Otto-von-Guericke-Universität Magdeburg

Fakultät für Informatik

Bachelorarbeit

Indoor-Positionsbestimmung mit Hilfe von Bluetooth-Low-Energy-Beacons und Pedestrian Dead Reckoning

Autor:

Anton Anders

29. März 2016

Betreuer:

Prof. Dr. Frank Ortmeier

Institut für Intelligente Kooperierende Systeme (IKS)
Chair of Software Engineering

Marco Filax

Institut für Intelligente Kooperierende Systeme (IKS)
Chair of Software Engineering

Anders, Anton:
Indoor-Positionsbestimmung mit Hilfe von Bluetooth-Low-Energy-Beacons und Pedestrian Dead Reckoning
Bachelorarbeit, Otto-von-Guericke-Universität Magdeburg, 2016.

Inhaltsangabe

Die voranschreitende Digitalisierung betrifft viele Menschen in fast jedem Bereich ihres Lebens. Unternehmen versuchen durch technologischen Fortschritt das Leben vieler zu verändern und es zu verbessern.
Der kulturelle Bereich blieb davon bisher jedoch weitestgehend unbeachtet und entwickelt sich technologisch deutlich langsamer. Vor allem die sinkende Finanzierungsbereitschaft der Regierungen für kulturelle Angebote und Ausstellungen zwingt jedoch die Betreiber zum Umdenken und zur Suche nach neuen und effektiveren Lösungen, um die Besucher zu begeistern und sie langfristig zu binden.
In dieser Arbeit soll ein grundlegendes Konzept für ein System zur Positionsbestimmung in geschlossenen Räumen entworfen und untersucht werden. Die Positionsbestimmung ist eine Schlüsseltechnologie zur digitalen Neugestaltung der Museums- oder Ausstellungsbesuche. Mit einer zuverlässigen Bestimmung der Nutzerposition können personalisierte Führungen gestaltet, die Navigation zu bestimmten Exponaten implementiert oder die Anzeige von Zusatzmaterialien angeboten werden. Das System wird auch von der Laufkundschaft der Ausstellungen einfach und kostengünstig benutzt werden können und eine Positionsbestimmung ohne Verbindung zu einem externen Dienst oder Server ermöglichen.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

BLE	Bluetooth Low Energy
GATT	Generic Attribute Profile
PDR	Pedestrian Dead Reconing
PF	Partikelfilter
RSSI	Receive Signal Strength Indicator
SIR	Sampling Imporance Resampling
SoC	System-on-a-Chip
SUS	Stochastic Universal Sampling
UUID	Universally Unique Identifier
WPD	Windowed Peak Detection

1. Einführung

Ausstellungen verschiedenster Arten ziehen jedes Jahr Millionen Besucher[1] in Deutschland an und bringen so Kultur und historische Ereignisse oder Personen jedem Einzelnen näher. Während eines Ausstellungsbesuches bieten themenspezifische Führungen einen bedeutend höheren Mehrwert für die Besucher dar und vermitteln ganz eigene Eindrücke der Ausstellung, welche ohne Führung verloren gegangen wären. Interessante Zusatzinformationen oder Anekdoten werden oft nicht auf den knapp bemessenen Beschreibungen der Exponate aufgeführt und können nur von kompetenten Personen angemessen vermittelt werden. Diese gehen auf die Besucher ein und bieten ihnen ein ganz besonderes Erlebnis der Ausstellung.
Mit den immer wieder aufkommenden Vorschlägen zur Kürzung der Kultursubventionen müssen immer mehr, vor allem kleine und spezialisierte, Einrichtungen um das eigene Fortbestehen kämpfen und die Ausgaben reduzieren. Die sehr personallastigen und kostenintensiven Führungen können so oft nur durch zusätzliche Führungsgebühren finanziert werden oder müssen in Folge der Kostenoptimierungen gänzlich wegfallen. Um trotz dieser bedenklichen Entwicklungen viele Menschen für Kultur und Geschichte zu begeistern, müssen neue Konzepte gefunden werden, mit denen Besucher ähnliche Erfahrungen machen können, wie dies bei einer Führung möglich ist.
Eine Möglichkeit besteht in der Nutzung von Smartphones. Auf diesen können Inhalte auf den Besucher zugeschnitten dargestellt und wiedergegeben werden. Es sind sogar virtuell geführte personalisierbare Führungen denkbar, bei denen jeder Besucher einen eigenen virtuellen Guide bekommt. Für die Bereitstellung personalisierter Führungen benötigen die Smartphones und darauf ausgeführte Applikationen eine möglichst genaue Positionsbestimmung in Innenräumen. Nur mit dieser sind Unterscheidungen zwischen nahe positionierten Exponaten und Teilausstellungen möglich. Die Positionsbestimmung in geschlossenen Räumen stellt jedoch, anders als die Outdoor-Positionsbestimmung, eine große Herausforderung dar und ist nicht hinreichend und zufriedenstellend gelöst. Bisherige Ansätze beschränken sich oft auf

[1] Laut Statistischer Gesamterhebung an den Museen der Bundesrepublik Deutschland für das Jahr 2014[fM15]

fest installierte Sensoren am Nutzer und sind damit nicht für eine große Laufkundschaft geeignet. Auch radiowellenbasierte Konzepte sind nur bedingt geeignet. Sie sind entweder in ihrer Genauigkeit beschränkt oder haben einen hohen Installations- und Wartungsaufwand und sind damit für viele kleinere Kultureinrichtungen keine geeignete Lösung.
Die in den letzten Jahren aufkommenden Lösungen für Indoor-Positionsbestimmung bieten hingegen eine hinreichend genaue und günstige Lösung zur Positionsfindung in Gebäuden an. Dabei basieren diese auf kostengünstigen iBeacon-Sendern, welche im Gebäude installiert und konfiguriert werden. Nutzer mit einem aktuellen Smartphone und entsprechender Software können diese Sender direkt zur Positionsbestimmung nutzen.
Die bekannten Anbieter wie estimote[2], indoo.rs[3] oder auch Google[4] bauen dazu ein geschlossenes Ökosystem auf und lagern die Positionsberechnung immer auf die unternehmenseigenen Server aus. Damit wird die Positionsbestimmung nur mit einer vorhandenen und aktivierten Internetverbindung nutzbar. Diese ist jedoch nicht in jedem Gebäude gegeben und stellt angesichts der stetig steigenden Leistung der Smartphones eine unnötige Einschränkung dar. Gleichzeitig ist die Positionsberechnung auf den Servern des Anbieters ein potenzielles Datenschutzrisiko und könnte leicht zur Berechnung von personenbezogenen Bewegungsprofilen genutzt werden.
Unter diesen Gesichtspunkten sollte ein neues Positionsbestimmungssystem entworfen werden, welches die negativen Elemente vorhergehender Systeme nicht mehr aufweist und die Anforderungen der Kultureinrichtungen gut erfüllt.

Zielstellung

Als grundlegendes Ziel dieser Arbeit gilt es, ein Positionsbestimmungssystem für gängige Gebäudetypen der Kultureinrichtungen zu entwerfen. Dieses System muss ohne externe Dienste und ohne aktive Netzwerkverbindung die Position eines Nutzers bestimmen können. Zudem muss das konzipierte System massentauglich sein und schnell eingeführt werden können. Dazu dürfen keine zum Smartphone zusätzlichen Geräte oder Sensoren auf Seiten des Nutzers benötigt werden. Die Installation der sonstigen Infrastruktur muss zudem einfach und kostengünstig durchgeführt werden können, damit eine schnelle Verbreitung und hohe Akzeptanz bei den Betreibern von Kultureinrichtungen ermöglicht wird. Die Positionsbestimmung eines Nutzers muss durch die Verwendung eines handelsüblichen Smartphones ohne weitere Modifikationen erfolgen. Der Nutzer soll nur durch die Installation einer bestimmten Software in die Lage versetzt werden, seine Position in ausgerüsteten Gebäuden bestimmen zu können. Dazu soll eine einfache und kostengünstige Infrastruktur von Bluetooth-Low-Energy-Geräten in den Gebäuden und weitere Sensoren der Smartphones genutzt werden.
Das Ziel ist eine maximale Abweichung von 5 Metern zwischen realer und berechneter Position. Es soll eine Sub-Raum-Lokalisierung eines Nutzers ermöglicht werden. Für die Evaluierung des Konzeptes müssen gängige Raumtypen eines öffentlichen Gebäudes identifiziert und das konzipierte System in diesen getestet werden.

[2] http://estimote.com/
[3] http://indoo.rs/
[4] https://developers.google.com/beacons/

Aufbau der Arbeit

Der weitere Inhalt dieser Bachelorarbeit ist folgendermaßen gegliedert: In Kapitel 2 werden alle für das Verständnis benötigten Grundlagen beschrieben. Auf diesen aufbauend wird in Kapitel 3 das konkrete Konzept eines Positionsbestimmungssystems ausführlich beschrieben und die Auswahl der genutzten Ansätze und Algorithmen begründet. In Kapitel 4 wird die Implementierung eines Prototypen des beschrieben Konzeptes skizziert.
Im Anschluss an die Implementierung wird die Evaluierung des Konzeptes in Kapitel 5 beschrieben. Dort werden die untersuchten Raumtypen, die Methoden und alle Ergebnisse dargestellt. Auf Grundlage dieser wird in einer abschließenden Einschätzung des Systems die Erreichung der Ziele untersucht.
Das letzte Kapitel 6 dient einer zusammenfassenden Darstellung des konzipierten Systems und der Ergebnisse. Zudem soll es einen Ausblick auf eine mögliche Erweiterung des Konzeptes und eine Fortführung der Untersuchungen geben.

2. Grundlagen

2.1 Bluetooth Low Energy

In der Literatur wurden viele verschiedene Ansätze([BP00], [YA05], [WYZ+13]) präsentiert, mit denen aktive RF-Sender zur Indoor-Positionsbestimmung genutzt werden. Für alle Ansätzen gilt, dass diese das Radiowellenpropagationsmodell oder Fingerprinting-Algorithmen([BP00]) nutzen, um die Position zu bestimmen. Trotz der Anfälligkeit für externe Störgrößen, erfreuen sich diese Methoden großer Beliebtheit, da sie eine einfache Umsetzung ermöglichen. Durch geeignete Filter- und Fehlerkorrekturalgorithmen können zudem die Einflüsse der Störgrößen minimiert werden.
Durch die Spezifikation des Bluetooth-Low-Energy-Protokollstacks [BLE10a] ist neben WiFi und herkömmlichen Bluetooth-Geräten eine weitere Alternative entstanden, welche sowohl im akademischen Bereich[DM14] als auch bei wirtschaftlich orientierten Unternehmen[1][2] zur Positionsbestimmung genutzt wird.

2.1.1 Beschreibung des Protokolls

Bluetooth Low Energy ist eine Teilspezifikation der Bluetooth 4.0 Core Specification und wurde von der Bluetooth Special Interest Group ausgearbeitet. Diese besonders energiesparende Alternative zum herkömmlichen Bluetooth-Standard wurde für den Einsatz im Gesundheitswesen, Unterhaltungselektronik oder auch das Internet of Things geschaffen[3]. Durch darauf aufbauende Protokolle wie iBeacon (Abschnitt 2.1.2) oder EddyStone (Abschnitt 2.1.3) werden spezielle Probleme adressiert. Diese umfassen vor allem die Berechnung der „Proximity“ zu Sendern und die Übertragung von einfachen Informationen zu beliebigen Nutzern ohne einen vorherigen Verbindungsaufbau.

[1] http://indoo.rs/
[2] http://estimote.com/
[3] https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics/low-energy

2.1.1.1 Protokollstack

Der Protokollstack ist in der Bluetooth 4.0 Core Specification [4] definiert.

Der neu definierte Bluetooth-Low-Energy-Stack hat 5 bzw. 6 Ebenen. Die Grundlegende Strukturierung folgt dabei den gleichen Konzepten wie andere Netzwerk-Stacks. Die Protokolle jeder Ebene bauen aufeinander auf, so dass mit höherer Ebene auch eine höhere Abstraktion erreicht wird.

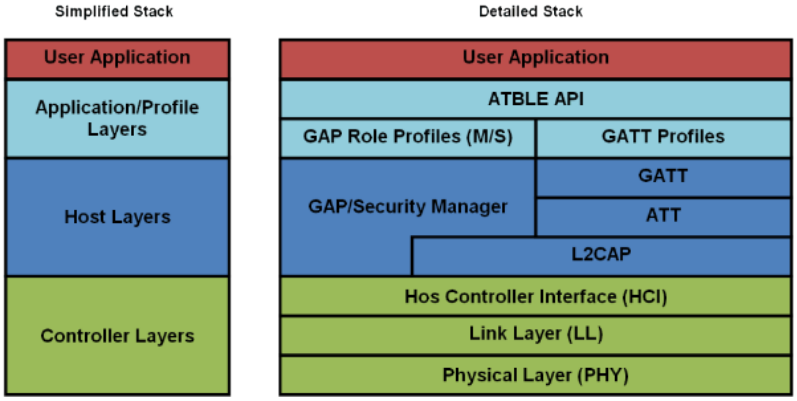


Abbildung 2.1: Bluetooth Low Energy Protokollstack[Inc]

Die im vorgeschlagenen System genutzten Protokolle iBeacon und EddyStone bauen auf GATT[5] auf und definieren eigene Datenfelder, welche in einem Advertisement-Paket übertragen werden.

2.1.2 iBeacon-Protokoll

Das iBeacon-Protokoll von Apple[6] baut auf Bluetooth-Low-Energy-Spezifikationen[BLE10a] auf und definiert ein Advertisement-Paket[iBe15]. Dieses nutzt den 23-Byte großen Payload eines Advertisement-Paketes, um die Identifikation des Beacons und eine Referenzempfangsleistung in einem Meter Entfernung zu übertragen. Diese kann zur Distanzbestimmung mit Hilfe des Radiopropagationsmodells genutzt werden. Zusätzlich definiert das Profil die einzelnen Felder des Advertisement-Headers[Gas14], wodurch das Paket insgesamt 30 Byte lang wird.

2.1.2.1 Aufbau des iBeacon-Advertising Paketinhaltes

Der Paketinhalt eines iBeacon-Paketes besteht grundlegend aus zwei Teilen, welche durch die BLE-Definition begründet sind. Der Advertisement-Header ist eine im Bluetooth Low Energy-Protokoll definierte Datenstruktur und dient der Einleitung der Paketart und der Beschreibung des Payloads. Darauf folgend wird ein 23 Byte langer Payload-Teil übertragen, welcher die Geräte-ID, den Beacon-Typ und die Referenzempfangsleistung in einem Meter Entfernung vom Beacon enthält.

