.

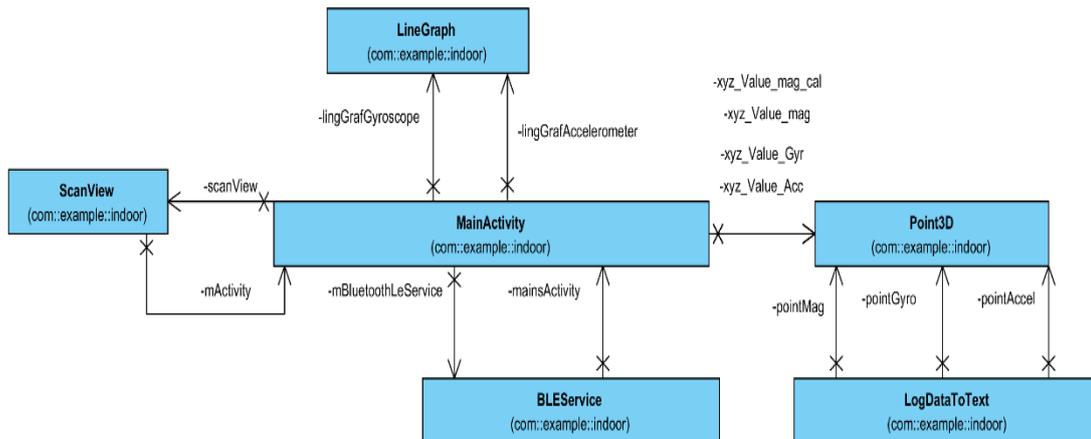


Abbildung 5.7: Erster Bereich der Applikation

BLEService ist ein Service für den Verbindungsaufbau und zur Kommunikation zwischen dem Nexus und dem SensorTag. Hierfür werden während der Verbindung die Sensorwerte empfangen und per Broadcast an die Applikation gesendet. In *Point3D* werden die Werte der Sensoren für die Ausgabe auf das Display und der Textdatei vorbereitet, außerdem werden die Beschleunigungen aus den erhaltenen Werten des Beschleunigungssensors berechnet. Die *LogDataToText* erzeugt Textdateien, in denen die erhaltenen Sensorwerte gespeichert werden. *ScanView* übernimmt die Verwaltung des Scan-Prozesses. Darunter fallen die Suche nach SensorTag-Geräten in der Nähe und der Darstellung der gefundenen Geräte auf das entsprechende View. In *LineGraph* werden die Liniendiagramme aufgebaut. Weiterhin werden die erhaltenen Werte des Beschleunigungssensors und des Gyroskops jeweils in eine grafische Oberfläche als Liniendiagramm dargestellt.

Die Klassen im zweiten Bereich sind für das Lokalisierungssystem zuständig. Hier sind alle vorgestellten Algorithmen aus dem vierten Kapitel implementiert. In der Abbildung 5.8 wird das Klassendiagramm des zweiten Bereiches dargestellt. Die Klasse *Map* ist der Hauptteil des zweiten Bereiches, hier werden die Oberflächen bzw. GUI verwaltet. Die Daten werden mit einem Broadcast Receiver empfangen und entsprechend in die anderen Klassen weitergereicht. Ein Beispiel wäre, dass Gyroskopdaten an die *RichtungErkennung* Klasse weiter gegeben werden. Die Schrittlängenberechnung (siehe Kapitel 4.3) befindet sich auch in der Klasse *Map*. Da die Beschleunigungswerte für die Berechnung der Schrittlänge benötigt werden, ist eine Speicherung der Werte vor der Filterung erforderlich. Dafür sorgt die Klasse *ZweiPoint*, sie sichert die gefilterten und gemessenen Beschleunigungswerte.

Schritts beinhaltet den Schritterkennung-Mechanismus bzw. -Algorithmus (siehe Kapitel 4.2), dazu werden die Beschleunigungswerte bearbeitet und gefundene Schritte der Klasse *Map* zugeteilt.

In der *KartenAnpassung* wird der vorgestellte Algorithmus für die Kartenanpassung betrieben, womit der vorgestellte Algorithmus (Kapitel 4.5) die Trajektorie korrigiert und *Map* mitteilt. Durch Nutzung des vorgestellten Algorithmus zur Richtungserkennung (siehe Kapitel 4.4) sorgt die Klasse *RichtungErkennen* für die Erkennung der Richtung. Die Klasse *Wegen* ist eine Enum-Klasse, sie enthält Information über die genutzte Karte, darunter auch die Deklaration der Flure und Eingänge. Sie beinhaltet auch zwei Prüfmethode, die den gewählten Punkt auf der Karte auf Gültigkeit prüft. Mit Hilfe der Klasse *FigurXY* werden die Koordinaten, sowie der Weg der aktuellen Position der Person gespeichert. Die Überprüfung jeder einzelnen Koordinate vor Einfügung findet hier statt. Die Klasse *WorkerSurfaceUpdate* aktualisiert die Oberfläche des Gerätes in einem vordefinierten Intervall.

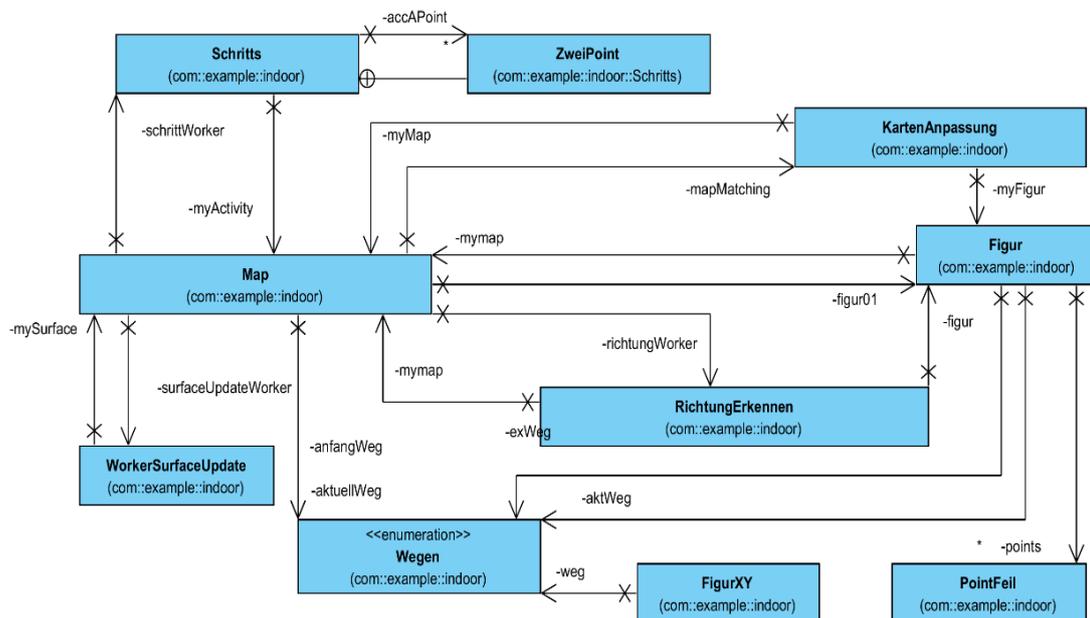


Abbildung 5.8: Zweiter Bereich der Applikation

5.2.3 Grafische Benutzeroberfläche (GUI)

In diesem Kapitel wird die grafische Benutzeroberfläche genauer beschrieben. Sie besteht hauptsächlich aus zwei Bereichen. In der ersten grafischen Oberfläche wird der Scanvorgang nach SensorTag-Geräten dargestellt. Hier gibt es die Möglichkeit das Intervall der einzelnen Sensoren einzustellen und den Protokollmodus ein- bzw. auszuschalten. Im anderen Teil werden die Karte und die Information für die Lokalisierung dargestellt.

Das Application-Icon bzw. -Symbol besteht aus drei Teilen. Die Person kennzeichnet den Anwender. Das Haus steht für die Verwendung im Indoor-Bereich und das Bluetooth Symbol deutet auf die Kommunikationsart hin. In [Abbildung 5.9](#) wird das Applikation-Icon dargestellt.



Abbildung 5.9: Applikation Icon

Der erste Bereich der grafischen Oberfläche wird in den Abbildungen [5.10](#) und [5.11](#) dargestellt. Nach dem Starten der Applikation wird nach SensorTag-Geräten in der Nähe gesucht. Sobald diese gefunden wurden, werden diese wie in der (Abbildung [5.10](#)) angezeigt. Nach der Auswahl eines SensorTag wird die Verbindung zwischen dem Nexus und dem SensorTag aufgebaut. Die Sensorwerte werden in Textform und als Liniendiagramm dargestellt (Abbildung [5.11](#)). Durch den Save OFF bzw. Save On Button wird der Protokollmodus ein- bzw. ausgeschaltet, wodurch die Sammlung der Sensorwerte ermöglicht wird. Der Map Button ist für den Wechsel auf die zweite grafische Oberfläche zuständig.