[4] https://www.bluetooth.com/specifications/adopted-specifications?_ga=1.54670699.1379411514.1455791221

[5] Generic Attribute Profile

[6] https://developer.apple.com/ibeacon/

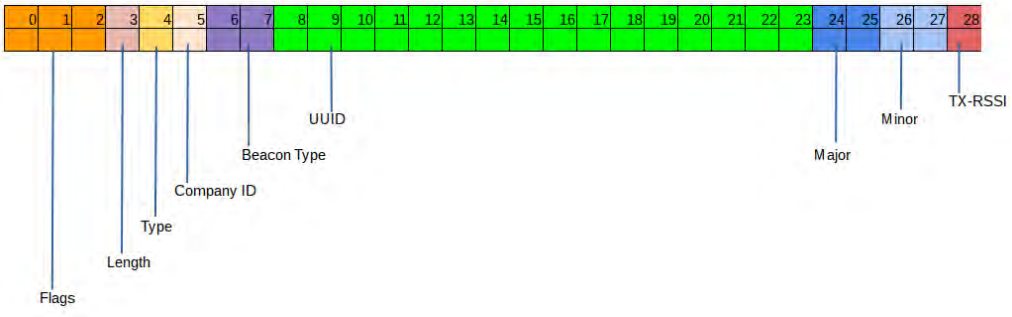


Abbildung 2.2: Aufbau des iBeacon-Paketes

iBeacon Header

Flags (3 Byte) [0x02 0x01 0x1A]
Dieses 3-Byte lange Tupel speichert die Fähigkeiten des iBeacons und überträgt diese. Das erste Byte ist die Längenbeschreibung und definiert, dass 2 weitere Bytes zur Repräsentation der Flags genutzt werden. Das zweite Byte gibt den Typ des dritten Bytes an. In diesem Fall steht der Wert 0x01 für Flags. Das dritte Byte hingegen enthält die eigentlichen Informationen über die Fähigkeiten des Gerätes. Ein handelsüblicher iBeacon beherrscht den Discovery-Mode im Low-Energy und im High-Energy-Bereich. Damit kann das Gerät von Empfängern über den Advertisement-Kanal gefunden werden.
Weiterentwickelte iBeacons beherrschen hingegen noch den Connection-Mode, in dem eine Verbindung mit dem iBeacon über das BLE-Protokoll hergestellt werden kann. Diese Verbindung wird meist zur Konfiguration der Beacons genutzt.

Length (1 Byte)
Das Längenbyte beschreibt die Länge des Payloads abzüglich der 3 Byte für die Flags und eines Bytes für die Länge. Da ein iBeacon-Paket immer 30 Byte groß ist, ist der Wert dieses Feldes immer (dezimal) 26.

Type (1 Byte) [0xFF]
Das Typ-Feld ist im iBeacon-Protokoll festgelegt und hat den Wert 0xFF. Dieser legt fest, dass die übrigen 25 Byte herstellerspezifische Daten sind.

Company ID (2 Byte) [0x004C]
BLE legt fest, dass der Payload mit einer Hersteller-ID beginnen muss. Diese ist 2 Byte lang und muss aufgrund der Lizenzbedingungen immer Apples Hersteller-ID sein.

iBeacon Payload

Beacon Type (2 Byte) [0x02 0x15]
Das erste Byte des Typ-Tupels legt die Art des Beacons fest. Beim iBeacon-Protokoll ist dieses immer auf 0x02 festgelegt, wodurch der Empfänger das Paket als iBeacon-Paket identifizieren kann.
Das 2. Byte dieses Tupels hingegen ist wahrscheinlich noch einmal eine Größenangabe des übrigen Payloads, da der Wert immer (dezimal) 21 ist und

dies zur Anzahl der übrigen Bytes passt. Eine Bestätigung dieser Annahme ist schwierig, da Apple das Protokoll nur bei Unterschrift einer Geheimhaltungserklärung freigibt.

Proximity UUID (16 Byte)
Die UUID, Major- und Minor-Nummern identifizieren den jeweiligen iBeacon eindeutig, wenn diese korrekt konfiguriert sind. Normalerweise wird die gleiche UUID für alle iBeacons genutzt, die demselben Unternehmen gehören. Damit werden die iBeacons grob eingeteilt.

Major Number(2 Byte)
Die Major-Nummer wird für die 2. Stufe der Kategorisierung der Beacons genutzt. Dabei kann ein Unternehmen 2^{16} Kategorien festlegen. Jeder Beacon gehört zu genau einer Kategorie.

Minor Number (2 Byte)
Die Minor-Nummer wird dagegen für die genaue Identifizierung eines Beacons in seiner Kategorie im Unternehmenskontext genutzt. Die Kombination UUID, Major-Nummer und Minor-Number sollte eindeutig sein.

Measured Power (1 Byte)
Das letzte Byte des Payload wird für die Übertragung der Referenz-Leistungsdämpfung genutzt. Dieser Wert kann zur Distanzberechnung genutzt werden. Der Wert dieses Bytes muss für jeden Beacon durch Messungen bestimmt werden. Für das iBeacon-Protokoll muss eine Messung in genau 1m Entfernung durchgeführt werden. Der so bestimmte Wert wird im Beacon gespeichert und ab dem Zeitpunkt immer übertragen. Alternativ kann die Software des Beacons die Referenzleistung theoretisch berechnen.

2.1.3 EddyStone-Protokoll

EddyStone ist ein von Google gestaltetes Konzept der BLE-Beacons welche als direkte Alternative zu iBeacon stehen sollen. Ebenso wie iBeacon nutzt das Protokoll dabei einen der im BLE-Protokoll spezifizierten Advertisement-Kanäle zum Broadcast der Daten.
Dazu werden bisher 3 Frametypen unterschieden, welche für verschiedene Anwendungsszenarien vorgesehen sind. Diese werden im Folgenden kurz beschrieben.

UID Der Frametyp UID überträgt ähnliche Informationen wie ein iBeacon. Es wird eine 16-Byte-Lange Identifizierung des Gerätes gesendet. Zudem wird die Referenzsendeleistung in dBm übertragen. Mit diesem Frametyp ist eine Distanzbestimmung zwischen Sender und Empfänger möglich.

URL Der Frametyp URL wird zur Bekanntgabe von Web-Adressen genutzt. Ein Nutzer kann so direkt angesprochen werden, sobald er sich dem Sender nähert. Zudem wird die Sendeleistung übertragen. Dieser Frametyp ist vor allem für Marketing-Anwendungen konzipiert. Durch die fehlende UUID kann zudem der keine eigene Identifizierungskennung bestimmt werden. Trotz dessen kann

ein Sender eindeutig über dessen MAC-Adresse[7] identifiziert werden. Dies ist jedoch nicht auf allen Geräten möglich[8].

TLM Der TLM-Frametyp ist für die Übertragung der Telemetriedaten eines Beacons konzipiert. Die Telemetriedaten umfassen dabei die Batteriespannung in mV, die Temperatur des Senders, einen Zähler der gesendeten Pakete und die Zeit seit dem letzten Start des Senders. Da keine Sendeleistung übertragen wird, ist dieser Frametyp nicht für eine Distanzbestimmung zwischen Sender und Empfänger geeignet.

Im Weiteren wird der Frametyp UID genau beschrieben, da dieser am besten zum Anwendungsfall passt. Durch die Bereitstellung einer UID und einer Referenzsendeleistung kann die Distanz zwischen Sender und Empfänger berechnet werden.

EddyStone-UID-Header

Flags (3 Byte) [0x02 0x01 0x1A]
Dieses 3 Byte lange Tupel speichert die Fähigkeiten des Beacons und überträgt diese. Das erste Byte ist die Längenbeschreibung und definiert, dass 2 weitere Bytes zur Repräsentation der Flags genutzt werden. Das zweite Byte gibt den Typ des dritten Bytes an. In diesem Fall steht der Wert 0x01 für Flags. Das dritte Byte hingegen enthält die eigentlichen Informationen über die Fähigkeiten des Gerätes.
Die zu setzenden Flags sind im Core Specification Suplement (CSS) für Bluetooth 4.0 im Teil A §1.3 definiert[9].

Length (1 Byte)
Das Längenbyte beschreibt die Länge der Service-UUID-Liste mit der EddyStone-UUID.

Type (1 Byte) [0x03]
Dieses Feld enthält den Datentyp der vollständigen 16-Bit Service-UUID-Liste. Diese hat laut CSS v5 des Bluetooth-Standard den Wert $0x03$ [10]. Darauf folgend ist die 16-bit Service UUID des Beacons zu finden.

Service UUID (Company ID) (2 Byte) [0xAAFE]
Die Service-UUID legt fest, um welche Art des BLE-Gerätes es sich handelt. Im Fall EddyStone ist diese auf den Wert 0xAAFF (Big Endian) festgelegt, wodurch diese Geräte als EddyStone-Beacons identifiziert werden können.

Länge (1 Byte) 0x17
Dieses Feld legt die Länge des übrigen Paketes fest. Für den Fall UID ist die Länge entweder $0x17$ oder $0x15$, je nachdem ob die letzten 2 reservierten Byte gesendet werden oder nicht.

[7] http://www.heise.de/netze/tools/mac/
[8] https://github.com/don/cordova-plugin-ble-central/issues/77
[9] https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=291904
[10] https://www.bluetooth.org/en-us/specification/assigned-numbers/generic-access-profile

Service Data Datentyp (1 Byte) 0x16
Dieser Wert besagt, dass folgend die Daten des Services zu finden sind. Die ersten 2 Byte sind dabei Herstelleridentifikationnummern.

Hersteller ID (2 Byte) 0xAAFE
Der Hersteller wird mit diesem Tupel identifiziert. Im Fall EddyStone ist dieser Wert immer gleich, egal welcher Hersteller die Hardware gefertigt hat.

EddyStone-UID-Payload

Frame Type (1 Byte) 0x00
Für EddyStone-UID wurde der Frametyp 0x00 festgelegt.

TX power (1 Byte)
Sendeleistung des Beacons in 0m Entfernung.

Namespace (10 Byte)
Die folgenden 10 Byte sind für die Definition des Namespaces eines Beacons. Dieser kann als Gruppe für bis zu 2^{10} Beacons betrachtet werden.

Instance-ID (6 Byte) 0x00
Neben dem Namespace muss eine Instanz-ID festgelegt werden. Diese ID muss einzigartig im Namespace sein um ein Gerät eindeutig zu identifizieren.

Reserved (2 Byte) 0x00
Die letzten 2 Byte sind für künftige Protokolländerungen vorgesehen.

2.1.4 RSSI - Reciever Signal Strength Indicator

RSSI ist ein dimensionsloser Wert zur Beschreibung der empfangenen Leistung bei kabelloser Kommunikation[VLB+08]. Bei Bluetooth Low Energy wird der RSSI-Wert beim Empfang jedes Advertisement- oder Datenpaketes bestimmt[BLE10a].
Die Skalierung der RSSI-Werte ist dabei den Chipherstellern überlassen, wodurch unterschiedliche Hersteller verschiedene RSSI-Werte liefern. Aus den Datenblättern kann jedoch entnommen werden, wie diese dimensionsloser Wert in eine Leistung umgerechnet werden kann[VLB+08]. Diese wird in dBm angegeben und stellt das Verhältnis in Dezibel zwischen der empfangenen Leistung in mW und $1mW$ Referenzleistung.
Die Abstraktion der Geräteherstellerschnittstellen durch darauf aufbauenden Betriebssysteme wie Android [11] oder iOS [12] gewährleistet, dass der RSSI-Wert als Leistungspegel in dBm bereitgestellt wird.
Trotzdem müssen für vergleichbare Werte zwischen unterschiedlichen Empfängern Korrekturfaktoren bestimmt werden. Die Unterschiede resultieren aus den verschiedenen Aufbauarten der Geräte, verschiedener Antennenlängen und Bluetooth-SoCs, welche die ermittelten RSSI-Werte beeinträchtigen.

[11] http://developer.android.com/reference/android/bluetooth/le/ScanResult.html#getRssi
[12] https://developer.apple.com/library/ios/documentation/CoreBluetooth/Reference/CBCentralManagerDelegate_Protocol/#//apple_ref/occ/intfm/CBCentralManagerDelegate/centralManager:didDiscoverPeripheral:advertisementData:RSSI:

2.1.5 Distanzbestimmung mittels RSSI

Eine relative Position im Raum kann bestimmt werden, wenn die Distanzen zu Fixpunkten im Raum bekannt sind. Die bekannten Positionen zusammen mit den ermittelten Distanzen fließen durch verschiedene Algorithmen wie Multilateration(Abschnitt 2.1.6.1)[WBLP09] oder auch Partikelfilter(Abschnitt 2.3) in die berechnete Position ein.
Es wurde beschrieben, dass ein Zusammenhang zwischen der Signalstärke beim Empfang von Daten und der Entfernung von Sender und Empfänger besteht. [WYZ+13]

$$RSSI = -(10n \log_{10} d + a) \tag{2.1}$$

Wobei:

n ... umgebungsabhängiger Dämpfungswert
a ... empfangene Signalstärke des Senders in einem Meter Entfernung
d ... Distanz zwischen Sender und Empfänger

Der Empfänger kann durch diesen Zusammenhang aus einer bekannten Signalstärke die Distanz zum Sender berechnen. Dazu muss jedoch eine Referenzsignalstärke an einer bekannten Entfernung bekannt sein. Die Empfangssignalstärke wird von allen Bluetooth-Low-Energy-Stacks beim Empfang von Daten bereitgestellt, da dies in der Spezifikation so festgelegt ist. Die Referenz hingegen muss entweder zuvor ermittelt werden oder diese wird vom aufbauenden Protokoll bereitgestellt[BLE10b].
Die Distanz zu einem Sender wird berechnet, indem der Zusammenhang Formel 2.1 angewendet wird.

$$d = 10^{\frac{a-RSSI}{10n}} \tag{2.2}$$

In der Literatur wird normalerweise $n = 3$ angenommen [EM06], wenn es eine Indoor-Umgebung ist und direkter Sichtkontakt zwischen Sender und Empfänger besteht.

Sollte eine Distanzbestimmung mit demselben Sender aber verschiedenen Empfängern vorgenommen werden, so müssen ggf. Korrekturfaktoren für jeden Empfänger bestimmt werden.

Zudem sollte beachtet werden, dass die Genauigkeit der Messungen und damit auch die Genauigkeit der Distanzbestimmung umso geringer ist, je höher die reale Entfernung zwischen Sender und Empfänger ist[XLL+10].

2.1.6 Positionsberechnungsalgorithmen

2.1.6.1 Multilateration

Multilateration bezeichnet ein geometrisches Verfahren, bei dem ein Punkt im 2-Dimensionalen Raum bestimmt werden kann, wenn die Distanzen zu mindestens 3

Punkten im Raum bekannt sind[ZZZZ08].
Anschaulich beschrieben wird dabei um jeden bekannten Punkt ein Kreis gelegt, welcher als Radius die bekannte Distanz zwischen dem gesuchten und bekannten Punkt besitzt. Der Schnittpunkt dieser Kreise ist der gesuchte Punkt[ZZZZ08].
In realen Anwendungen lassen sich Distanzen zu einzelnen Punkten nur mit hohem Aufwand genau bestimmen. Aus diesem Grund kommen oft Fehlerminimierungsalgorithmen wie das Least-Squares-Verfahren zum Einsatz[ZZZZ08]. Dabei wird nicht der exakte Punkt berechnet, sondern es wird jener berechnet, der am besten zu allen Messergebnissen passt.
In der Literatur wird Multilateration oft genutzt, um die Position im Raum zu bestimmen. Die Tatsache, dass für jede Positionsbestimmung mindestens 3 bekannte Punkte benötigt werden, erhöht im gegebenen Anwendungsfall jedoch die Komplexität der BLE-Beacon-Installationen. Diese müssten sehr dicht verteilt werden, um an jedem möglichen Punkt im Raum eine gültige Positionsbestimmung durchführen zu können.
Durch die hohen Anforderungen an die Installation der Geräte und den damit hohen Kosten- und Wartungsaufwand wird Multilateration nicht weiter betrachtet. Stattdessen wird ein iteratives Positionsbestimmungsverfahren auf Basis der bestimmten Distanzen zu einzelnen BLE-Beacons implementiert, welches fehlende Informationen durch historische Daten kompensiert.

2.1.6.2 Triangulation - Angle-of-Arrival

Für die Berechnung einer Position durch Triangulation werden von der unbekannten Position des Nutzers Winkel zu bekannten Markierungen im Raum bestimmt[WKM08].
Von den Stationen werden Sichtgeraden im Raum mit dem gemessenen Winkel berechnet. Der Schnittpunkt aller Geraden ist dabei die zuvor unbekannte Position des Nutzers[MA98]. Gleichzeitig kann mit diesem Verfahren auch die Ausrichtung des Nutzers bestimmt werden[MA98].
Für die Nutzung von Triangulationsansätzen mit Radiowellen-Sendern benötigt der Empfänger spezielle Antennen, welche die Winkelbestimmung der ankommenden Radiosignale erlaubt[WKM08].

2.1.6.3 Fingerprinting

Das Fingerprinting von Radiosignalen ist eine weit verbreitete Methode zur Positionsbestimmung ([MVFB10][BP00][DM14]). Dazu werden Signalstärken zu verschiedenen Sendern gemessen und mit einer Datenbank verglichen. Um die Position des Nutzers zu bestimmen, werden die gemessenen Signalstärken in der Datenbank nachgeschlagen und die Position herausgesucht, welche am besten zu den Messwerten passt[MVFB10].
Die Datenbank muss im Vorfeld erstellt werden. Diese enthält zu jeder bekannten Position im Raum die gemessenen Signalstärken zu jedem Sender[MVFB10].
Der Vorteil dieser Methode ist vor allem die Einfachheit der Implementierung und die Robustheit gegenüber permanenten Störgrößen wie Wänden oder anderen stationären Objekten[GLG+12]. Es muss jedoch im Vorfeld eine Datenbank erstellt werden, welche sehr detailliert aufgebaut sein muss. Dies stellt einen hohen Aufwand dar und ist nur für sich nicht stark ändernde Umgebungen geeignet. Bei gravierenden Änderungen der Umgebung oder der Positionen einzelner Sender muss die Datenbankerstellung neu durchgeführt werden, um die Fingerprints zu aktualisieren[GLG+12].

2.1.7 Zuverlässigkeit und Genauigkeit

Es wurde gezeigt, dass die Distanzbestimmung mit Hilfe des Propagationsmodells der Radiowellen unzuverlässig ist [DM14] und durch menschliche Körper, Gegenstände und sonstiges Rauschen gestört werden kann. Eine gestörte Distanzbestimmung ist dabei für die darauf aufbauenden Positionsbestimmung schädlich und führt unweigerlich zu einer größeren Unsicherheit der berechneten Position. Die Störeinflüsse können zwar durch Mittelwertfilter oder rekursive Filter (Abschnitt 2.3) gemindert werden, eine akkurate Position kann jedoch auch damit nicht berechnet werden.
Des Weiteren sind diese Ansätze auf dicht verteilte RF-Sender angewiesen. Bei einem Ausfall einzelner Sender kann keine flächendeckende Positionsbestimmung gewährleistet werden.

2.2 Pedestrian Dead Reckoning

Ein weiterer, sehr beliebter Ansatz([PWH12],[Bea06], [QMYL13]) ist die Positionsbestimmung durch ein inertiales System. Dabei werden alle die Bewegungen des Nutzers kontinuierlich aufgenommen, ausgewertet und daraus stets neue Positionen berechnet. Dieser Ansatz zeichnet sich vor allem durch die iterative Art der Positionsbestimmung aus. Zudem werden keine Installationen von Sendern oder Markierungen in den Räumen benötigt. Der Nutzer muss nur inertiale Sensoren wie Accelerometer, Magnetometer und Gyroskope im Smartphone zur Verfügung haben.
Mit Hilfe der erfassten Sensorwerte wird die Bewegungsrichtung und Bewegungsweite berechnet. Bei einer bekannten Startposition $\vec{pos}_0$ zum Zeitpunkt t_0 kann die Position zum Zeitpunkt t_n durch Integration aller dazwischen liegenden Bewegungen berechnet werden.

$$\vec{pos}_t = \vec{pos}_0 + \int_{i=0}^{t} \vec{bewegung}_i dt \tag{2.3}$$

Bei einer theoretischen Betrachtung ist ein Inertiales Positionsbestimmungssystem sehr genau und kann nicht durch externe Störgrößen wie Jamming-Angriffe[13] gestört werden. Diese Betrachtungsweise setzt jedoch ideale Sensoren voraus, die keine Drifterscheinungen zeigen und zudem über eine unendlich kurze Samplingrate verfügen. In realen Versuchen konnte gezeigt werden, dass dies nicht der Fall ist([Woo07]). Vielmehr sind Sensoren immer gewissen Störgrößen ausgesetzt und weisen immer Drifterscheinungen auf. Durch das iterative Konzept beeinflussen Fehler in der Bewegungsberechnung zu einem beliebigen Zeitpunkt alle darauf folgenden Positionen negativ und erhöhen so den Fehler.

2.2.1 Vorverarbeitung der Sensordaten

Für eine zuverlässige und möglichst genaue Position sollten die stark verrauschten und von Störgrößen beeinflussten Sensordaten gefiltert werden. Dieser Prozess wirkt temporären Einflüssen entgegen. Bei inertialen Systemen, welche Beschleunigungsdaten nutzen, werden diese meist durch verschiedenartige Filter von Störeinflüssen bereinigt. Um hochfrequentes Rauschen der Sensoren zu entfernen, werden zum Beispiel Tiefpassfilter oder Medianfilter eingesetzt[WGH+11].

[13] http://www.gps.gov/spectrum/jamming/

Die genannten Filter können auch für die Bereinigung der Sensordaten anderer Sensoren wie Magnetometer oder Gyroskopen verwendet werden. Magnetometerdaten können mit Hilfe einer Filterung von Rauschen und temporären Störgrößen, wie starken Türmagneten, bereinigt werden[ARL11b].
Zusätzlich zur Filterung der Daten sollte die durch Accelerometer gemessene Beschleunigung in dessen zwei Bestandteile aufgespalten werden. Die Beschleunigungsdaten setzen sich im Rohzustand aus der wirkenden gravitationsbedingten Beschleunigung von $-9,81\frac{m}{s^2}$ und der linearen Beschleunigung, welche durch Bewegungen entsteht.

$$\vec{acceleration}_t = \vec{g}_t + \vec{linearAcceleration}_t \tag{2.4}$$

Die durch Gravitation induzierte Beschleunigung kann durch einen Hochpassfilter herausgefiltert werden[PWH12].

$$\vec{g}_t = \alpha * \vec{g_{t-1}} + (1 - \alpha) * \vec{acceleration}_t \tag{2.5}$$

Die lineare Beschleunigung wird durch einfache Subtraktion von $\vec{acceleraton}_t$ und $\vec{g}_t$ bestimmt.

$$\vec{linearAcceleration}_t = \vec{acceleration}_t - \vec{g}_t \tag{2.6}$$

2.2.1.1 Tiefpassfilter

Ein Tiefpassfilter wird oft auch als Glättungsfilter bezeichnet[KR77]. Diese Art von Filtern betrachten eine Datenreihe von $2n + 1$ Datenpunkten um den Wert des n-ten Datenpunktes zu berechnen. Dazu geht jeder ursprüngliche Datenpunkt d_k mit einem Gewichtungsfaktor w_k in Berechnung ein. Diese wird normalerweise als einfache Summation der Werte gelöst[KR77].

$$y'_i = \sum_{k=i-n}^{i+n} w_k * y_k \tag{2.7}$$

Mit dieser Berechnung der gefilterten Daten entsteht eine Verzögerung der Datenwerte. Diese beträgt n Datenpunkte[KR77].

Für die Glättung von Sensordaten wird oft ein Tiefpassfilter über einem gleitenden Fenster genutzt[LCL15][PWH12]. Dabei geht jeder Ausgangswert $2n + 1$-Mal in die Berechnung ein. Über die Ausgangsdatenreihe wird ein Fenster mit fester Länge gelegt und dessen Mittelwert berechnet. Mit jedem neuen Datenpunkt verschiebt sich das Fenster um einen Index und es wird erneut ein Mittelwert berechnet.

2.2.1.2 Medianfilter

Ein Medianfilter nutzt immer $2n + 1$ Datenpunkte um den Wert eines Datenpunktes zu bestimmen. Dazu werden die gewählten Datenpunkte auf oder absteigend sortiert. Der Median des gewählten Datensatzes ist dabei der Wert an der Position $(2n + 1)/2$ [WGH+11]. Dieser Wert wird als gefilterter Wert zur weiteren Verarbeitung genutzt.
Ein Medianfilter wird normalerweise in einem Sliding-Window-Ansatzt implementiert [WGH+11]. Das bedeutet, dass jeder Datenpunkt $2n+1$-Mal in die Berechnung der gefilterten Datenreihe eingeht.

2.2.2 Integration der wirkenden Beschleunigungen

Die naivste Art, eine Bewegung des Nutzers in eine Position umzurechnen, ist die Beschleunigung des Gerätes in allen 3 Achsen zu integrieren. Mit einer gegebenen Startposition $\vec{pos}_{t=0}$ und Startausrichtung kann durch Auswertung der Bewegungen die Position zu einem beliebigen Zeitpunkt t berechnet werden[Cha97]. Dazu können Gesetze der klassischen Mechanik genutzt werden. Nach den newton'schen Gesetzen gilt[Woo07]:

$$\vec{velocity}_t = \vec{velocity}_0 + \int_0^t \vec{acceleration}_i * \delta t \tag{2.8}$$

$$\vec{pos}_t = \vec{pos}_0 + \int_0^t \vec{velocity}_i * \delta t \tag{2.9}$$

Somit kann durch doppelte Integration der wirkenden Beschleunigungen die Position in Bezug zu einer Startposition berechnet werden.
Generell ist diese Art der Positionsbestimmung sehr interessant, da sie nicht von externen Störquellen gestört werden kann[WT04] und nur durch die vorhandenen Sensoren beschränkt wird. Dies ist jedoch auch die große Schwachstelle dieser Verfahren. Die realen Sensoren sind für kontinuierliches Rauschen und auftretende Drifterscheinungen anfällig[Woo07]. Durch die notwendige doppelte Integration der wirkenden Beschleunigungen werden auch alle darin enthaltenen Fehler integriert[Woo07]. Diese wirken ab dem Zeitpunkt ihres Auftretens kontinuierlich störend auf die Positionsbestimmung und können nicht durch folgende Messungen berichtigt werden.

2.2.3 Zero-Velocity-Update Methode

Um dem stetigen Einfluss des Sensordrifts und Rauschens entgegenzuwirken, kann zusätzliches Wissen über die Bewegung genutzt werden. In unserem Anwendungsfall geht es um die gehende Bewegung einer Person.
Es wurde vorgeschlagen, die Integration der Beschleunigungen nur während einer tatsächlichen Bewegung des Nutzers durchzuführen[SHNR10]. Dadurch kann der Einfluss von Störfaktoren während des Stillstandes ausgeschlossen werden. Das System würde weiterhin Sensordaten erhalten, diese aber nicht in die Positionsberechnung einfließen lassen. Durch dieses Vorgehen steigt der Fehler einer berechneten Position nur, wenn sich der Nutzer tatsächlich bewegt[PS10]. Des Weiteren können durch die Bewegungserkennung in der Stillstandsphase die Störfaktoren analysiert und die gewonnenen Einsichten zur Korrektur vorhergehender und folgender Sensormesswerte genutzt werden[SHNR10].
Diese Methode wurde vor allem für am Fuß befestigte Sensoren entwickelt und vorgeschlagen. Nur an dieser Körperstelle kann ein Sensor genau zwischen der Schwingphase und der Stillstandsphase eines Fußes unterscheiden[JTSW11]. Diese exakte Unterscheidung ist aber sehr wichtig für das korrekte Funktionieren des Systems und kann somit nicht auf smartphonebasierte inertiale Systeme angewendet werden[QMYL13].