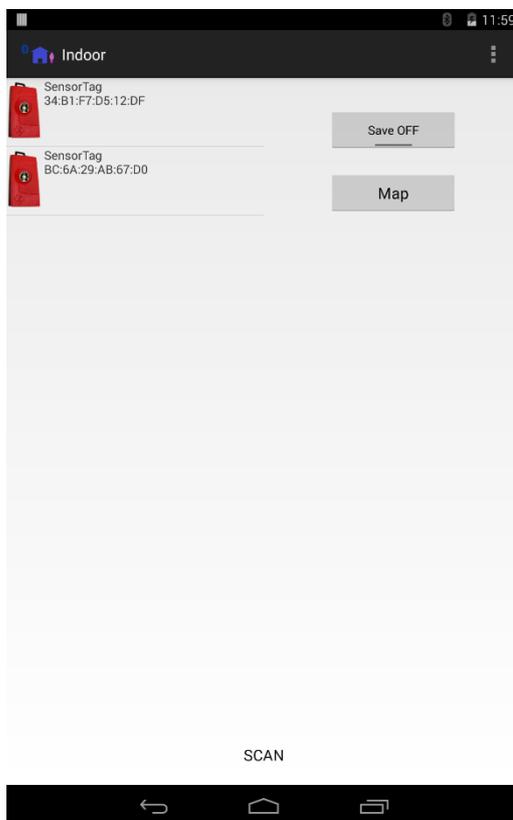


Abbildung 5.10: Erste Seite nach starten den Scanvorgang

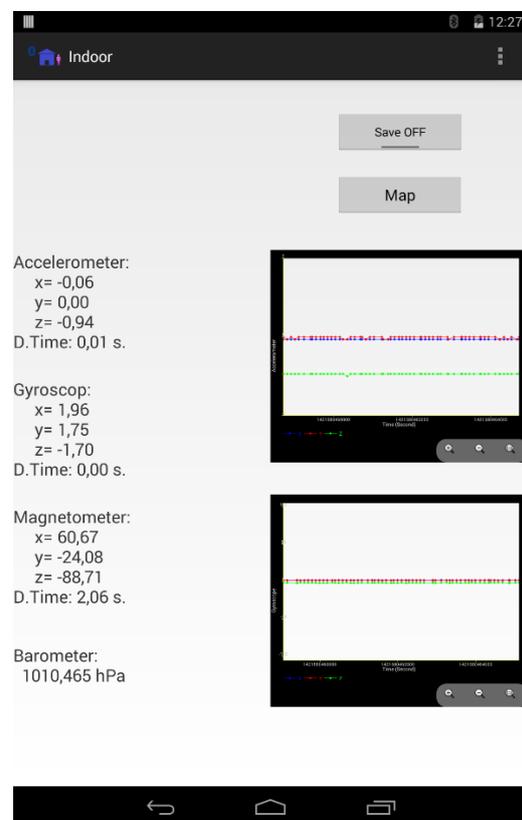


Abbildung 5.11: nach Verbindung mit einem SensorTag

Die Abbildungen [5.12](#) und [5.13](#) zeigen ebenfalls die erste Grafische Oberfläche, wobei in der Abbildung [5.12](#) die verschiedenen Sensorwerte dargestellt werden. In der Abbildung [5.13](#) werden die Einstellungsmöglichkeiten für die Intervalle der einzelnen Sensoren, das Kalibrieren des Magnetometer-Sensors (Mag. Kalb) und das An- bzw. Ausschalten des Liniendiagramms (Graph On/OFF) dargestellt.

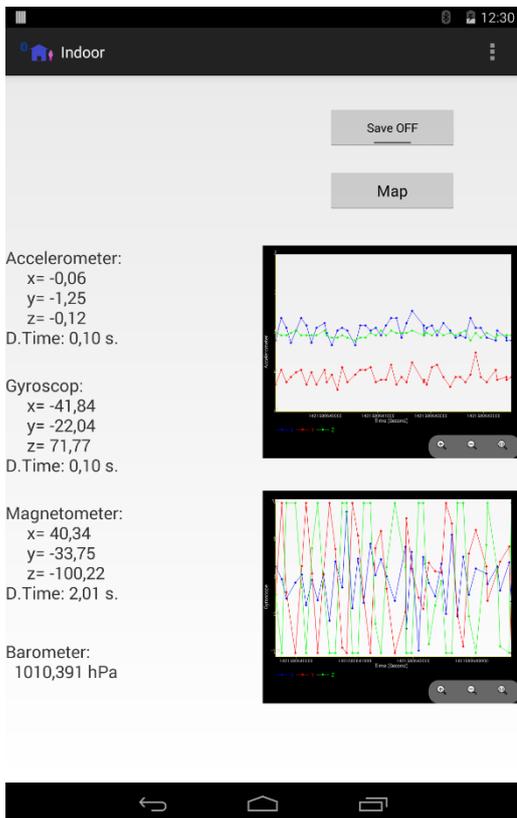


Abbildung 5.12: Erste Seite mit verschiedenen Sensorwerte

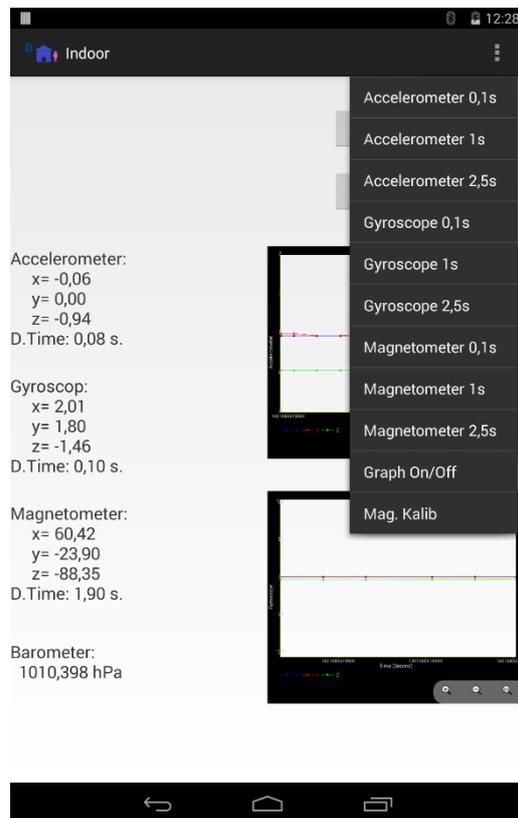


Abbildung 5.13: Erste Seite mit der Einstellungsübersicht

Die Abbildungen [5.14](#) und [5.15](#) zeigen die zweite grafische Oberfläche. Sie bildet die grafische Benutzeroberfläche für das Lokalisierungsprogramm, indem die Karte und die Information über die Position der Person dargestellt werden.