2.2.4 Schrittbasierte Methode

Um die Einschränkungen der Zero-Velocity-Update-Methode zu umgehen, die grundlegende Idee dieser aber weiter nutzen zu können, werden allgemeinere schrittbasierte Methoden angewendet[QMYL13]. Bei diesen Methoden werden die Bewegungen

nicht direkt aus den Beschleunigungen abgeleitet[Woo07]. Stattdessen wird ein Bewegungsvektor $\vec{v_t}$ für jeden Schritt berechnet, der aus der geschätzten Schrittlänge l und Schrittrichtung ϕ besteht[JTSW11].
Die Position zum Zeitpunkt t kann berechnet werden, indem die Position zum Zeitpunkt $t-1$ um den Vektor $\vec{v_t}$ verschoben wird[JTSW11].

$$\vec{pos}_t = \vec{pos}_{t-1} + \vec{v_t} \tag{2.10}$$

Daraus folgt, dass eine Position zum Zeitpunkt t durch die Startposition $\vec{pos}_0$ und alle Bewegungsvektoren $\vec{v_1}..\vec{v_t}$ berechnet werden kann.

$$\vec{pos}_t = \vec{pos}_0 + \sum_{i=1}^{t} \vec{v_i} \tag{2.11}$$

Für jeden Bewegungsvektor muss in den rohen Beschleunigungsdaten ein Schritt erkannt, die Schrittlänge geschätzt und die Bewegungsrichtung während des Schrittes bestimmt werden[JTSW11].
Das Nutzen einer schrittbasierten Methode hat vor allem den Vorteil, dass Driftfehler und Rauschen nur während eines Schrittes in die Positionsbestimmung einfließen[JTSW11]. Für ein zuverlässiges Ergebnis müssen jedoch geeignete Schritterkennungs- und Schrittlängenschätzalgorithmen gefunden werden. Zudem muss die Bewegungsrichtung möglichst genau bestimmt werden können. Diese Algorithmen müssen für den Smartphone-Anwendungsfall geeignet und vor allem robust gegenüber Änderungen der Sensorlage sein[QMYL13].

2.2.4.1 Schritterkennung

Schritte können vor allem in den Beschleunigungsdaten eines am Nutzer befestigten Sensors erkannt werden. Diese Sensordaten enthalten typische Beschleunigungsverläufe, welche das Anheben und Aufsetzen des Fußes markieren[BH13]. Im Weiteren werden die gängigsten Algorithmen vorgestellt, die zur Schritterkennung genutzt werden können. Diese können in zwei grundlegende Klassen eingeteilt werden. Zum einen kann das so genannte Zero-Crossing-Counting genutzt werden, zum anderen kann ein Peak-Detection-Verfahren zum Einsatz kommen[BH13]. Durch die bessere Eignung der Peak-Detection-Verfahren[BH13] werden diese genauer vorgestellt. Methoden, welche maschinelles Lernen[KIK10], neuronale Netze[Bea06] oder Fouriertransformationen[VK95] nutzen werden hingegen nicht weiter beschrieben, da die hohen Ressourcenanforderungen dieser[BH13] nur schwer auf mobilen Geräten zur Verfügung gestellt werden können.

Zero-Crossing Counting

Die Schritterkennung durch Zero-Crossing Counting wird durchgeführt, indem ein Nulldurchgang in der vertikalen Beschleunigung detektiert wird[Bea06]. Dieser Nulldurchgang tritt zum Zeitpunkt auf, bei dem die betrachtete Person ihren Fuß nach einem zuvor erfolgten Aufsetzen eines anderen Fußes anhebt. Ein Schritt ist dabei zwischen zwei positiven Nulldurchgängen zu finden[Bea06]. Da solche Nulldurchgänge durchaus auch durch andere Aktionen des Nutzers hervorgerufen werden können, müssen diese durch weitere Algorithmen validiert werden[BH13]. Dies kann zum Beispiel auf Basis der Zeitabstände zweier Nulldurchgänge erfolgen[Bea06].

Peak detection

Eine Schritterkennung durch das Erkennen von Minima und Maxima in den Beschleunigungsdaten kann zu sehr guten Ergebnissen führen und kann dabei verhältnismäßig einfach und ressourcenschonend implementiert werden[BH13]. Die Detektion der Minima und Maxima kann durch verschiedenste Algorithmen wie Threshold-Ansätze[PWH12], Windowed Peak Detection[CTC12], Normalized Autocorrelation SC[Boe93] oder Dynamic Time Warping[RZJM07] gelöst werden. Die so bestimmten Peaks müssen darauf folgend durch weitere Schritte verifiziert werden[BH13]. Diese Verifikation erfolgt oft durch die Analyse der Schrittfrequenz[LCL15].
Ansätze, welche auf Machine Learning[KIK10], FFT[VK95] oder Neuronale Netze[Bea06] bauen, sind hingegen weniger für den mobilen Einsatzbereich geeignet, da diese hohe Ressourcenanforderungen haben[BH13].
In den Untersuchungen von A.Brajdic und R.Harle[BH13] wurde festgestellt, dass Windowed Peak Detection sehr gut für das Erkennen von Schritten geeignet ist und besonders durch die Genauigkeit und Performanz bei Smartphoneanwendungen überzeugt. Ein neuer Algorithmus zur Schrittdetektion wurde von Hwan-hee Lee, Suji Choi und Myeong-jin Lee vorgestellt[LCL15], welcher für Smartphone-Anwendungen konzipiert wurde und einen bekannten Windowed-Peak-Detection-Algorithmus[CTC12] sowie zwei properitäre Algorithmen übertrifft[LCL15]. Dieser Algorithmus kombiniert dabei die Konzepte eines Windowed-Peak-Detection-Ansätzen und eines adaptiven Threshold-Peak-Detection-Ansätzen[LCL15].

Thresholdbasierte Schrittdetektion Thresholdbasierte Schrittdetektion erkennt Schritte durch das Auffinden von aufeinanderfolgenden Maxima und Minima in der vertikalen Beschleunigung des Sensors. Die Minima und Maxima werden dabei detektiert, wenn die gemessene Beschleunigung eine festgelegte Grenze unter- bzw. überschreitet[PWH12]. Diese Werte können sowohl statisch festgelegt[KJHP04] als auch dynamisch[JTSW11] berechnet werden.
Die so berechneten Minima und Maxima müssen im Weiteren validiert und ggf. verworfen werden. Diese Validierung basiert meist auf der durchschnittlichen Schrittfrequenz des Nutzers[PWH12].

Windowed Peak Detection Beim Windowed Peak Detection wird ein gleitendes Fenster über die Beschleunigungsdaten gelegt. In diesem betrachteten Bereich wird der Datenpunkt gesucht, der den höchsten bzw. niedrigsten Wert besitzt[AQD+02]. Dieser Datenpunkt wird als mögliches lokales Minimum oder Maximum betrachtet und muss im Weiteren validiert werden[LCL15], damit möglichst keine unnötigen Minima und Maxima detektiert werden[PWH12]. Detektierte und validierte Peaks können als Hinweise auf einen Schritt betrachtet und für weitere Berechnungen genutzt werden.

Peak-Valley Step Detection Das Peak-Valley-Step-Detection-Verfahren von Lee et al.[LCL15] kombiniert die Ideen der Thresholdbasierten und Windowed-Peak-Detection-basierten Schritterkennung und führt zeitgleich die zeitbasierte Validierung der detektierten Minima und Maxima ein.

Es wird angenommen, dass ein Schritt mit einem Maximum beginnt und darauf folgend ein Minimum detektiert werden muss. Wenn beide in einem angemessenen zeitlichen Abstand zueinander gefunden werden, so können diese Datenpunkte als Anfangs- und Endpunkte eines Schrittes gesehen werden.
Die Kandidaten eines Minima oder Maxima werden detektiert, indem drei aufeinanderfolgende Datenpunkte betrachtet werden. Sollte das der mittlere Datenpunkt deutlich höher sein, als die zwei umliegenden, so wird dieser als mögliches Maxima detektiert. Analog dazu werden Minima gesucht.
Die gefundenen Datenpunkte müssen dabei über bzw. unter einer Schranke Th_{min} und Th_{max} liegen, um als mögliche Anfangs-/Endpunkte eines Schrittes zu gelten. Die genannten Schranken berechnen sich dabei auf Grundlage der durchschnittlichen Beschleunigung und der Standardabweichung dieser über die letzten n Datenpunkte. Zusätzlich wird ein Korrekturfaktor α einbezogen. Somit wird erreicht, dass bei verschiedenen Schrittgeschwindigkeiten Schritte zuverlässig erkannt werden[LCL15].
Wenn Kandidaten durch die Thresholds Th_{min} bzw. Th_{max} validiert worden sind, muss eine weitere Validierung bezüglich der Zeitkomponente stattfinden. Diese soll gewährleisten, dass spontan auftretende Minima und Maxima nicht zu einem detektierten Schritt führen. Nur Peaks und Valleys, die in einem definierten Abstand zueinander anzutreffen sind, können valide Schritte sein[LCL15]. Diese Zeitabstände werden durch einen Mittelwert der Zeitabstände vorher detektierter Schritte, der Standardabweichung dieser und eines Korrekturfaktors β validiert. Durch dieses Vorgehen kann gewährleistet werden, dass unterschiedliche Schrittgeschwindigkeiten nicht zu falschen oder fehlenden Detektionen führen. Die dynamisch berechneten Schranken gewährleisten eine gute Detektionsrate auch bei sich ändernden Schrittgeschwindigkeiten[LCL15].

2.2.4.2 Schrittweitenbestimmung

Für die Bestimmung eines Bewegungsvektors während eines detektierten Schrittes muss die Schrittweite bekannt sein. Da die Schrittweite von der Beinlänge[JSPG09], der Laufgeschwindigkeit[Sca07] und der Schrittfrequenz[JSPG09] abhängt, muss diese mit Hilfe von Parametern und den Beschleunigungswerten geschätzt werden. Dazu wurden in der Literatur verschiedene Ansätze vorgestellt. Die gängigsten werden im Weiteren beschrieben:

Statische Methode

Die statische Schätzung ist die einfachste und naivste Methode der Schrittweitenschätzung. Dazu wird angenommen, dass jeder Schritt einer Person gleich lang ist und nur von wenigen Parametern der Zielgruppe abhängt[PWH12]. Eine oft angewendete Berechnungsvorschrift nutzt die Höhe und das Geschlecht des Nutzers und berechnet daraus statisch die Schrittlänge.

$$schrittlaenge = hoehe * k \tag{2.12}$$

Dabei wurde für den Parameter k empirisch ermittelt, dass dieser bei Frauen $k = 0.413$ und bei Männern $k = 0.415$ beträgt[PWH12].

Kim-Methode

Zu den dynamischen Schätzern gehört unter anderem die Kim-Methode[KJHP04]. Bei dieser wird ein Zusammenhang zwischen der Schrittlänge und der durchschnittlichen Beschleunigung während eines Schrittes genutzt. Zudem wird angenommen, dass mit zunehmender Geschwindigkeit einer Person deren Schrittfrequenz höher, deren Schritte länger und die daraus resultierende vertikale Beschleunigung höher wird[KJHP04]. Die entstandene Gleichung zur Berechnung der Schrittweite ist dabei aus experimentell gewonnenen Daten hergeleitet worden.

$$schrittlaenge = k * \sqrt[3]{\frac{\sum_{k=1}^{N} |A_k|}{N}} \tag{2.13}$$

Der Parameter k sollte für die Zielgruppe empirisch bestimmt werden.

Scarlet-Methode

Der von Jim Scarlett 2007 vorgestellte Schrittlängenschätzer[Sca07] ist ebenfalls dynamisch und nimmt einen Zusammenhang zwischen der Schrittweite und der minimalen, maximalen und durchschnittlichen Beschleunigung avg an. Diese ist bei längeren Schritten höher als bei weniger langen Schritten[Sca07].

$$schrittlaenge = k * \sqrt{\frac{(max - min)}{(avg - min)} * \sum\sum(accel - avg)} \tag{2.14}$$

Die dargestellte Gleichung ist ebenfalls empirisch ermittelt worden.

Weinberg-Methode

Bei der Weinberg-Methode[Wei02] wird ebenfalls ein Zusammenhang zwischen der Auftrittsstärke während eines Schrittes und der Schrittlänge angenommen. Bei dieser Methode wird die Auftrittsstärke als die Differenz zwischen dem Minima und Maxima eines Schrittes der Beschleunigungsdaten berechnet. Wenn eine Person schneller geht, so ist deren Schrittlänge weiter. Gleichzeitig ist der beim Auftreten mit einem Fuß entstehende Stoß stärker und die Auftrittsstärke größer.

$$schrittlaenge = k * \sqrt[4]{A_{max} - A_{min})} \tag{2.15}$$

Der Faktor k muss, wie bei den anderen Methoden auch, empirisch bestimmt werden.

Weitere Schätzer

In der Literatur haben Park et al. [PPC+12] und S. Beauregard und H .Haas [BH06] Ansätze vorgestellt, die durch Machine-Learning anhand der Beschleunigungsdaten gute Schätzwerte liefern. Diese werden jedoch nicht weiter betrachtet, da die Analyse der Daten aufwändiger und nur unter Einschränkungen auf mobilen Geräten ausgeführt werden können.

Einschätzung der Methoden

Generell sollten dynamische Schätzer eingesetzt werden, da die Schrittlänge je nach Person, Schrittgeschwindigkeit oder Gangart unterschiedlich sein kann. Ein statischer Schätzer kann solche Unterschiede nicht abbilden[Wei02].
Unter den vorgestellten dynamischen Schätzern liefert die Scarlet-Methode bessere Schätzungen als die Kim- oder Weinberg-Methode[PWH12].

2.2.4.3 Schätzung der Bewegungsrichtung

Neben der Schrittlänge benötigt man auch die Bewegungsrichtung während eines Schrittes, um den Bewegungsvektor zu berechnen. Die Bewegungsrichtung kann durch verschiedene Sensoren des Smartphones bestimmt werden. Zum einen kann ein Gyroskop[Woo07] genutzt werden, um alle Rotationen aufzuzeichnen und in eine Bewegungsrichtung umzurechnen. Des Weiteren kann auch ein Magnetometer[KNHL12] für die Ausrichtung des Smartphones gegenüber dem magnetischen Nordpol verwendet werden.
Die Nutzung des Gyroskopes benötigt eine initiale Ausrichtung des Nutzers, da das Gyroskop nur relative Daten liefert und keine Bezugsquelle im Raum besitzt[Woo07]. Durch ein Magnetometer hingegen kann der Winkel zwischen der Hauptachse des Smartphones und des magnetischen Nordpols berechnet werden[YBM08]. Dieser Winkel ist gleichzeitig die globale Ausrichtung des Gerätes und kann direkt weiterverwendet werden. Für die Berechnung des Winkels wird neben den Sensordaten des Magnetometers auch die Gravitation benötigt, mit der die Rotation des Gerätes im Weltkoordinatensystem berechnet werden kann[YBM08].