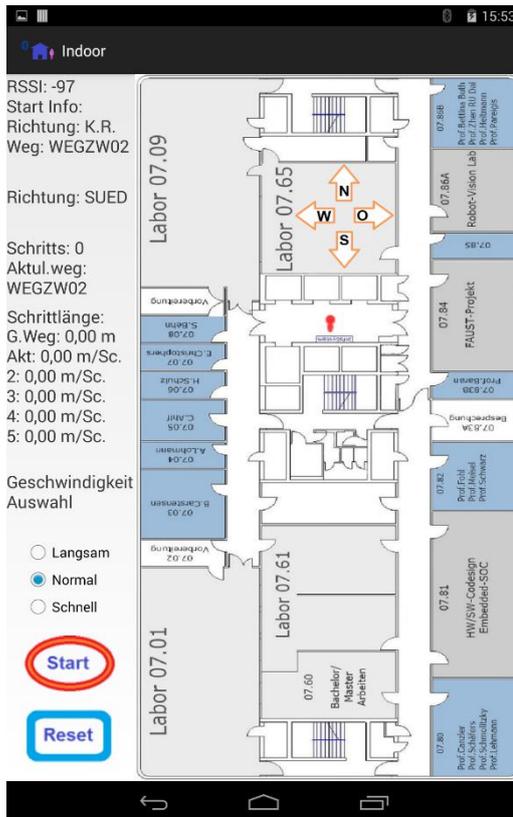


Abbildung 5.14: Zweite Seite vor Start

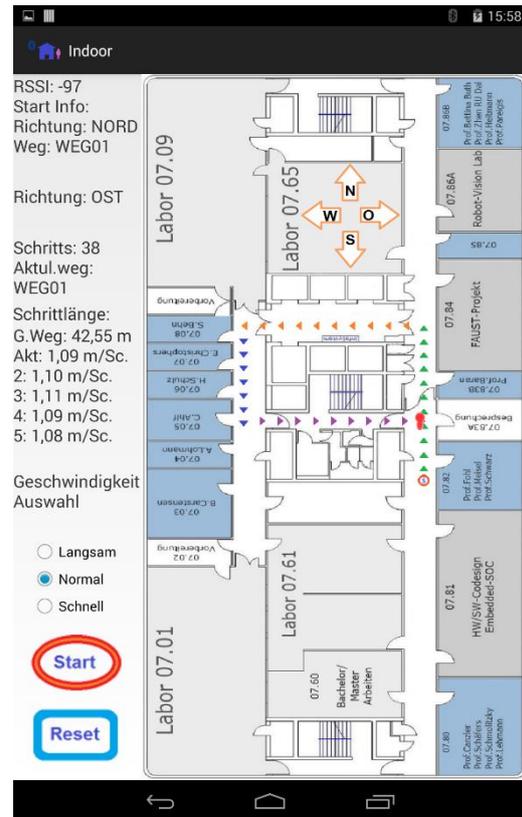


Abbildung 5.15: Zweite Seite mit Trajektorien

Hier ist die Einstellung der Anfangsrichtung, der Position sowie der Schrittgeschwindigkeit möglich. Außerdem werden die aktuelle Richtung, der Weg und die Trajektorie dargestellt. Das Einstellen der Richtung erfolgt durch die Berührung eines der vier orangenen Pfeile. Um die Anfangsposition festzulegen, wird durch das gedrückt halten des roten Symbols, das die zu lokalisierende Person darstellt, durch das Ziehen und Loslassen zur gewünschte Position auf der Karte ermöglicht. Nach der Festlegung der Anfangsposition und der Richtung wird durch Drücken des Start Buttons das Lokalisierungssystem gestartet. Hierbei wird die Anfangsposition auf der Karte mit einem roten Kreis, der den Buchstaben „S“ enthält, gekennzeichnet. Durch den „Reset“ Button wird das Programm auf den Anfangszustand zurückgesetzt. Weiterhin werden die Anfangsposition und die Richtung Links oben als Text angezeigt. Das zweite „Richtung“ Textfeld zeigt die aktuelle Richtung. Bei „Schritts“ wird die vergangene Schrittzahl angezeigt. Die „Aktul.weg“ zeigt den aktuellen Weg der Person.

Des Weiteren werden die Schrittlängeninformationen, bestehend aus sieben Zeilen Text, dargestellt. In der ersten Zeile ist der Titel „Schrittlänge“. Die zweite Zeile ist die gesamte Strecke in Meter „G.Weg“.

Die dritte Zeile ist die Länge des aktuellen Schrittes. Die vierte bis zur siebten Zeilen zeigen jeweils die Länge des vorherigen Schrittes. Zuletzt die Optionsfelder Langsam, Normal und Schnell. Sie sind für die manuelle Einstellung der Schrittgeschwindigkeit. Nach dem Start des Lokalisierungssystem wird anhand der Sensorwerte die Trajektorie dargestellt, sie wird durch einzelne Dreiecke symbolisiert. Sie kennzeichnen die Richtung des Weges. Hierbei sind für die Richtung Nord Grüne Dreiecke, für die Richtung West Orange Dreiecke, für die Richtung Süd Blaue Dreiecke und für die Richtung Ost lila farbige Dreiecke (Abbildung 5.15). „RSSI“ (engl. Received Signal Strength Indication) zeigt die Empfangsfeldstärke in dBm (Dezibel Milli Watt) des SensorTag zum Zeitpunkt der ersten Verbindung mit dem Nexus, an.

Allerdings ist auch zu erwähnen, dass bei dem Einstellen der Position begrenzte Bereiche der Karte Erlaubt sind. Dieser Bereich wird in der Abbildung 5.16 dargestellt, hierbei sind sie als rote Rechtecke mit grün und blau farbiger Füllung markiert. Die Vierecke mit blauer Füllung sind wichtig bei dem Kartenanpassungs-Algorithmus (siehe Kapitel 4.5), diese sind als Eingänge markiert.

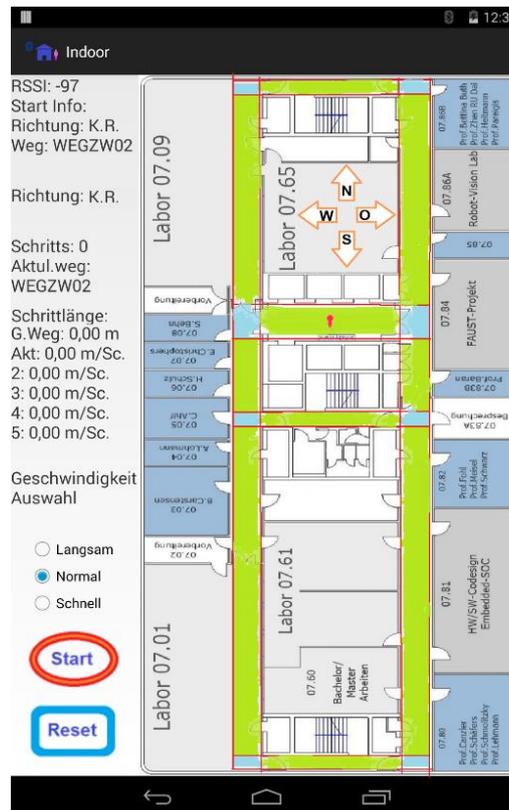


Abbildung 5.16: Definierte Bereiche auf der Karte

5.3 Entwicklungsumgebung

Die Entwicklungsumgebung (engl. integrated development environment kurz IDE) der Applikation wird in diesem Kapitel näher beschrieben. Hierbei werden die Anwendungsprogramme, die Entwicklung und die Bearbeitung der Applikation ermöglichen, vorgestellt.

5.3.1 Eclipse

Eclipse ist ein Programmierwerkzeug zur Entwicklung der Software in verschiedenen Programmiersprachen. Es wurde Ursprünglich für die Programmiersprache Java benutzt, aber wegen seiner Erweiterbarkeit wird es für viele andere Entwicklungsaufgaben eingesetzt, darunter auch die Android Applikation, die hauptsächlich mit Eclipse (IDE) entwickelt werden. Allerdings wird dafür der Plugin Android Development Tools (kurz ADT) benötigt, wodurch seine Fähigkeit für die Entwicklung der Android Applikationen erweitert wird. Android Studio ist die alternative zu Eclipse, die seit Mai 2013 von Google auf einer Konferenz angekündigt wurde. Es wurde kurz danach als Testversion zu Verfügung gestellt und schließlich am 8. Dezember 2014 als offizielle Android IDE veröffentlicht, daher fiel die Entscheidung auf Eclipse Version 4.2 (Juno).

5.3.2 Software Development Kit (SDK)

Eine Sammlung von Anwendungen und Werkzeugen wird ein Software Development Kit (kurz SDK) genannt. Es spielt auch eine besondere Rolle bei der Code Übersetzung bzw. Kompilierung für die entsprechenden Betriebssysteme. Daher wird zusätzlich zu dem ADT auch ein Android SDK benötigt. Da die Bluetooth Low Energy bei Android-Geräten ab Android Version 4.3 unterstützt wird, wird die aktuelle SDK API 21 (Android 5.0.1) verwendet. Allerdings unterstützt die Applikation auch Geräte die ab API 18 (Android 4.3) betrieben werden.

6 Evaluation

Im nachfolgenden Kapitel werden die Komponenten des entwickelten Lokalisierungssystems evaluiert. Weiterhin werden anhand Szenarien die Zusammenarbeit der Komponenten bzw. verwendete Lokisierungsalgorithmen unter dem Lokalisierungssystem getestet.

6.1 Testaufbau

In diesem Kapitel wird der Aufbau der Hardware für die Komponenten- und Systemtest erklärt. Zunächst wird der SensorTag unten am Bein befestigt, da beim Gehen sich der Fuß mehr bewegt als andere Körperteile. Dadurch können mehr Sensordaten zum Analysieren gesammelt werden. Der SensorTag wird vertikal angebracht, so dass die vordere Seite nach vorne zeigt (vgl. Abbildung [5.4](#)). Die Position des SensorTag spielt eine wichtige Rolle bei der Implementierung und Analyse der Sensordaten. Beim Richtungserkennungsalgorithmus werden die Daten der y-Achse für das Gyroskops benötigt. In Abbildung [6.1](#) wird die Position des SensorTag an einer Person dargestellt.



Abbildung 6.1: SensorTag Position am Bein der Testperson

Als Testumgebung wurde das siebte Obergeschoß der eigen besuchten Hochschule ausgewählt, wobei hier nur die zwei Korridore und vier zwischen Gänge verwendet wurden. In Abbildung 6.2 ist eine Karte zusehen, worauf der Testbereich markiert wurde. Die grünen Rechtecke sind die Testbereiche und diese wurden mit unterschiedlichen Namen bezeichnet (grünfarbige Beschriftung). Die vier orangefarbenen Pfeile sind die selbst festgelegten Himmelsrichtungen.

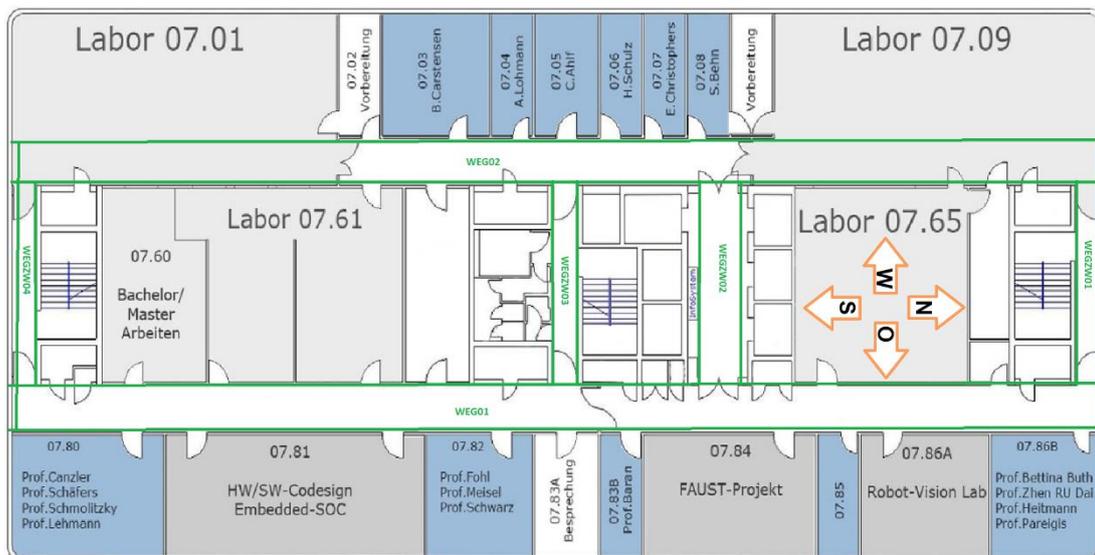


Abbildung 6.2: Testbereiche

Vor jeden Testanfang werden die Sensoren auf ein Intervall von 10 Hz eingestellt und eine Kalibrierung (Kapitel 2.7) durchgeführt, wodurch die Schwellwerte der Schritterkennung und des Richtungserkennungsalgorithmus festgelegt werden.

6.2 Komponententest

Die verwendeten Algorithmen für die Lokalisierung wurden als Komponenten des Lokalisierungssystems in dem Kapitel 5.2.1 vorgestellt. Sie werden in diesem Kapitel unabhängig voneinander auf Funktionalität und Genauigkeit getestet.

6.2.1 Schritterkennung

Zuerst werden 48 Schritte in dem vorgestellten Testaufbau gemacht. Der gegangene Weg wird am Anfang des WEG01 gestartet und geht bis zum Ende des Korridors. Hierbei werden 47 Schritte von 48 erkannt, was eine Genauigkeit von ca. 98% auf einer geraden Strecke zeigt. Hierbei liegt die Fehlerrate bei ca. 2%, was ein akzeptables Resultat darstellt, in Anbetracht

der Datenmenge welche durch das geringe Intervall von 10 Hz des SensorTag zur Auswertung und Erkennung eines Schrittes zur Verfügung stehen. In Abbildung 6.3 wird die Trajektorie dargestellt.

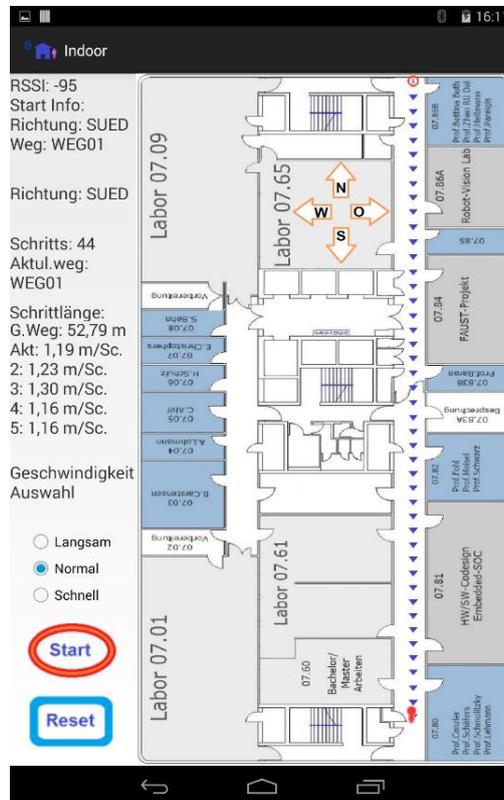


Abbildung 6.3: Test des Schritterkennungsalgorithmus

6.2.2 Schrittlänge

Der Schrittlängenalgorithmus wird in drei verschiedenen Geschwindigkeiten getestet: Langsames, normales und schnelles Gehen. Hierfür wird eine Strecke von 50 Meter gegangen. In der Tabelle 6.1 sind die Resultate dargestellt. Hierbei ergibt sich bei einer normalen Geschwindigkeit einer Fehlerrate von ca. 3,6%. Beim schnellen Gehen beträgt diese ca. 5,8% und beim langsamen Gehen liegt diese bei ca. 7%. Da die Beschleunigungswerte eine wichtige Rolle bei der Berechnung der Schrittlänge spielen (Formel 4.2), wird die Genauigkeit des Beschleunigungssensors eine direkte Auswirkung auf das Resultat der Schrittlänge haben. Daher sind die oben genannten Resultate mit der Nutzung des SensorTag akzeptable Ergebnisse.

Geschwindigkeit	Erkannte Schritte	Gemessene Werte in Meter	Fehlerrate
schnell	31	47,1 m	5,8 %
normal	36	48,2 m	3,6 %
langsam	44	53,5 m	7 %

Tabelle 6.1: Schrittlänge mit drei verschiedenen Geschwindigkeiten bei einer 50 m Strecke

6.2.3 Schrittrichtung

Der Schrittrichtungsalgorithmus ermittelt die Richtung nicht winkelgenau. Die Gründe dafür liegen bei der Nutzung eines nicht geeigneten Sensors und die zu lokalisierende Umgebung, da eine winkelgenaue Berechnung aufgrund der rechteckig gebauten Gebäude nicht notwendig ist. Der Algorithmus reagiert nur auf eine starke Drehung in ein definiertes Zeitfenster und legt entsprechend die Richtung fest. Deshalb wird die Richtungserkennung in zwei Drehrichtungen getestet (Kapitel 4.4). Hierfür wechselt die Testperson zehnmal seine Richtung, jeweils nur nach links und dann nur nach rechts. Dabei wird festgestellt, dass die Drehung nach links keine Fehlerrate zeigt, dagegen wird bei einer Drehung nach rechts eine Fehlerrate von ca. 1% festgestellt. Das liegt daran, dass der SensorTag am rechten Bein befestigt ist und bei einer Rechtsdrehung der Schwung sehr geringer ist als bei eine Linksdrehung. Durch das geringere Intervall (10 Hz) des Sensors ist die Fehlerrate von 1% akzeptabel.

6.2.4 Map Matching

Wie schon in Kapitel 4.5 erläutert, werden die zwei Bereiche des Algorithmus entsprechend der Situationen getestet.

Zuerst wird der erste Bereich des Algorithmus getestet. Hierfür wird die Person drei bis vier Schritte vor einem rechtwinkligen Korridorwechsel (WEG01 und WEG02 in der Abbildung [6.2](#)) platziert. Die Testperson startet daraufhin den Test, indem sie den Korridor wechselt. Hierbei ist darauf zu achten, dass bei der Richtungsänderung nur eine leichte Drehung erfolgt, so dass die Richtungsänderung nicht von dem Richtungserkennungsalgorithmus erfasst wird und somit ein Plausibilitätskonflikt verursacht wird, wodurch der Kartenanpassungsalgorithmus gestartet wird. Dabei wird erwartet, dass der Algorithmus die aktuelle Richtung erkennt und die Trajektorien an die Karte anpasst. In Abbildung [6.4](#) und [6.5](#) werden die Tests dargestellt. Die Tests lieferten erfolgreiche Resultate.

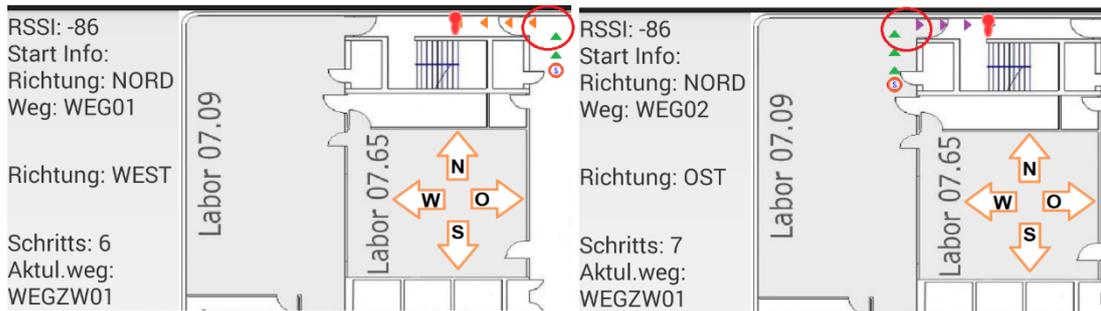


Abbildung 6.4: Türkorrektur in Richtung Nord

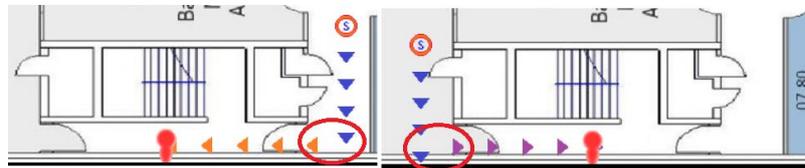


Abbildung 6.5: Türkorrektur in Richtung Süd

Weiterhin wird anhand zwei weiterer Szenarien der zweite Bereich des Algorithmus getestet, indem die Raumwechselkorrektur stattfindet. Das erste Szenario sieht vor, dass die Person ein oder zwei Schritte vor einem Eingang bzw. Durchgang schlürft, so dass sie ihren sensorbehafteten Bein beim Laufen nicht richtig anhebt, was zu einem nicht erkannten Schritt führt. Folglich wird ein Plausibilitätskonflikt verursacht, der den Kartenanpassungsalgorithmus startet. Erwartet wird nun, dass der nächstmögliche Eingang bzw. Durchgang für den Raumwechsel gefunden wird und entsprechend die Trajektorien angepasst werden. In [Abbildung 6.6](#) wird das erste Szenario mit vier erfolgreichen Resultaten dargestellt.