Störgrößen

Indoor-Umgebungen sind im Gegensatz zu Outdoor-Umgebungen sehr vielen Störgrößen ausgesetzt[KNHL12]. Vor allem Metallgerüste in den Wänden, Magnetfelder von Lautsprechern und automatischen Türschließanlagen oder Stromleitungen können die Messungen eines Magnetometers stark beeinflussen[KNHL12]. Eine Richtungsberechnung mit gestörten Messungen ist nicht mehr zuverlässig und führt zu Fehlern in der Positionsberechnung.
Das typische Rauschen von Sensoren kann durch weit verbreitete Low-Pass-Filter entfernt werden[JTSW11]. Starke Störgrößen, wie externe Magnetfelder, können hingegen nur mit zusätzlichen Informationen identifiziert und behoben werden. So kann die Kombination von Gyroskop und Magnetometer zu guten Ergebnissen führen[ARL11a], da diese unterschiedliche Modalitäten messen und somit nicht von denselben Störeinflüssen beeinträchtigt werden.

2.3 Partikel Filter

Es wurde in [HB04] und [GGB+02] gezeigt, dass Partikelfilter zur Positionsberechnung genutzt werden können und dabei mindestens genauso gute Ergebnisse wie deterministische Verfahren erreichen. Zudem sind sie gut zur Fusion von verschiedenartigen Informationen geeignet ([HB04], [CDDV07]), solange eine gültige uns sinnvolle Aktualisierungsvorschrift der Partikel auf Basis der Information gefunden

werden kann.
Das Konzept eines sequentiellen Monte-Carlo-Filters, auch Partikelfilter genannt, wird in vielen Quellen([DGA00],[Kün05],[CDDV07],[HB04]) beschrieben. Der grundlegende Aufbau ist bei allen Beschreibungen gleich. Für die einzelnen Bestandteile des Algorithmus wurden dabei jedoch weitere Quellen hinzugezogen. Diese sind im weiteren Verlauf genannt.

Ein Partikelfilter ist eine Menge von Hypothesen zu einer gegebenen Zielstellung und für den Anwendungsfall definierten Berechnungsvorschriften. Die Hypothesen, auch Partikel genannt, stellen dabei eine diskrete und beliebig gute Lösung des Problems im gegebenen Lösungsraum dar. Bei einem für Positionsbestimmung entworfenen Partikelfilter stellt jede Hypothese eine diskrete mögliche Position dar, an der sich ein Nutzer befinden kann. Jeder dieser Hypothesen werden im Verlauf der Sensordatenaufnahme Gewichtungen zugeordnet. Eine hohe Gewichtung zeigt dabei, dass die Hypothese gut zu den gegebenen Sensorwerten passt. Geringe Gewichtungen bedeuten im Umkehrschluss, dass die Hypothese unwahrscheinlich ist[DJ09].
Auf Grundlage der berechneten Gewichtungen jedes Partikels wird ein Neuverteilen der Partikel entsprechend dieser durchgeführt (Sampling Importance Resampling - SIR)[WF10]. Dadurch fallen besonders unwahrscheinliche Hypothesen weg und werden durch wahrscheinlichere ersetzt[WF10].

2.3.1 Initialisierung

Die Initialisierung des Partikelfilters erfolgt, indem Hypothesen zufällig im Lösungsraum emittiert werden. Dabei sollte eine möglichst hohe Abdeckung angestrebt werden, damit Lösungen nicht durch fehlende Hypothesen unentdeckt bleiben. Bei flächenmäßig sehr großen Räumen und Gebäuden kann eine feine Abdeckung des Lösungsraumes zu Laufzeitproblemen führen, da besonders viele Partikel erstellt und betrachtet werden müssen[GGB+02]. Vor allem bei mobilen Geräten ist so etwas nicht erwünscht und kann sogar zum Versagen des Systems durch fehlende Ressourcenkapazitäten führen.
Um diesem Problem entgegenzuwirken, können bekannte Informationen zur Initialisierung hinzugezogen werden, welche den Lösungsraum zum Zeitpunkt t_0 auf einen kleinen Bereich begrenzen[WH08].

2.3.2 Aktualisierung der Partikelgewichte durch neue Informationen

Der Partikelfilter sollte jede neu verfügbare Information nutzen, um die Hypothesen zu prüfen, deren Güte zu berechnen und ggf. neue Hypothesen zu erstellen. Die Güte w der Hypothesen wird generell berechnet, indem ein abstrakter Abstand d zwischen der gemessenen Information und der betrachteten Hypothese berechnet wird. Dabei gilt, dass die Güte umso höher, je geringer der berechnete Abstand d ist[GGB+02].
Durch eine darauf folgende Normalisierung der Gewichte kann die Partikelwolke als eine diskrete Wahrscheinlichkeitsverteilung zur gegebenen Information betrachtet werden[DJ09]. Besonders hoch gewichtete Partikel haben damit eine hohe Wahrscheinlichkeit, dass sie die beste Lösung des gegebenen Problems sind.

Die so berechnete Güte wird im Weiteren zum Sampling Importance Resampling und zur Berechnung der aktuellen Position genutzt.
Die genauen Distanzfunktionen und Gewichtsfunktionen müssen für den Anwendungsfall und die gegebenen Informationen definiert werden.

2.3.3 Neuinitialisierung der Partikel - Resampling

Wie bisher beschrieben, schlagen sich neue Informationen in der Gewichtung aller Partikel nieder. Nach einigen Rekursionsschritten kann jedoch angenommen werden, dass die meisten Partikel ein sehr geringes Gewicht besitzen werden[DJ09]. Nur einige wenige Hypothesen werden sehr hohe Gewichte haben, da sie die zuvor gesammelten Informationen besonders gut repräsentieren. Durch die Funktionsart des Partikelfilters müssen trotz dessen alle Partikel bei jeder neuen Information aktualisiert werden, was im Fall gering gewichteter Partikel meist zu unnötigen Schritten führt, da diese kaum Einfluss auf weitere Berechnungen haben werden.
Um diese unnötigen Aktualisierungen zu umgehen wurde ein sogenanntes Sampling Importance Resampling[DJ09] eingeführt. In diesem Prozess wird der Partikelfilter neu initialisiert, d.h. die Partikel werden neu verteilt. Dies geschieht auf Grundlage der Gewichtungen einzelner Partikel. Diese Gewichtung bestimmt die Wahrscheinlichkeit, mit der ein Partikel in die neue Menge der Hypothesen übernommen wird. Besonders hoch gewichtete Partikel können, je nach Art des Selektionsalgorithmus[DJ09], auch mehrmals in die neue Menge übernommen werden, um deren Bedeutung zu erhöhen.
Um die Diversität der Lösungskandidaten zu erhöhen, kann zusätzlich eine zufällige Mutation der Hypothesen genutzt[WF10] oder es können zusätzliche zufällig generierte Partikel hinzugefügt werden. Dadurch kann eine bessere Durchsuchung des Lösungsraumes ermöglicht und ein vorzeitiges Konvergieren der Partikelwolke verhindert werden[WF10].
Nach dem Resampling-Schritt werden normalerweise die Gewichtungen der Partikel zurückgesetzt, so dass alte Gewichtungen nicht in die neue Gewichtsberechnung eingehen. Alte Gewichtungen, und damit die dafür verantwortlichen Informationen, gehen dabei nicht verloren, da diese in die Partikelposition durch das Resampling eingeflossen sind.
Normalerweise wird ein Resampling-Schritt nach jeder Aktualisierung der Gewichte durchgeführt. Dies kann jedoch bei mobilen Geräten und einer hohen Anzahl der Partikel zu Performanzproblemen führen. Aus dem Grund kann das Ausführen dieses Schrittes an bestimmte Vorbedingungen geknüpft werden. Ein Vorschlag ist das Ausführen eines Resamplingschrittes, wenn die Varianz der Partikelwolke eine bestimmte Grenze übersteigt [DJ09]. Dies führt dazu, dass erst größere Änderungen der Informationen zu einer Verlagerung der Partikelwolke führt.
Für das Durchführen eines SIR-Schrittes wird ein Selektionsalgorithmus benötigt, welcher Partikel mit höheren Gewichtungen wahrscheinlicher auswählt als Partikel mit geringeren Gewichtungen. Im Weiteren werden vier verschiedene fitnessproportionale Selektionsalgorithmen beschrieben. Eine endgültige Auswahl für den Anwendungsfall wird im Konzept beschrieben und begründet.

2.3.3.1 Roulett-Wheel-Selection

Das wohl bekannteste und verbreitetste fitnessproportionale Selektionsmodell ist die so genannte Roulett-Wheel-Selection, auch Glücksradauswahl genannt.
Bei dieser wird jedem Element aus der Ausgangsmenge $e \in P$ eine Selektionswahrscheinlichkeit $p(e)$ zugeordnet[LL11]. Die Summe der Wahrscheinlichkeiten soll dabei 1 sein.
Um n Elemente aus der Ausgangsmenge P auszuwählen, wird die Glücksradselektion n-Mal durchgeführt. Bei jeder Durchführung wird, bildlich gesprochen, ein virtuelles Glücksrad gedreht, welches in $|P|$ Abschnitte unterteilt ist. Die Größe der einzelnen Abschnitte ist dabei proportional zur Selektionswahrscheinlichkeit der jeweiligen repräsentierten Elemente. Bei jeder virtuellen Drehung wird das Element aus der Ausgangsmenge $e \in P$ in die Selektionsmenge P' übernommen, auf dessen Feld das Glücksrad stehen geblieben ist[LL11].
Dieses Verfahren kann auf verschiedene Weisen implementiert werden, wodurch dessen Zeitkomplexität je nach Implementierung unterschiedlich ist. Er kann als naive $linearWalk$-Version mit einer $O(n^2)$-Komplexität, als $binarySearch$-Version mit einer $O(n * log(n))$-Komplexität oder als verbesserte $stochasticAcceptanceSampling$-Version [LL11] mit einer $O(n)$-Komplexität implementiert werden. Vor allem die $O(n)$-Version ist gut zur Selektion von n Elementen geeignet. Dabei müssen jedoch die Schwachstellen dieses Selektionsalgorithmus beachtet werden. So führt dieser ein Bias zu hochwahrscheinlichen Elementen ein, mit dem ein Diversitätsverlust und eine schnelle Konvergierung der Menge einhergeht. Dabei werden hoch gewichtete Elemente mehrmals in die neue Menge übernommen, schwächere werden hingegen kaum gewählt([Tal09]). Die selektierte Menge repräsentiert somit zwar die aktuelle Wahrscheinlichkeitsverteilung der Elemente, verhindert aber gleichzeitig eine weitere Durchsuchung des Lösungsraumes abseits der temporär wahrscheinlichen Elemente.

2.3.3.2 Stochastic Universal Sampling

Stochastic Universal Sampling ist eine Weiterentwicklung der Roulett-Wheel-Selection und adressiert vor allem die negativen Punkte des Vorgängers[Tal09]. Anstatt n-Mal ein Glücksrad mit einem Zeiger zu drehen, wird eines mit n Zeigern genutzt. Dieses wird nur ein mal gedreht. Die Zeiger haben dabei alle den gleichen Abstand zueinander. Das einmalige Drehen bedeutet nur eine Verschiebung aller Zeiger um einen bestimmten Wert.
Jedes Element aus der Ausgangsmenge, auf welches einer der Zeiger zeigt, wird aus der Ausgangsmenge P in die Selektionsmenge P' übernommen. Nach der Selektion gilt damit $|P'| = n$[Bak87].
Stochastic Universal Sampling kann mit einer $O(n)$-Zeitkomplexität implementiert werden, wobei dieser mit nur einer Traversierung der Ausgangsmenge auskommt. Die Einfachheit, Performanz und das nicht vorhandene Bias einer Glücksradauswahl [Tal09] macht diesen Algorithmus zu einem guten Kandidaten eines Selektionsalgorithmus.

2.3.3.3 Tournament Selection

Bei einer Turnierselektion werden mindestens 2 Elemente der Ausgangsmenge $e_1, ..e_k \in P$ zufällig ausgewählt[Bli00]. Dasjenige Element, welches die höchste Güte/das höchste Gewicht besitzt, wird in die Selektionsmenge P' übernommen[MG95]. Es gilt also:

$$e_i \geq e_j \forall j \in [1, .., k]; i \neq j \tag{2.16}$$

Damit die Ausgangsmenge n Elemente enthält, muss ein Turnier n-Mal wiederholt werden ([Tal09]). Dies ist vor allem rechenaufwändig und anspruchsvoll gegenüber dem Randomisierungsalgorithmus, da dieser mindestens $n * k$ Zufallszahlen für die Auswahl der Turnierteilnehmer benötigt.

2.3.3.4 Rank based Selection

Bei einer rangbasierten Selektion wird für jedes Element $e \in P$ ein Rang auf Grundlage seiner Güte berechnet[BHS97]. Dieser Rang ist für die spätere Selektion eines Elementes entscheidend. Je höher ein Rang ist, desto wahrscheinlicher wird ein Element ausgewählt[Bak85].
Die Besonderheit dieses Verfahrens liegt in der Möglichkeit, bestimmte Ränge gemäß ihrer Wichtigkeit für die Anwendung zu skalieren. Theoretisch können damit die besten und schlechtesten Elemente ausgeschlossen werden, da deren Rang als nicht wichtig bezeichnet wird. Dadurch müsste der Algorithmus nur die mittleren Elemente auswählen.
Neben der rein rangproportionalen Selektion können bei dieser Methode weitere Ansätze wie Variable Life Span, nonlinear Ranking Selektion oder die Boltzmann-Selektion zum Einsatz kommen[Tal09].
Der Nachteil dieses Verfahrens liegt vor allem in der exakten Modellierung der Wichtigkeiten einzelner Ränge und dem erhöhten Berechnungsaufwand durch den zusätzlichen Rangberechnungsschritt.