Im zweiten Szenario wird ein Plausibilitätskonflikt erzeugt, indem die Person eins bis zwei Schritte auf der Stelle geht. Dies hat zu Folge, dass bei Erreichen eines Durchgangs bzw. Eingangs der Schritterkennungsalgorithmus bereits ein bis zwei Schritte weiter ist. Bei einem jetzigen Raumwechsel wird nun der Kartenanpassungsalgorithmus gestartet. Zu erwarten ist, dass der nächstmögliche Eingang bzw. Durchgang für den Raumwechsel gefunden wird und entsprechend die Trajektorien angepasst werden. In [Abbildung 6.7](#) wird das zweite Szenario mit einem erfolgreichen Resultat dargestellt.

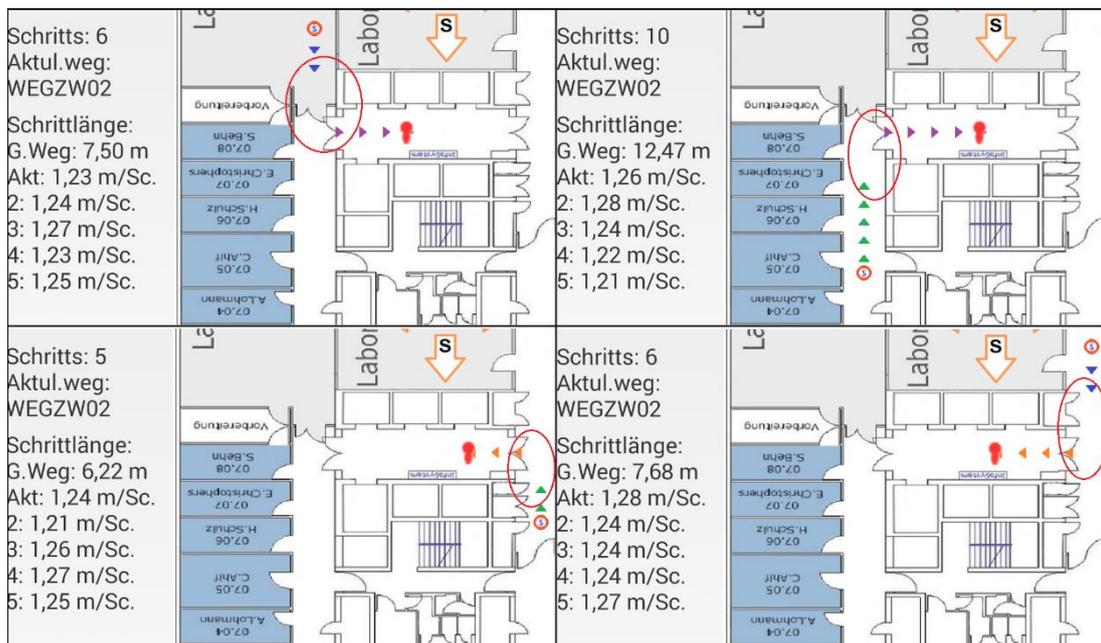


Abbildung 6.7: Raumwechselkorrektur bei der Fehlererkennung des Schrittes

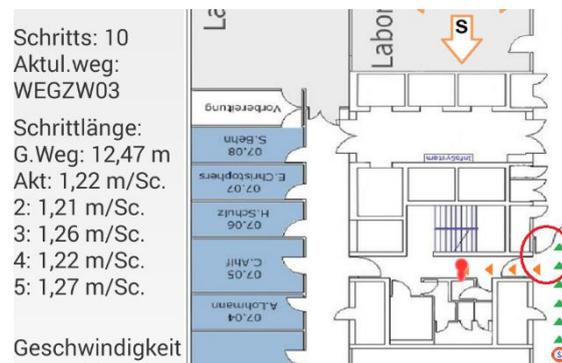


Abbildung 6.6: Raumwechselkorrektur bei der Falscherkennung des Schrittes

6.3 Systemtest

Das Gesamtsystem wird durch zwei Szenarien überprüft. In dem ersten Szenario wird die Zusammenarbeit der Komponenten überprüft. Bei dem zweiten Szenario wird das Lokalisierungssystem auf die Funktionalität und die Genauigkeit überprüft.

6.3.1 Erstes Szenario

In diesem Szenario wird versucht, die Zusammenarbeit der Komponenten zu testen. Hierbei geht die Testperson eine gerade Strecke und wechselt den Korridor, dabei werden die Zusammenarbeit der Algorithmen für Schritterkennung, Schrittrichtung und Kartenanpassung getestet.

Abbildung 6.8 stellt die Trajektorien der Testperson dar. Die Testperson startet vor dem Raum 07.60 und durch den Gang WEGZW03 wechselt sie den Korridor von WEG01 auf WEG02 und beendet ihre Route gegenüber Raum 7.61. Die Startpunkt und Endpunkt sind jeweils mit ein Kreis markiert, wobei der schwarze Kreis der Endpunkt und der orange farbige Kreis den Startpunkt darstellen.

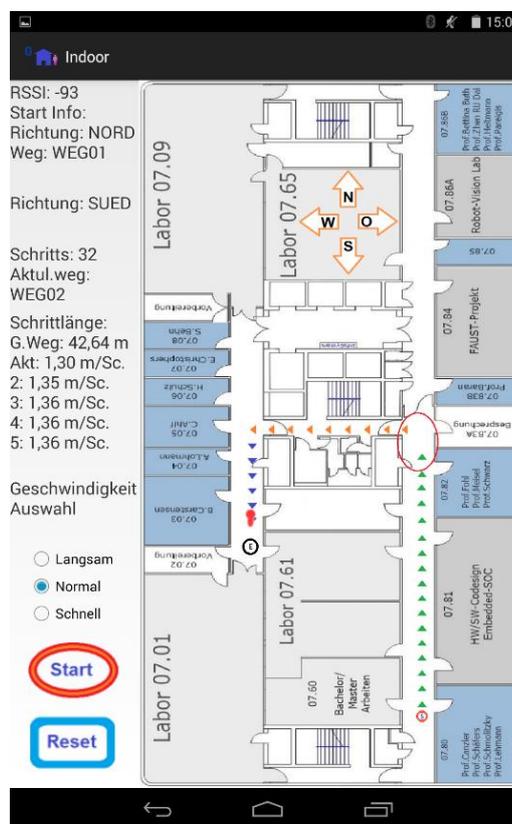


Abbildung 6.8: Erstes Szenario

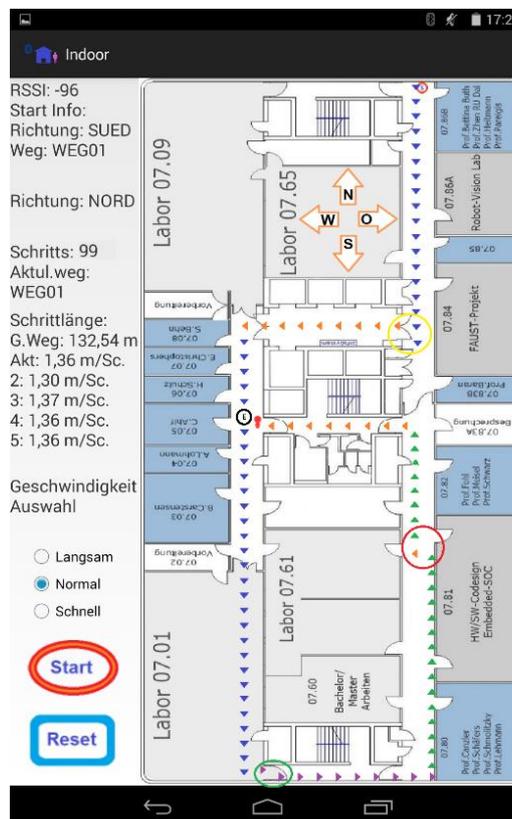
Durch das Szenario wird festgestellt, dass die Komponenten gut miteinander arbeiten. Hierdurch werden 32 von 33 gegangene Schritte erkannt, daher ist die Zielposition (schwarzer Kreis) nicht erreicht. Durch den Kartenanpassungsalgorithmus wird ein unerlaubter Raumwechsel verhindert. Dies wird durch der rote Kreis in Abbildung 6.8 dargestellt. Dieser zeigt auch den nicht erkannten, fehlenden Schritt.

Die gegangene Strecke ist ca. 44m lang. Bei diesem Test wird eine Strecke von ca. 42,6m, welche eine Differenz von 1,3m entspricht, gemessen. Das Experiment hat sowohl die Zusammenarbeit der Komponenten bestätigt, als auch die Lokalisierung im Gebäude mit einer Ungenauigkeit von ca. 1m, was eine akzeptables Resultat für die Indoor Lokalisierung durch Inertialsensoren ist.