2.3.4 Prediction: Verschieben der Partikel gemäß eines Voraussagemodels

Nachdem die Hypothesen des Partikelfilters mit allen aktuellen Informationen aktualisiert wurden, muss der nächste Iterationszyklus der Positionsberechnung vorbereitet werden. Dabei wird die Partikelwolke derart verschoben, dass sie an der wahrscheinlich nächsten Position ihr Zentrum besitzt[GGB+02].
Mit diesem Schritt wird die vorhergehende Bewegung des Nutzers durch ein Bewegungsmodell einbezogen[Woo07], um eine möglichst gute Ausgangslage für den nächsten Gewichtsbestimmungsschritt zu ermöglichen. Die dahinter stehende Überlegung ist, dass ein Partikelfilter die zum Zeitpunkt t geltenden Informationen zu einer Positionsbestimmung nutzt[DJ09]. Im Zeitraum zwischen t und $t + 1$ bewegt sich der Nutzer jedoch vom Zentrum der Partikelwolke weg. Bei der nächsten Sensormessung und daraus abgeleiteten Informationen würden Partikel auf der einen Seite der Wolke höher bewertet werden als auf der anderen Seite dieser. Beim Resampling würde das Zentrum der Partikelwolke richtigerweise in Richtung der höher gewichteten Partikel verschoben werden.
Sollte die Partikelwolke hingegen soweit konvergiert sein, dass keine Partikel an der

realen Position zum Zeitpunkt $t + 1$ existieren, so würde die Partikelwolke zwar in die richtige Richtung verschoben werden, eine genaue Positionsbestimmung ist damit jedoch nicht möglich.
Aus diesem Grund sollte die Partikelwolke an die wahrscheinlich nächste Position des Nutzers verschoben werden, um eine möglichst gute Abdeckung der im Schritt $t + 1$ betrachteten Region zu erreichen.

3. Konzept

Das Problem der Positionsbestimmung wurde für Outdoor-Umgebungen weitestgehend durch globale Systeme wie GPS[1] oder Galileo[2] gelöst. Für Indoor-Umgebungen sind diese Systeme hingegen nicht geeignet.
In der Literatur wurden verschiedene Ansätze präsentiert, mit denen eine Positionsbestimmung in Gebäuden durchgeführt werden kann. So wurden Pseudolitensysteme[3] konzipiert, welche das GPS/Galileo-Signal durch kleine stationäre Sender in Räumen ausstrahlt. Dieser Ansatz ist jedoch noch in der Entwicklung und ist wegen der hohen Anschaffungskosten nicht für großflächige Installationen und kleine Ausstellungsbetreiber geeignet.
Ein sehr weit verbreitete Methode ist die inertiale Positionsbestimmung (Abschnitt 2.2), welche aus den wirkenden Kräften auf den Empfänger und der bekannten Startposition die veränderte Position des Empfängers berechnen können. Solch ein inertiales System lässt sich mit gängigen Smartphones nutzen, da diese über alle nötigen Sensoren verfügen. Es werden keine weiteren externen Installationen benötigt, wodurch der Installations- und Wartungsaufwand eines solchen Systems minimiert wird.
Ein inertiales Positionsbestimmungssystem ist jedoch, wie zahlreiche Untersuchungen belegen, stark fehleranfällig und kann nicht für eine korrekte Positionsbestimmung über einen längeren Zeitraum genutzt werden (Abschnitt 2.2). Vor allem Sensordrifts und die Rauschanfälligkeit der Sensoren verursachen starke Positionsfehler, welche mit voranschreitender Zeit summiert werden.
Die entstehenden Fehler eines inertialen Systems können durch ergänzende Systeme korrigiert werden, indem zusätzliche Informationen über die Bewegung und Position des Nutzers gewonnen werden.
Alle im Handel erhältlichen Smartphones besitzen eine Vielzahl an Sensoren und Kommunikationsschnittstellen, welche zur Positionsbestimmung genutzt werden können. Zusätzlich zu den inertialen Informationen können auch Informationen aus Teilen des elektromagnetischen Spektrums gewonnen werden. Alle gängigen Smartpho-

[1] http://www.gps.gov/systems/gps/
[2] http://www.esa.int/ger/ESA_in_your_country/Germany/Galileo_Europas_Unabhaengigkeit_und_Kooperation
[3] http://www.iis.fraunhofer.de/de/ff/lok/tech/laufzeit/pseudolites.html

nes besitzen die Fähigkeit, Bluetooth und WiFi-Signale zu senden und zu empfangen. Diese Signale können mit Hilfe verschiedener Algorithmen wie z.B. Triangulation, Multilateration oder Fingerprinting (siehe Abschnitt 2.1.6) zur Positionsberechnung genutzt werden. Da die meisten Geräte jedoch keine speziellen Empfänger zur Bestimmung der Auftrittswinkel der Signale besitzen, kann Triangulation nicht genutzt werden.
Multilaterationsansätze können hingegen Anwendung finden, indem die Entfernung zwischen Sender und Empfänger berechnet wird. Dies geschieht auf Grundlage der empfangenen Signalstärke oder durch Messung der Zeitdifferenz, welche durch die Übertragung eines Signals zwischen Senden und Empfangen entsteht. Die Messung der Zeitdifferenzen muss sehr exakt sein, da bereits kleinste Abweichungen zu starken Fehlern führen[CR12]. Diese Genauigkeit kann auf mobilen Geräten und speziell auf Verbraucher-Smartphones nicht generell vorausgesetzt werden.
Entfernungsmessungen auf Grundlage der Signalstärke sind hingegen weit verbreitet. Die Messung der Signalstärke von WLAN- und Bluetooth-Sendern kann mit jedem Smartphone erfolgen, da sie durch die IEEE 802.11-Spezifikation[4] festgelegt ist.
Das gesendete Signal unterliegt jedoch starken Störeinflüssen und kann durch Wände, Gegenstände oder auch Menschen geschwächt, reflektiert oder absorbiert werden. Daraus ergeben sich Fehler in der Distanzbestimmung, welche zu Ungenauigkeiten in der Positionsberechnung führen. Um die genannten Störfaktoren zumindest teilweise einzubeziehen, wird oft die Idee des Fingerprintings genutzt (vgl. Abschnitt 2.1.6.3). Dieses Verfahren ist zwar potenziell genauer aber auch deutlich aufwändiger in der Durchführung. Zudem müssen Referenzmessungen mit jeder Veränderung der Sender oder Gegenstände neu durchgeführt werden. Dieser Aufwand ist für sich ständig ändernde Ausstellungen nicht vertretbar.
Die Verwendung von Distanzmessungen auf Basis von Signalstärkemessungen ist ein guter Kompromiss zwischen einfacher Handhabung und Genauigkeit. Durch die Fusion eines inertialen Systems zur kontinuierlichen Positionsbestimmung und einem radiowellenbasierten System als Korrekturfaktor sollen potenziell bessere Positionsergebnisse erreicht werden, als dies bei den einzelnen Systemen der Fall ist.
Die Fusion der beiden unterschiedlichen Informationen über die Position des Nutzers soll durch einen rekursiven Monte-Carlo-Filter erfolgen. Dieser ist besonders gut für solche Aufgaben geeignet und ermöglicht die Fusion der Informationen über eine diskrete Wahrscheinlichkeitsverteilung der möglichen Position eines Nutzers. Im Gegensatz zu diskreten Algorithmen für einen bestimmten Anwendungsfall, kann ein Partikelfilter beliebig erweitert werden, indem Aktualisierungsvorschriften für neue Informationen definiert werden (vgl. Abschnitt 2.3 und Abschnitt 2.3.2).
Es wurde zudem bei Untersuchungen gezeigt, dass rekursive Monte-Carlo-Filter keine schlechteren Ergebnisse als diskrete Algorithmen liefern (vgl.Abschnitt 2.3), wodurch die Verwendung dieser keinen Nachteil gegenüber anderen Herangehensweisen darstellt.
Der grundlegende Aufbau des vorgeschlagenen Indoor-Positionsbestimmungssystems ist in Abbildung 3.1 gezeigt. Sowohl die Daten eines RF-basiertes Systems als auch die Daten eines PDR-Systems werden mit Hilfe eines Partikelfilters fusioniert und darauf folgend eine Position berechnet.

[4] http://standards.ieee.org/about/get/802/802.11.html

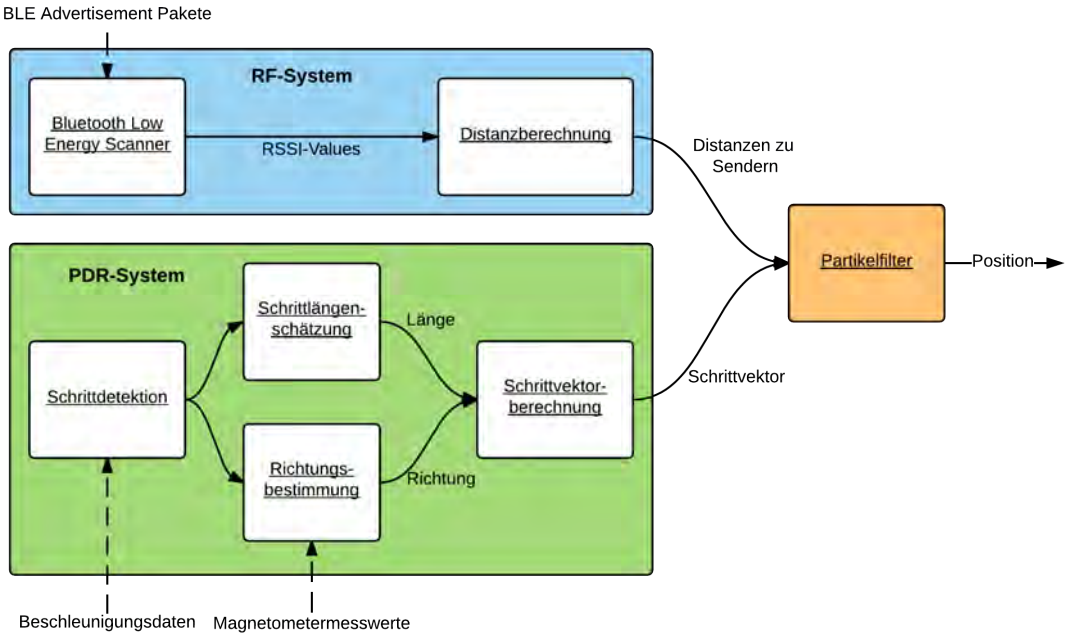


Abbildung 3.1: Allgemeines Konzept des Positionsbestimmungssystems

In dem vorgestellten Konzept soll die Positionsbestimmung eines Nutzers mit Hilfe dessen Smartphones erfolgen. Dazu werden inertiale und radiofrequenzbasierte Systeme kombiniert, um die Genauigkeit der Positionsbestimmung zu erhöhen und die Fehleranfälligkeit zu senken.
Als Radiosender kommen Bluetooth-Low-Energy-Sender zum Einsatz, welche über eine geringe Reichweite verfügen. Somit wird der maximale Fehler bei der Distanzberechnung und damit bei der Positionsbestimmung beschränkt.
Im Weiteren werden die einzelnen Bestandteile des Konzeptes im Detail beschrieben.

3.1 Pedestrian Dead Reckoning

Für eine kontinuierliche Positionsbestimmung ohne hohe Anforderungen an die Umgebung wird ein Pedestrian-Dead-Reckoning-Verfahren angewendet. Dazu werden die Bewegungen des Nutzers durch Beschleunigungsmessungen aufgezeichnet und analysiert.
Der Bewegungsvektor in einem begrenzten Zeitraum wird durch eine schrittbasierte Methode bestimmt. Dazu werden Schritte in den Beschleunigungsdaten detektiert und zu jedem Schritt die Schrittlänge anhand der Beschleunigungswerte dynamisch geschätzt. Für den gleichen Zeitraum wird zudem die Hauptausrichtung des Smartphones bestimmt. Vereinfachend wird angenommen, dass die Ausrichtung des Smartphones der Bewegungsrichtung des Nutzers entspricht. Sonderfälle, wie das Halten des Gerätes im Querformat werden dabei nicht betrachtet. Sie können jedoch durch Nutzung weiterer Sensoren, wie des Gyroskops, und durch das Einbeziehen der Bildschirmausrichtung in die Bewegungsberechnung einfließen.
Die Hauptausrichtung des Smartphones wird mit Hilfe des eingebauten 3-Achsen-Magnetometers berechnet.

Sobald ein Schritt zum Zeitpunkt t detektiert wird, werden für den Zeitraum $t-1$ bis t die Schrittweite l und Bewegungsrichtung ϕ berechnet.

$$\vec{Bewegung}_t = (l, \phi)^T \tag{3.1}$$

Durch die iterative Art der Positionsberechnung kann die Position $\vec{pos}_t$ zum Zeitpunkt t berechnet werden, indem der Bewegungsvektor $\vec{Bewegung}_t$ auf die vorhergehende Position $\vec{pos}_{t-1}$ addiert wird. Dazu muss dieser Vektor zuvor noch aus den Polarkoordinaten in das kartesische Koordinatensystem umgerechnet werden.

$$\vec{pos}_t = \vec{pos}_{t-1} + \begin{pmatrix} l * cos(\phi) \\ l * sin(\phi) \end{pmatrix} \tag{3.2}$$

Dies entspricht auch der in Abschnitt 2.2 vorgestellten Formel 2.3 zur Berechnung der Position aus einer bekannten Startposition und aller darauf folgenden Bewegungsvektoren. Die Varianz des Bewegungsvektors wird der Einfachheit halber nur von der Varianz der Richtungsbestimmung s_ϕ^2 und der Schrittlänge abhängig definiert.

$$Variance_{PDR} = k * (sin(s_\phi) * l)^2 \tag{3.3}$$

Die Richtungsvarianz wird dabei auf Grundlage der berechneten Azimutwinkel[5] $x_1..x_n \in X$ berechnet:

$$s_\phi^2 = E(X^2) - E(X)^2 \tag{3.4}$$

Der Mittelwert μ hingegen wurde zuvor über alle n Messungen gebildet.
Die Varianz der Schrittlängenschätzung muss empirisch bestimmt werden und soll nicht in die weitere Berechnung einfließen.

3.1.1 Vorverarbeitung der Sensordaten

Für die Berechnung der Bewegungsvektoren werden zwei Sensordatenarten verwendet. Die Daten des Accelerometers $acceleration_t$ müssen dabei durch einen Hochpassfilter in zwei verschiedene Komponenten, die Gravitationsbeschleunigung g_t und die tatsächliche lineare Beschleunigung $linearAcceleration_t$, aufgespalten werden. Dies ist in Abschnitt 2.2.1 beschrieben..