6.3.2 Zweites Szenario

Für die Auswertung des gesamten Systems werden zwei Tests durchgeführt, die alle dem gleichen Szenario entsprechen.

In diesem Szenario werden alle zuvor definierten Bereiche (Abbildung 6.2) mindesten einmal durch eine willkürliche Route getestet. Es wird keine bestimmte Route festgelegt, damit die Lokalisierung nach einer bestimmte Zeit getestet wird. Es wird hierfür eine Zeitlang durch die definierten Bereiche spaziert. In Abbildung 6.9 wird der erste Test dargestellt.



Beim ersten Test startet die Testperson vor dem Raum 07.86B und endet gegenüber dem Raum 07.05. Es werden 102 Schritte gemacht. Von 102 Schritten werden 99 Schritte erkannt. Die grünen und gelben Kreise zeigen die korrigierten Trajektoriensegmente durch den Kartenanpassungsalgorithmus an. Der grüne Kreis zeigt die Anpassung von einer Position, da der nächste zu laufende Schritt durch eine Wand gehen würde. Beim gelben Kreis wird ein Raumwechsel durch das „durch die Wand gehen“ verhindert, hierfür werden die Trajektoriensegmente auf die richtige Position verschoben. Eine manuelle Richtungskorrektur markiert der rote Kreis. Hierbei wird ein Schritt nach links gemacht, um die manuelle Änderung der Schrittrichtung zu testen (hier: Durch Drücken auf den Pfeil Richtung Nord). Das Endziel wurde erreicht, hierdurch hat das entwickelte Lokalisierungssystem sich in Bereichen Genauigkeit und Funktionalität als Gut bewiesen.

Im zweiten Test wird eine Strecke über 442m gegangen. Es werden 329 Schritte gemacht, dabei werden 320 Schritte gemessen. Die Bereiche, welche mit einem Kreis markiert sind, zeigen die Arbeit bzw. die Korrektur des Kartenanpassungsalgorithmus an. In [Abbildung 6.10](#) wird der zweite Test dargestellt. Die roten Kreise zeigen die normale Verschiebung bzw. Korrektur der Trajektoriensegmente beim Wechsel in einen erlaubten Durchgang. Die gelben Kreise zeigen die Verschiebung der Position von Ende der WEGZW01 nach WEGZW02 /

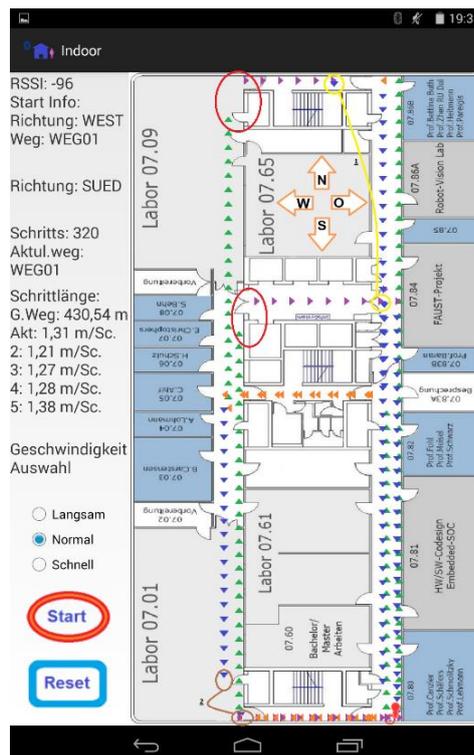


Abbildung 6.10: Zweiter Test des zweiten Szenarios

WEG01. Trotz nicht eingezeichneter Trajektoriestegmente auf der Karte, wird die Position richtig ermittelt. Der gleiche Sachverhalt wird mit dem braunen Kreisen dargestellt.

Der zweite Test hat auch das Endziel erreicht, obwohl falsche oder auch fehlende Trajektoriestegmente auf der Karte zu sehen sind. Trotzdem hat das Lokalisierungssystem sein Ziel, die Lokalisierung einer Person in einem Gebäude durch Inertialsensoren, erreicht.

7 Schluss

In diesem Kapitel wird eine abschließende Betrachtung der Bachelorarbeit erfolgen. Zunächst werden die erreichten Ziele zusammengefasst und danach eine Bewertung des entwickelten Lokalisierungssystems unterzogen. In einem Ausblick werden mögliche Weiterentwicklungen und Ideen thematisiert.

7.1 Zusammenfassung

Die vergleichbaren Verfahren (Kapitel 3) haben gezeigt, dass es eine grundsätzliche Unterscheidung bei der Auswertung der Messdaten in *Inertialnavigation* und *Koppelnavigation* gibt. In einem Verfahren wird durch Doppelintegration der Sensorwerte die eigene Position ermittelt, was allerdings einen hochwertigen Sensor erfordert. In einem anderen Verfahren wird hauptsächlich die Erkennung der Schritte durch Schrittmuster und Richtungserkennung getrieben, dabei spielen die benutzten Sensoren und deren Hochwertigkeit eine nicht so große Rolle. Weiterhin zeigen die Algorithmen für die *Ortungsverbesserung* ein erwähnenswertes Resultat. Sowohl bei dem Abgleich mit einem zugrunde liegenden Gebäudemodell, als auch der Filterung der Messdaten können unwahrscheinliche und offenbar fehlerhafte Ortungen korrigiert bzw. ausgeschlossen werden.

Im Verlauf der Analyse der verwendeten Lokisierungsalgorithmen (Kapitel 4) wurden die zur Lokalisierung entwickelten Algorithmen vorgestellt. Diese wurden in mehreren Bereichen untersucht. In *Algorithmen für die Lokalisierung* wurden die Algorithmen für die Lokalisierung durch Inertialsensoren erklärt, hierbei wurde für das Lokalisierungsgebiet die Koppelnavigation entschieden, wodurch die Zusammenarbeit der Schritterkennung, Schrittlänge und Schrittrichtung erforderten. Hierfür wurden die entwickelten Algorithmen anhand vorgestellten Algorithmen (Kapitel 3) und die Betrachtung der verwendeten Sensoren vorgestellt. Reicht man rein messwertbasierte Ortungssysteme mit zusätzlichen Informationen an, so wird durch andere Realisierungen eine verbesserte Korrektur erreicht. Aufgrund dessen wurde ein Filter für die Vorbearbeitung der Messdaten vorgestellt.

Weiterhin wurde mit einem zugrunde liegenden Gebäudemodell eine weitere Realisierung der Lokalisierungsergebnisse (Map-Matching) vorgestellt.

Die Realisierung und Implementierung der eigenen Entwicklung erfolgte in Kapitel 5, hierfür erforderte die Beschreibung sowohl der verwendeten Hardware, als auch die Architektur der entwickelten Software bzw. der Applikation. Einerseits wurde die verwendeten Sensoren, die von der Firma Texas Instruments unter einem Sensor-Kit (SensorTag) zusammengefügt wurden und das Android Gerät in Detail vorgestellt. Andererseits erforderten die selbst entwickelten Algorithmen eine eigene Applikation, damit sowohl das Empfangen und Auswertung der Daten der Inertialsensoren als auch die benötigten Informationen über das Gebäudemodell an einer zentralen Stelle zu vereinen und einen Zugriff zu ermöglichen. Weiterhin wurde die Architektur der selbst entwickelten Applikation „Indoor“ anhand Komponenten und Klassen vorgestellt. Hierzu wurden die grafischen Benutzeroberflächen mit entsprechenden Bildern beschrieben. Abschließend wurde für die Entwicklung der Applikation benötigten Entwicklungsumgebungen vorgestellt.

Die Genauigkeit und Leistungsfähigkeit des entwickelten Lokalisierungssystems wurde anhand der Evaluation (Kapitel 6) überprüft. Hierfür wurde zuerst der Aufbau der Tests beschrieben. Einerseits wurden die Komponenten des Lokalisierungssystems einzeln und unabhängig voneinander evaluiert, und deren Genauigkeit bewertet. Andererseits erfolgte ein Systemtest, wodurch des gesamtes Lokalisierungssystems auf Funktionalität und Genauigkeit überprüft wurde. Hierfür mussten repräsentative Szenarien entwickelt und mit unterschiedlichen Toleranzen durchlaufen werden.

7.2 Bewertung

Das Ziel der Bachelorarbeit war, die Entwicklung einer prototypischen Anwendung in Android Systemen zur Lokalisierung in Gebäuden. Die Lokalisierung erfolgt durch Auswertung der Inertialsensoren, welche sich durch Bluetooth Low Energy mit dem Android Gerät verbinden. Die entwickelte Applikation bietet vielerlei Funktionalitäten, die alle Anforderungen des Zieles beinhalten. Hierfür sind das Abtasten der Umgebung nach SensorTag sowie die Herstellung eine stabile Verbindung mit dem Sensor einige Funktionen der Applikation. Weiterhin sorgt die Applikation für die reibungslose Bearbeitung das Lokalisierungssystem und dessen Komponenten. Sie bietet sowohl die manuelle Einstellung des Intervalls der einzelnen Sensoren, als auch die Darstellung der aktuellen Werte der Sensoren als Text und Liniendiagramm. Sie bietet auch durch ihre Vielfältigkeit die Darstellung der Trajektorien und der aktuelle Position auf der Karte in Echtzeit. Es werden auch mehrere Information der zu lokalisierende Person in Echtzeit als Text dargestellt. Diese Applikation ist nicht Marktreif und dient dem eigenen Gebrauch und zu Testzwecken des Lokalisierungssystems. Daher gibt es auch ein paar kleine Fehler, die eigentlichen Funktionalitäten nicht beeinträchtigen. Wenn die Applikation gestartet und beendet wird, wird das Beenden der Applikation nicht sauber durchgeführt, so dass diese noch im Hintergrund läuft. Die Applikation kann dann mit Hilfe

des Android-Taskmanager beendet werden. Allerdings wurde dafür gesorgt, dass vor dem Beenden der Applikation die Verbindung mit dem Sensor getrennt wird.

Die entwickelten Lokalisierungsalgorithmen zeigten in Anbetracht des Intervalls der Sensoren ein akzeptables Resultat. Der Schritterkennungsalgorithmus zeigte eine Fehlerrate von ca. 2%. Bei der Richtungserkennung lag die Fehlerrate bei ca. 1%. Durch den ausgewählten Algorithmus wurde die Bewegungsfreiheit erheblich eingeschränkt. Weiterhin erwies sich die Schrittlängenberechnung als gut. Allerdings wurde die Schrittlängenberechnung abhängig von den erkannten Schritten entwickelt, was als Nachteil des Lokalisierungssystems betrachtet wird. Neben der Lokalisierungsalgorithmen erfolgt zusätzlich eine Plausibilitätsüberprüfung auf Basis eines einfachen Bewegungsmodells, wodurch ermittelte Positionen nur in definierten Bereichen zugelassen werden. Dies führt auch zur Verbesserung des Systems.

Der entwickelte Kartenanpassungsalgorithmus liefert zwar keine direkte Auswirkung innerhalb von Räumen, aber erst bei einer Bewegung durch mehrere aufeinanderfolgende Räume innerhalb eines Gebäudes zeigte dieser gute Resultate. Insbesondere auch dann, wenn die Fehlerkennung durch andere Lokalisierungsalgorithmen verursacht wurde. Zum Beispiel wurde ein unerlaubter Raumwechsel durch Fehlerkennung der Schritte zuerst erkannt und entsprechend korrigiert.

7.3 Ausblick

Bei dem Richtungserkennungsalgorithmus wurde nur die Umdrehung um ca. 90° wahrgenommen, wobei der genaue Winkel mit einem der vorgeschlagenen Algorithmen und einem besseren Sensor bestimmen kann.

Für die Nutzung der Information der Karte wurde die hierfür benutzte Karte manuell digitalisiert, wobei eine Digitalisierung der gesamten Karte sehr aufwendig ist. Daher wurde nur ein kleiner Bereich der Karte digitalisiert. Es existieren mehrere Softwares, mit denen eine Karte automatisch digitalisiert werden kann, dadurch kann die gesamte Karte bei dem Lokalisierungssystem verwendet werden. Weiterhin wurde nur ein Teilbereich des entwickelten Systems durch den Kartenanpassungsalgorithmus beobachtet und gegebenenfalls korrigiert. Dies kann erweitert werden, so dass das gesamte System beobachtet und ggf. korrigiert wird.

Die entwickelte Applikation stellt die Verbindung nur mit einem SensorTag her, hierfür könnte durch die Erweiterung der Applikation die Verbindungsanzahl auf zwei erhöhen, wodurch zwei verschiedene Sensoren mehrere Messungen zu Verfügung stellen. Die Sensoren können dann anstatt auf einen Bein an beiden Beinen eingesetzt werden.

Weiterhin soll für ein sauberes Beenden der Applikation gesorgt werden. Die Nutzung der Sensoren des SensorTag sind nur in einem begrenzten Intervall, welche durch seine Firmware

definiert ist, zugelassen. Die Firmware kann modifiziert werden, um das Intervall der einzelnen Sensoren zu erhöhen. Dadurch kann eine Verbesserung des Lokalisierungssystems ermöglichen.

Abbildungsverzeichnis

Die Struktur von Android (Android)	10
Drei Achsen von Beschleunigungssensor	12
Umdrehung um die Hauptachsen	13
Einen Laufverhalten des Beines	16
Schrittlänge	17
Eine Tür-Korrektur.....	18
Vereinfachter Strapdown-Algorithmus von Johannes Schmid (vgl. Schmid, Johannes) -----	20
Beschleunigungsdaten eines Schrittes-----	21
Die Schritterkennung Phasen (Wanner, Nico)-----	22
Horizontal- und Vertikal Beschleunigung-----	23
Ein nichtdeterministischer endlicher Automat (Wanner, Nico)-----	23
Erste und zweite Schwungphase (vgl. Kim, Jeong W., et al) -----	24
Flussdiagramm des Schritterkennungs-Algorithmus -----	25
Das Beschleunigungssignal von 60 cm und 80 cm Schritt (vgl. Kim, Jeong W., et al)-----	25
Der HDR-Algorithmus (Borenstein, Johann, Ojeda, Lauro and Kwanmuang, Surat) -----	28
Test des HDR-Verfahrens (Borenstein, Johann, Ojeda, Lauro and Kwanmuang, Surat) -----	29
Ablaufdiagramm des CADMS Frameworks (vgl. CADMS TEAM)-----	32
Durchgangskorrektur-CADMS (vgl. CADMS TEAM) -----	33
IMU Pfad vor und nach Map Matching (vgl. CADMS TEAM)-----	33
Partikelwolke für den nächsten Schritt (vgl. Meyer, Johannes) -----	34
Übersicht des Systems	36
Schrittmuster	38
Schrittmuster nach Filterung	39
Zustandbereiche	39
Drehung um einen rechten Winkel.....	43
Blockdiagramm des SensorTag (vgl. Texas Instruments, block d.)	50
SensorTag-Attributtabelle des Beschleunigungssensors (vgl. Texas Instruments, GATT S.) .	51
SensorTag-Attributtabelle des Gyroskops (vgl. Texas Instruments, GATT S.).....	52
Achsenrichtungen des Gyroskops (Texas Instruments, Gyroscope, 2013).....	53

Seitlicher Knopf des SensorTag	54
Komponentendiagramm des Lokalisierungssystems.....	56
Erster Bereich der Applikation	57
Zweiter Bereich der Applikation	58
Applikation Icon	59
Erste Seite nach starten den Scanvorgang.....	60
nach Verbindung mit einem SensorTag	60
Erste Seite mit verschiedenen Sensorwerte	61
Erste Seite mit der Einstellungsübersicht	61
Zweite Seite vor Start.....	62
Zweite Seite mit Trajektorien.....	62
Definierte Bereiche auf der Karte	63
SensorTag Position am Bein der Testperson.....	65
Testbereiche.....	66
Test des Schritterkennungsalgorithmus	67
Türkorrektur in Richtung Nord.....	69
Türkorrektur in Richtung Süd.....	69
Raumwechselkorrektur bei der Falscherkennung des Schrittes.....	70
Raumwechselkorrektur bei der Fehlerkennung des Schrittes.....	70
Erstes Szenario	71
Zweites Szenario – erster Test	72
Zweiter Test des zweiten Szenarios	73
SensorTag Position am Bein der Testperson.....	65
Testbereiche.....	66
Test des Schritterkennungsalgorithmus	67
Türkorrektur in Richtung Nord.....	69
Türkorrektur in Richtung Süd.....	69
Raumwechselkorrektur bei der Falscherkennung des Schrittes.....	70
Raumwechselkorrektur bei der Fehlerkennung des Schrittes.....	70
Erstes Szenario	71
Zweites Szenario – erster Test	72
Zweiter Test des zweiten Szenarios	73

Diagrammverzeichnis

Verwendete Filter	37
Schritterkennung.....	38
Verbesserter Schritterkennungsalgorithmus.....	41
Richtungserkennung Algorithmus.....	44
Kartenanpassungsmechanismus (Teil 01).....	46
Kartenanpassungsmechanismus (Teil 02).....	47

Tabellenverzeichnis

Der Mittelwert des absoluten Beschleunigungswertes (vgl. Kim, Jeong W., et al)	26
Der Zeitraum eines Schrittes (vgl. Kim, Jeong W., et al)	26
Schrittlänge mit drei verschiedenen Geschwindigkeiten bei einer 50 m Strecke	68

Literaturverzeichnis

- ABYARJOO, Fatemeh; BARRETO, Armando; COFINO, Jonathan; ORTEGA, Francisco R.: *Implementing a sensor fusion algorithm for 3D orientation detection with inertial/magnetic sensors: Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering*:Springer, 2015.
- Android: *Android software stack*. 2014,
<https://source.android.com/devices/tech/security/>:Android.
- ASCHER, C.; KESSLER, C.; WANKERL, M.; TROMMER, G. F.: *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, 2010.
- BEAUREGARD, Stéphane: *Positioning, Navigation and Communication, 2007. WPNC'07. 4th Workshop onIEEE*, 2007.
- BECKER, Arno: *Die Architektur von Android Innenansichten eines Smartphone-Betriebssystems*. 2014, <http://heise.de/-1203743>:heise online, 2011.
- Bluetooth: *A brief tutorial on Bluetooth wireless technology*. 2014,
<http://www.bluetooth.com/Pages/Fast-Facts.aspx>:Bluetooth.
- Bluetooth Smart: *The Low Energy Technology Behind Bluetooth Smart*. 2014,
<http://www.bluetooth.com/Pages/low-energy-tech-info.aspx>:Bluetooth.
- BORENSTEIN, Johann; OJEDA, Lauro; KWANMUANG, Surat: Heuristic reduction of gyro drift for personnel tracking systems. *In: Journal of Navigation*, (2009), Nr. 62, S. 01, 41-58.