Nach einer erfolgreichen Aufspaltung der gemessenen Beschleunigung in die beiden Komponenten, wird auf die in späteren Schritten benötigte gemessene Beschleunigung durch einen Filter von kurzen temporären Störeinflüssen befreit werden. Dazu kommen sowohl ein Medianfilter als auch ein Mittelwertfilter zum Einsatz. Der Medianfilter dient dazu, sehr starke, durch Rauschen verursachte Ausschläge zu beseitigen. Der darauf folgende Mittelwertfilter glättet dagegen den Signalverlauf, um starke Sprünge in diesem auszugleichen.
Sowohl der Medianfilter als auch der Mittelwertfilter agieren durch ein Sliding-Window-Ansatz auf den Rohdaten, wodurch keine für intervallbasierte Filter typischen Sprünge im Signalverlauf auftreten. Die Nutzung beider Filter resultiert in einer Phasenverzögerung von $n * m$ Samples des gefilterten Signals gegenüber dem Rohsignal. Dabei ist $2 * n + 1$ die Größe des Medianfilter-Fensters und $2 * m + 1$ ist die Größe des Tiefpassfilter-Fensters.

[5] Winkel zwischen der Hauptachse des Gerätes und dem magnetischen Nordpool

Die beiden Parameter m und n sollten nicht zu hoch gewählt werden, da sonst erlaubte Ausschläge im Signal zu sehr gedämpft oder sogar eliminiert werden. Dies kann bei schnellen Schritten zu falschen Schrittergebnissen führen.
Des Weiteren wird die Magnitude der Beschleunigung berechnet, um so möglichst unabhängig von der Drehung des Gerätes zu sein.

3.1.2 Schritterkennung

Für die Erkennung der Schritte werden die vorverarbeiteten Beschleunigungsdaten des Smartphones analysiert. Durch ein Peak-Detection-Verfahren können Minima und Maxima in der Magnitude der Beschleunigung als Schritte identifiziert werden. Diese werden zusätzlich durch die Anwendung eines zeitbasierten Filters verifiziert, um die Wahrscheinlichkeit von false-positives zu verringern. Als Schrittdetektionsverfahren kommt das relativ neue Peak-Valley-Step-Detection-Verfahren zum Einsatz, welches vor allem für Smartphone-Anwendungen sehr gut geeignet ist (vgl. Abschnitt 2.2.4.1). Dieses Verfahren benötigt keinen Beschleunigungssensor, der an einer festgelegten Körperregion befestigt ist, wie dies bei vielen anderen Ansätzen der Fall ist. Um Fehler bei der Detektion wichtiger Minima und Maxima zu vermeiden, werden diese verifiziert und nur die vielversprechendsten ausgewählt. Dies ist ebenfalls im gewählten Verfahren vorgesehen und beschrieben.
So detektierte Minima und Maxima werden neben der Schritterkennung auch für die Schätzung der Schrittweite genutzt.

3.1.3 Schätzung der Schrittweite

Um die Bewegung von einer alten Postion in eine Neue berechnen zu können, benötigt man neben der Information, dass ein Schritt gemacht wurde, auch die Schrittweite dessen. Zur Schätzung der Schrittweite soll das in Abschnitt 2.2.4.2 beschrieben Scarlet-Verfahren zur Schrittweitenschätzung genutzt werden. Dieser dynamische Schätzer der Schrittweite kann dabei die zuvor in der Schrittdetektion ermittelten Minima und Maxima der Beschleunigung nutzen. In einer vergleichender Untersuchung (vgl. Abschnitt 2.2.4.2) wurde dieser Schätzer als die beste der untersuchten Methode zur Schrittlängenbestimmung für Smartphoneanwendungen beschrieben.
Der zur Berechnung einer Schrittweite benötigte Faktor k der Scarlet-Methode muss empirisch ermittelt werden und ist für jede Person unterschiedlich. Bei einer großflächigen Verwendung dieses Schätzers müssen Gruppen von Menschen identifiziert werden, welche ähnliche Faktoren haben. Eine solche Klassifizierung kann auf Basis der Körpergröße, Geschlechtes und des Alters stattfinden. Für eine Evaluierung des Konzeptes ist dies jedoch nicht notwendig, wodurch der Faktor im Vorfeld festgelegt wird. Des Weiteren wird ergänzend eine maximale Schrittweite definiert, welche unrealistische berechnete Werte verhindern soll.
Trotz der gefilterten Sensordaten, welche der Schrittweitenschätzung zu Grunde liegen, und der dynamischen Schätzung, können Fehler nicht ausgeschlossen werden. Um temporale Störeinflüsse zu minimieren, wird ein Tiefpassfilter verwendet. Dieser bildet einen Mittelwert über die vorhergegangenen Schrittweiten und gewährleistet, dass zeitlich aufeinanderfolgende Schritte keine zu großen Schrittweiteunterschiede aufweisen. Es wird ein Sliding-Window-Mittelwert berechnet, welcher als Grundlage für den Bewegungsvektor genommen wird.

Störeinflüsse können bei einem im Smartphone eingebauten Beschleunigungssensor durch die Bedienung des Gerätes, durch das Umverlagern dieses von einer Hand in die Andere oder durch das ruckartige Verschieben des Smartphones in der Hosentasche, Handtasche oder dem Rucksack auftreten. Diese beeinflussen die Intensität der Minima und Maxima und führen zu falschen Schätzungen der Schrittweite.

3.1.4 Schätzung der Bewegungsrichtung

Zur Schätzung der Bewegungsrichtung wird ein digitales 3-Achsen-Magnetometer genutzt, welches fast jedes handelsübliche Smartphone inzwischen besitzt. Die Messung des Erdmagnetfeldes und der Gravitationskraft gibt Aufschluss über die Ausrichtung des Smartphones im Weltkoordinatensystem.
Es wird eine Richtung im 2-Dimensionalen Raum auf der Erdoberfläche ermittelt, welche als Winkel ϕ zwischen dem Vektor $\vec{N}$, der zum magnetischen Nordpool zeigt, und dem Geräteausrichtungsvektor $\vec{Device}$ angegeben wird. Dabei gilt die bekannte Gleichung

$$cos(\phi) = \frac{\vec{N} * \vec{Device}}{|\vec{N}| * |\vec{Device}|} \tag{3.5}$$

Durch verschiedene Störgrößen in Indoor-Umgebungen muss das Signal jedoch gefiltert werden und die daraus abgeleiteten Informationen müssen mit einer Güte versehen werden. Diese Güte bestimmt über den Einfluss der Information auf die Positionsbestimmung.
Zum Minimieren der Störeinflüsse wird ein Mittelwertfilter mit einem Sliding Window implementiert, welcher sehr schnelle Änderungen der Ausrichtung abfängt und glättet. Ein dauerhafter Störeinfluss wie ein sehr starkes Magnetfeld in der Nähe von Lautsprechen kann jedoch mit der Methode nicht eliminiert werden. Für die Evaluation des vorgestellten Konzeptes wird gezielt ein Raum ausgewählt werden, der starke Störfaktoren aufweist um die Eignung auf der Sensorart für solche Umgebungen zu untersuchen und das Korrekturkonzept zu testen.

3.2 Radiowellenbasierte Positionsbestimmung

Die Nutzung von Pedestrian Dead Reckoning wird in der Literatur oft zur Positionsbestimmung innerhalb von Gebäuden genutzt. Das System hat jedoch, unabhängig von der Umsetzung, einige Schwachstellen, die es zu beheben gilt. So benötigt das System immer eine möglichst exakte Startposition um daraus folgend alle weiteren Positionen errechnen zu können. Zudem wird die bestimmte Position mit voranschreitender Zeit immer ungenauer, da vor allem Sensordrifterscheinungen und externe Störfaktoren wie Magnetfelder die Genauigkeit stark beeinträchtigen. Um diese Schwachstellen bei der Positionsbestimmung zu beheben, soll ein weiteres System unterstützend wirken und auftretende Fehlerquellen durch Korrekturen wieder ausgleichen.
Dazu wird ein ebenfalls gut beschriebenes Konzept zur Positionsgewinnung mit Hilfe des Radiowellenpropogationsmodells verwendet(siehe Abschnitt 2.1.4) und abgewandelt. Im Gebäude werden, an bestimmten Stellen und zufällig verteilt(vgl. Abschnitt 3.4), Bluetooth-Low-Energy-Beacons installiert, deren Positionen bekannt sind. Diese Beacons werden sowohl zur initialen Positionsbestimmung genutzt, als

auch zur Korrektur der durch PDR bestimmten Position.
Wie in Abschnitt 2.1.5 beschrieben, können bei bestimmen BLE-Protokollen wie iBeacon(Abschnitt 2.1.2) oder EddyStone (Abschnitt 2.1.3) eine Distanz zwischen Sender und Empfänger berechnet werden. Diese berechnete Distanz kann, zusammen mit den bekannten Positionen der Sender, zum Eingrenzen des Lösungsraumes im betrachteten Zeitpunkt führen. Eine mittels PDR berechnete Position, welche nicht im eingegrenzten Lösungsraum liegt, ist dementsprechend wahrscheinlich falsch und muss korrigiert werden.
Um das Startpunktproblem zu lösen, werden ebenfalls Distanzen zu installierten BLE-Beacons gemessen. Beim Erfassen von Distanzen zu einem oder mehreren Sendern kann so ein kleiner Bereich im Gebäude bestimmt werden, welcher als Startbereich in Frage kommt. Dieser kleine Bereich von wenigen Metern ist vor allem dem geringen Senderadius der BLE-Beacons geschuldet und erwünscht.

3.2.1 Vorverarbeitung der Sensordaten

Radiowellen sind verschiedenen Störfaktoren ausgesetzt, die eine theoretisch zentimetergenaue Positionsberechnung unmöglich machen. Durch die gewählte empfangsleistungsabhängige Distanzbestimmung zwischen Sender und Empfänger kommen verschiedene Störgrößen vor. So werden die im 2,4GHz-Frequenzspektrum gesendeten Daten stark von menschlichen Körpern, Wänden und sonstigen Gegenständen absorbiert oder reflektiert. Dadurch entsteht ein signifikanter Leistungsverlust während der Übertragung. Die so gedämpfte Leistung wird im Empfänger gemessen und führt zu falschen Schlussfolgerungen der Distanzberechnung. Die berechnete Distanz ist deutlich höher als die reale, da die empfangene Leistung mit zunehmender Distanz abnimmt.
Zudem führen Reflexionen der Signale zum sogenannten Multi-Path-Problem. Dabei wird ein Datenpaket mehrmals empfangen, weil es zuerst direkt und darauf folgend über einen zweiten, längeren, Pfad zum Empfänger gelangt ist. Das zweite Empfangene Paket lässt dabei ohne Filterung auf eine deutlich höhere Distanz schließen, als dies real der Fall ist. Um die beschriebenen temporären Störfaktoren zu mindern, soll ein Tiefpassfilter genutzt werden. Dieser sollte mit einem Sliding Window für jeden Sender ein Mittelwert der empfangenen Leistung w_{dBm} berechnt und diesen Mittelwert für weitere Berechnungen zur Verfügung stellen. Diese empfangene Leistung wird dabei in der Einheit dBm erfasst und als RSSI durch die Betriebssystemschnittstellen bereitgestellt.
gleichzeitig zur Filterung der Messwerte auch die Varianz dieser berechnet und zusammen mit dem Mittelwert zur Verfügung gestellt werden.

$$Variance_{RSSI} = E(w_{dBm}^2) - E(w_{dBm})^2 \tag{3.6}$$

Die Varianz der Distanz kann mit Hilfe der RSSI-Distanz-Umrechnungsfunktion Formel 2.2 berechnet werden. Dazu wird für jeden Rohwert vor der Vorverarbeitung die zugehörige Distanz bestimmt. Diese fließt in die Varianz ein.

$$Variance_{Distance} = E(d(RSSI)^2) - E(d(RSSI))^2 \tag{3.7}$$

3.2.2 Distanzbestimmung zwischen Sender und Empfänger

Die Entfernung zwischen einem BLE-Beacon und einem Smartphone wird mit Hilfe des Radiopropagationsmodells berechnet. Dazu wird die Leistung der Übertragung eines Advertisement-Paketes auf Empfängerseite gemessen. Durch eine aus dem Advertisement-Paket extrahierte Referenzleistung wird mit der in den Grundlagen beschriebenen Formel 2.2 die Distanz zwischen Sender und Empfänger berechnet.

Dafür wird die umgebungsabhängige Variable n bestimmt. Normalerweise wird $n = 3$ angenommen (vgl. Abschnitt 2.1.5), wenn es eine Indoor-Umgebung ist und direkter Sichtkontakt zwischen Sender und Empfänger besteht. Die Variable a ist dagegen vom jeweiligen Sender abhängig und stellt den Referenz-RSSI-Wert dar. Diese wird ermittelt, indem die Leistungsdämpfung in einem Meter vom Sender gemessen wird. Dieser Parameter kann entweder in einer Datenbank gespeichert oder aber aus den Advertisement-Paketen des iBeacon-Protokolls (Abschnitt 2.1.2) extrahiert werden.

3.2.3 Nutzung der gewonnen Informationen

Die mittels RSSI-Werten berechneten Distanzen können im vorgestellten Positionsbestimmungssystem für zwei verschiedene Vorgänge genutzt werden.

Initialiale Position für PDR

Für die initiale Berechnung einer Position müssen möglichst zuverlässige Distanzen d_j zu jedem sichtbaren Beacon j bestimmt werden. Im 2. Schritt werden - geometrisch gesehen - Kreise K_j um die Positionen der bekannten Beacons gelegt, welche den Radius d_j besitzen. Die Schnittfläche aller Kreise ist dabei der wahrscheinlichste Bereich des Lösungsraumes, in dem die reale Position des Nutzers sein kann.
Die initiale Position des Nutzers wird dabei als der Mittelpunkt des bestimmten Bereiches angenommen. In der Literatur ist dieses Vorgehen als Multilateration bekannt und benötigt in der Regel mindestens 3 gleichzeitig sichtbare Sender. Im vorgeschlagenen System wird dieser Prozess der Multilateration durch einen Partikelfilter ersetzt (siehe Abbildung 3.2), welcher aus allen bekannten Distanzen einen Bereich bestimmt, in dem die wahrscheinlich reale Position des Nutzers liegt. Dieses Verfahren ist in Abschnitt 2.3.1 beschrieben und benötigt mindestens einen nahen sichtbaren Sender.