- CADMS TEAM: *An Indoor Positioning System for Disaster Management*, http://portal.tugraz.at/portal/page/portal/Files/i2330/cadms/CADMS_description.pdf. Graz: Institute for Building Informatics Graz University of Technology, 2009.
- CHO, Seong Y.; PARK, Chan G.: MEMS based pedestrian navigation system. *In: Journal of Navigation*, (2006), Nr. 59, S. 01, 135-153.
- EPCOS, T5400: *Pressure Sensor*. 2015, <http://www.epcos.com/inf/57/ds/T5400.pdf>:EPCOS, 2013.
- FREESCALE, MAG3110: *MAG3110: Freescale High Accuracy, 3D Magnetometer*. 2015, http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MAG3110:Freescale.
- GRUBBS, Frank E.: Procedures for detecting outlying observations in samples. *In: Technometrics*, (1969), Nr. 11, S. 1, 1-21.
- GRUNAU, Sina; SPRINGER, Ing T.: KONZEPTION UND EVALUATION ERWEITERTER MECHANISMEN ZUR POSITIONSBESTIMMUNG IN GEBÄUDEN.
- GUBI, Kurt: Roughmaps: Indoor positioning using existing infrastructure and symbolic maps. *In: Bericht, CHAI Group, School of IT, The University of Sydney*, (2010).
- GUSENBAUER, Dominik; ISERT, Carsten; KROSCHE, J.: *Indoor positioning and indoor navigation (IPIN), 2010 international conference on IEEE*, 2010.
- IFFLÄNDER, Lukas; GAGEIK, Dipl.-Ing N.: *Entwicklung und Evaluierung eines Systems zur Bestimmung der Orientierung und Position eines Objektes durch inertielle und magnetische Sensoren*, http://www8.informatik.uni-wuerzburg.de/fileadmin/10030800/user_upload/quadcopter/Abschlussarbeiten/INS_Lukas_Ifflaender_BA.pdf. Würzburg: Julius-Maximilians-Universität Würzburg, 2012.
- INVENSENSE, IMU-3000: *Gyroscope IMU-3000*. 2015, <http://www.invensense.com/mems/gyro/imu3000.html>:Invensense.
- JAHN, Jasper; BATZER, Ulrich; SEITZ, Jochen; PATINO-STUDENCKA, Lucila; GUTIÉRREZ BORONAT, J.: *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on IEEE*, 2010.

- KIM, Jeong W.; JANG, Han J.; HWANG, Dong-Hwan; PARK, Chansik: A step, stride and heading determination for the pedestrian navigation system. *In: Positioning*, (2004), Nr. 1, S. 08, 0.
- KIONIX, KXTJ9: *Accelerometer (KXTJ9)*. 2015,
<http://www.kionix.com/accelerometers/kxtj9>:Kionix.
- KNEŽEVIĆ, Karmen; KRALJEVIĆ, Lidija: DEUTSCH IN DER PHYSIK I.
- KOUROGI, Masakatsu; KURATA, Takeshi: *Proceedings of the 2Nd IEEE/ACM International Symposium on Mixed and Augmented Reality*. Washington, DC, USA: IEEE Computer Society, 2003. ISBN- 0-7695-2006-5.
- LINK, JA B.; SMITH, Paul; VIOL, Nicolai; WEHRLE, Klaus: *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on IEEE*, 2011.
- MAO, Guoqiang; FIDAN, Barış; ANDERSON, Brian D.: Wireless sensor network localization techniques. *In: Computer networks*, (2007), Nr. 51, S. 10, 2529-2553.
- MEISTER, O.: *Entwurf und Realisierung einer Aufklärungsplattform auf Basis eines unbemannten Minihelikopters mit autonomen Flugfähigkeiten*:Logos-Verlag, 2010. - ISBN 9783832526030.
- MERCER, JA; BATES, BT; DUFEK, JS; HRELJAC, A.: Characteristics of shock attenuation during fatigued running. *In: Journal of Sports Science*, (2003), Nr. 21, S. 11, 911-919.
- MEYER, Johannes: *Ortungsverbesserung durch hybride Verfahren*, 2013.
- OJEDA, Lauro; BORENSTEIN, Johann: Non-GPS navigation for security personnel and first responders. *In: Journal of Navigation*, (2007), Nr. 60, S. 03, 391-407.
- PAKALSKI, Ingo: *Android war ursprünglich für Digitalkameras gedacht*. 2014,
<http://www.golem.de/news/andy-rubin-android-war-urspruenglich-fuer-digitalkameras-gedacht-1304-98779.html>:golem redaktion, 2013.
- PARK, Sang K.; SUH, Young S.: Height compensation using ground inclination estimation in inertial sensor-based pedestrian navigation. *In: Sensors*, (2011), Nr. 11, S. 8, 8045-8059.
- RICHARD, Hans A.; SANDER, Manuela: *Technische Mechanik: Festigkeitslehre*:Springer, 2006.

- RUNGE, Martin; REHFELD, Gisela: *Mobil bleiben: Pflege bei Gehstörungen und Sturzgefahr; Vorsorge, Schulung, Rehabilitation*:Schlütersche, 2001.
- SCHMID, Johannes: *In: Ad-Hoc Personenlokalisierung in Drahtlosen Sensornetzwerken*, (2012).
- SENSIRION, SHT21: *SHT2x - Digital Humidity & Temperature Sensor (RH/T)*. 2015, <http://www.sensirion.com/en/products/humidity-temperature/humidity-sensor-sht21/>:sensirion.
- SPASSOV, Ivan: Algorithms for map-aided autonomous indoor pedestrian positioning and navigation. , (2007).
- TEXAS INSTRUMENTS, block d.: *Bluetooth SensorTag block diagram*. 2014, http://processors.wiki.ti.com/index.php/File:Bluetooth_SensorTag_block_diagram.png#filehistory:Texas Instruments, 2012.
- TEXAS INSTRUMENTS, GATT S.: *BLE SensorTag GATT Server*. 2014, http://processors.wiki.ti.com/images/a/a8/BLE_SensorTag_GATT_Server.pdf:Texas Instruments.
- TEXAS INSTRUMENTS, Gyroscope. 2014, http://processors.wiki.ti.com/index.php/File:ST_android_app_screenshot_3.png:Texas Instruments, 2013.
- TEXAS INSTRUMENTS, SensorTag: *CC2541 SensorTag Quick Start Guide*. 2015, <http://www.ti.com/lit/ml/swru324b/swru324b.pdf>:Texas Instruments, 2013.
- Texas Instruments, SensorTag Development Kit: *SensorTag Development Kit*. 2014, <http://www.ti.com/tool/CC2541DK-SENSOR>:Texas Instruments.
- TEXAS INSTRUMENTS, TMP006: *Infrared Thermopile Contactless Temperature Sensor in WCSP Package*. 2015, <http://www.ti.com/product/tmp006>:Texas Instruments.
- U.S. Government: *Official U.S. Government information about the Global Positioning System (GPS) and related topics*. 2015, <http://www.gps.gov/systems/gps/>:Official U.S. Government, 2014.
- VOSKUHL, Sören: *Modellunabhängige Kontextinterpretation in einer Smart Home Umgebung*, 2011.

-
- WANNER, Nico: *Nutzung der Sensorik des MSB430 zur Lokalisierung und Detektieren von sich bewegenden Personen*, http://page.mi.fu-berlin.de/wanner/publications/wanner07_da.pdf. Berlin:Freie Universität Berlin, 2007.
- WEINBERG, Harvey: Using the ADXL202 in pedometer and personal navigation applications. *In: Analog Devices AN-602 application note*, (2002).
- WIKIPEDIA, Magnetometer: *Magnetometer*. 2015, <http://de.wikipedia.org/wiki/Magnetometer>:Wikipedia, 2014.
- WOODMAN, Oliver J.; HARLE, Robert: An introduction to inertial navigation. *In: University of Cambridge, Computer Laboratory, Tech.Rep.UCAMCL-TR-696*, (2007), Nr. 14, 15.
- WU, Yuanxin; HU, Xiaoping; HU, Dewen; LI, Tao; LIAN, Junxiang: Strapdown inertial navigation system algorithms based on dual quaternions. *In: Aerospace and Electronic Systems, IEEE Transactions on*, (2005), Nr. 41, S. 1, 110-132.
- YUN, Xiaoping; BACHMANN, Eric R.; MOORE, Hyatt; CALUSDIAN, James: *Robotics and Automation, 2007 IEEE International Conference on*IEEE, 2007.

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 16 Februar 2015

Amir Baghery