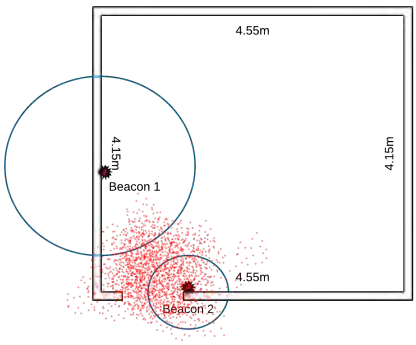


Abbildung 3.2: Partikel nach Initialisierung durch im Eingang positionierte Beacons

Korrektur der zuvor berechneten Position

Sobald die Positionsbestimmung durch Pedestrian Dead Reckoning gestartet wurde, werden die Distanzinformationen zu einzelnen Beacons nicht mehr vorrangig zur Positionsbestimmung genutzt. Stattdessen werden diese zur Korrektur der iterativ berechneten Postion verwendet.
Wenn die zum Zeitpunkt t berechnete Position deutlich weiter vom Beacon entfernt ist, als die berechnete Distanz, so muss die Position entsprechend verschoben werden. Diese Verschiebung gewährleistet eine verbesserte Positionsberechnung durch PDR in den nachfolgenden Schritten. Gleiches gilt, wenn die Position zu nah am Beacon ist.
Die beschriebenen Korrekturen werden durch den bereits erwähnten Partikelfilter durchgeführt. Alle Partikel müssen dazu entsprechend ihrer Differenz der RSSI-Distanz und der Entfernung zum Beacon gewichtet werden. Eine geringe Differenz hat dabei eine hohes Gewicht zur Folge.

$$\delta Distanz = Distanz_{RSSI} - Distanz_{Partikel} \tag{3.8}$$

Durch das im Partikelfilter übliche Resampling wird die berechnete Position in Richtung der höher gewichteten Partikel verschoben. Die Entfernung zwischen der Position und der gedachten Kreisoberfläche verringert sich(siehe Abbildung 3.3).
Für diesen Schritt sollten nur BLE-Beacons betrachtet werden, die eine maximale Distanz nicht überschreiten. Diese ist wichtig, da die Zuverlässigkeit der Messungen mit höherer realer Distanz abnehmen und so die Positionsbestimmung zu stark negativ beeinträchtigen können. Die maximale Distanz sollte zwischen einem und drei Metern liegen. In diesem Bereich kann eine hohe Zuverlässigkeit der Messwerte angenommen werden(siehe Abschnitt 2.1.5).

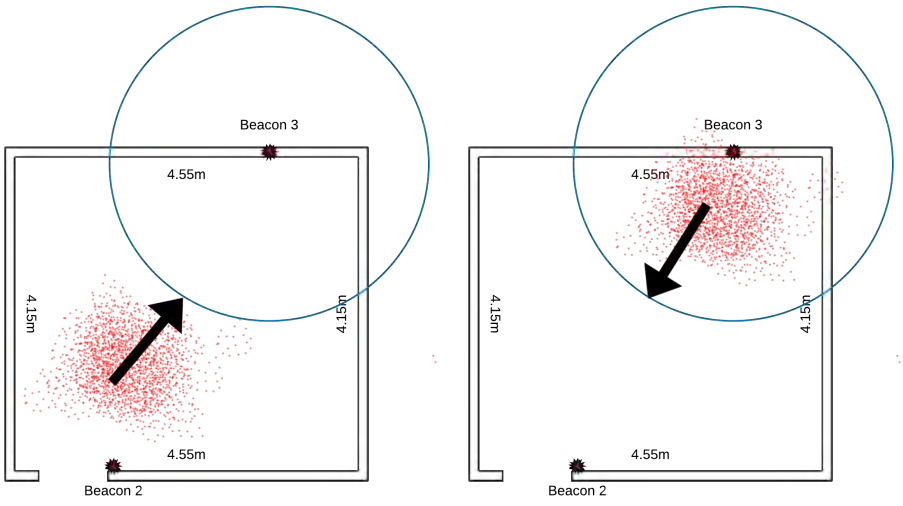


Abbildung 3.3: Korrektur der Partikel durch Distanzbestimmung zu einem Bluetooth-Low-Energy-Sender

3.3 Lokalisierung mit Hilfe eines Partikelfilters

Die Berechnung der wahrscheinlichsten Position im Lösungsraum wird mit Hilfe eines rekursiven Monte-Carlo-Filters(vgl. Abschnitt 2.3), auch Partikelfilter genannt,

durchgeführt. Dazu werden alle aktuell verfügbaren Informationen der Sensoren, aber auch die historischen Informationen aus vorhergehenden Positionsberechnungenen einbezogen.
Der Partikelfilter nutzt alle zum Zeitpunkt verfügbaren Information, um jede Hypothese zu prüfen und deren Wahrscheinlichkeit des Zutreffens zu berechnen. Die Partikelwolke wird daraufhin in Richtung der höchstgewichteten Partikel verschoben, wodurch eine neue Position auf Basis dieser berechnet werden kann.
Einen Partikelfilter zeichnet vor allem aus, dass dieser verschiedenartige Informationen verarbeiten kann, um eine Position zu berechnen. Diese Eigenschaft wird im vorgestellten Konzept genutzt, indem inertiale Informationen und Distanzinformationen zu bekannten Punkten im Raum fusioniert werden. Inertiale Informationen stehen dabei aus der Vorverarbeitung der Daten in Form eines Bewegungsvektors zur Verfügung. Die Distanzinformationen sind hingegen ein Tupel aus der bekannten Senderposition und der berechneten Distanz. Diese führen zur Berechnung neuer Partikelgewichte und der damit verbundenen Aktualisierung der Partikelpositionen durch das sogenannte Resampling.
Um eine zeitliche Versetzung zwischen realer und berechneter Position zu vermeiden, wird zudem ein Voraussageschritt eingebaut. Dieser analysiert die vorhergehende Bewegung und berechnet den wahrscheinlich nächsten Bewegungsvektor des Nutzers. Dieser Bewegungsvektor wird zu den Partikelpositionen addiert. Die so entstehende verschobene Partikelwolke sollte sich an der nächsten wahrscheinlichen Position befinden.

3.3.1 Initialisierung des Partikelfilters

Ein Partikelfilter wird in der Literatur oft initialisiert, indem Partikel im gesamten Lösungsraum zufällig nach einer stetigen Gleichverteilung verteilt werden. Um eine genügend feine Auflösung der möglichen Positionen zu erreichen und dabei die Menge der Partikel gering zu halten, können initiale Informationen genutzt werden. Dadurch werden Partikel nur an den Orten erstellt, die beim Betrachten der apriori-Informationen als nicht unwahrscheinlich erscheinen.
In diesem System werden dazu vor der Initialisierung BLE-Beacons detektiert und die Distanz zu diesen bestimmt. Die Partikel des Partikelfilters müssen damit nur noch in der Nähe aller sichtbaren Beacons emitiert werden. Diese Einschränkung begrenzt vor allem bei sehr großen Lösungsräumen die Anzahl benötigter Partikel stark. Diese müssen nicht mehr Bereichen emittiert werden, in denen die entdeckten Beacons nicht sichtbar sein können, weil deren Reichweite nicht bis zu dem Bereich reicht.
Die Positionen der Partikel werden dabei durch eine normalverteilte Zufallsvariable bestimmt. Eine Position des Partikels ist umso wahrscheinlicher, je geringer die Differenz zwischen der berechneten RSSI-Distanz d_{RSSI} und der Distanz zwischen dem Beacon und der Position $d_{Partikel,Beacon} = |\vec{Partikel} - \vec{Beacon}|$ ist.
Gleichzeitig wird auch die Güte der berechneten RSSI-Distanz mit Hilfe der Standardabweichung in den Emittierungsprozess der Partikel einbezogen. Eine sehr genaue Distanzbestimmung hat dabei eine geringere Standardabweichung der Zufalls-

variable zur Folge, als eine sehr unsichere berechnete Distanz.
Der genaue Algorithmus ist im Weiteren als Pseudocode angegeben.

Input : $SampleSize$
Output : Particles
Result : Array of particles with calculated positions
`/* Scan for BLE-Beacons until you find one */`
repeat
  $Beacons \leftarrow ScanForBeacons()$;
until $Beacons.count > 0$;
`/* Emit Particles around every Beacon at the distance` $radius$
  `calculated from RSSI with a gaussian distributed noise factor */`
$samplesPerBeacon \leftarrow SampleSize \div Beacons.count$;
for $Beacon \in Beacons$ **do**
  $radius \leftarrow CalculateDistanceFromRSSI(Beacon.RSSI)$;
  $Particles \leftarrow emitParticles(Beacon.Position, radius, samplesPerBeacon)$;
end
`/* Resample the Particles to get a correct distribution of particles`
  `for the prior knowledge. This step is done to let the Particles`
  `converge into the correct distribution */`
for i *from 1 to* n **do**
  $Particles \leftarrow calculateParticleWeights(Particles, Beacons)$;
  $Particles \leftarrow resampleParticles(Particles)$;
end
Algorithmus 1 : Initialisierungsprozedur des Partikelfilters durch n verschiedene Bluetooth-Low-Energy-Sender

3.3.2 Aktualisierung der Partikel durch Informationen aus einer Beobachtung

Nach der initialen Verteilung der Partikel müssen diese mit jeder neuen Information aktualisiert werden. Dies geschieht dabei für jeden Informationstyp einzeln. Damit umgeht man die Schätzung der Güte verschiedener Informationen im Vergleich zueinander, um die Gewichtungen einzelner Partikel festzulegen.
Die Gewichtungen der einzelnen Hypothesen werden aus der Güte der Information und dem Abstand dieser zur Hypothese berechnet. Dabei gilt, dass eine Gewichtung umso kleiner, je größer der Abstand zwischen Information und Hypothese ist.

3.3.2.1 Aktualisierung der Partikelgewichte bei neuen PDR-Informationen

Die Gewichte der Partikel werden immer aktualisiert, wenn ein neuer Schritt des Nutzers erkannt wird. Zu diesem Schritt einhergehend wird auch ein Bewegungsvektor $\vec{v}$ (siehe Formel 3.2) berechnet.
Mit Hilfe des Bewegungsvektors $\vec{v}$ und der zuletzt gültigen Position $\vec{pos}_{t-1}$ wird eine neue Position $\vec{pos}_t$ im Raum berechnet, die zur Bewegung des Nutzers passt.

$$\vec{pos}_t = \vec{pos}_{t-1} + \vec{v} \tag{3.9}$$

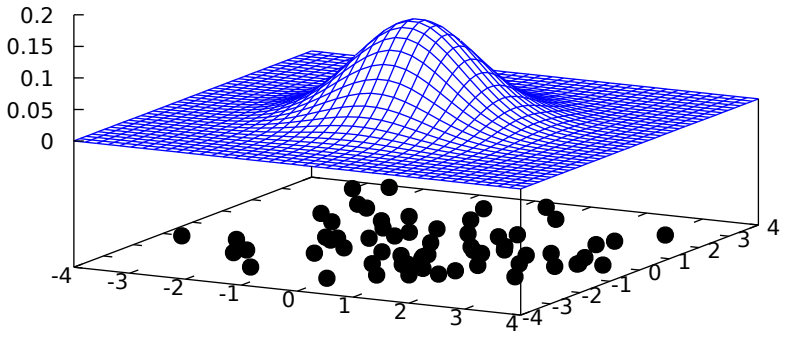


Abbildung 3.4: Diskrete Partikel im Lösungsraum (schwarz) und eine Dichtefunktion der vorliegenden Information (blau) zur Bestimmung der Gewichte einzelner Partikel

Die Gewichtungen der einzelnen Partikel werden daraufhin unter Berücksichtigung der Entfernung zwischen Partikel $\vec{partikel}$ und der berechneten Position $\vec{pos}_t$ bestimmt. Dabei fließt auch die Güte des Bewegungsvektors in die Gewichtsbestimmung mit ein.
Sollte ein Bewegungsvektor sehr ungenau sein, so erhöht sich die Standardabweichung der Dichtefunktion zur Berechnung der Gewichtung, wodurch die Gewichtsverteilung flacher wird und die Gewichtungen einzelner Partikel ähnlich hoch zueinander werden. Die Standardabweichung σ ist dabei direkt von der $Variance_{PRD}$ des Bewegungsvektors abhängig (siehe Formel 3.12).

$$w_{partikel,t} = gewicht(d_{partikel,pos,t}, \sigma) \tag{3.10}$$

$$d_{partikel,pos,t} = |\vec{partikel} - \vec{pos}_t| \tag{3.11}$$

$$\sigma = \sqrt{Variance_{PRD}} \tag{3.12}$$

wobei

$$gewicht(d, \sigma) \sim N(0, \sigma) \tag{3.13}$$

Es ist zu beachten, dass die Varianz des Bewegungsvektors vor allem von den Sensorwerten des Magnetometers abhängen. Die vom Schrittlängenschätzer eingebrachte Varianz kann hingegen nicht durch eine Analyse der Accelerometerdaten bestimmt werden. Diese muss in empirischen Versuchen ermittelt werden. Für das aktuelle System wird jedoch die Varianz der Schrittschätzung nicht weiter betrachtet.

3.3.2.2 Aktualisierung der Partikelgewichte bei neuen BLE-Informationen

Bei dem RSSI-basierten Distanzmessverfahren werden bei der Tiefpassfilterung aller aufgenommenen Werte auch die Varianz $Variance_{RSSI}$ des Mittelwertes für jeden Sender berechnet.
Die berechnete Distanz zwischen dem sendenden Beacon und dem Empfänger besitzt eine von der Varianz eingeflossener RSSI-Werte $Variance_{RSSI}$ (Formel 3.6) abhängige Varianz der Distanz $Variance_{Distance}$(Formel 3.7). Die letztere Varianz bestimmt dabei direkt die Gewichtungen der einzelnen Partikel. Die Berechnungsvorschrift für die Gewichtungen der Partikel unterscheidet sich dabei kaum von den in Abschnitt 3.3.2 vorgestellten Vorschriften. Einzig die Distanzfunktion wird anders definiert, und an die Informationsart angepasst.

$$w_{Partikel,t} = gewicht(|d_{Partikel,Sender,t} - d_{Sender,RSSI,t}|, \sigma) \tag{3.14}$$

$$d_{Partikel,Sender,t} = |\vec{Partikel} - \vec{Sender}| \tag{3.15}$$

$$d_{Sender,RSSI,t} = 10^{\frac{a-RSSI_t}{10*n}} \tag{3.16}$$

$$\sigma = \sqrt{Variance} \tag{3.17}$$

wobei

$$gewicht(d, \sigma) \sim N(0, \sigma) \tag{3.18}$$

3.3.3 Resampling der Partikel nach jeder Aktualisierung der Gewichte

Beim Resampling wird ein Partikelfilter neu initialisiert. Dabei werden jedoch die vorherigen Positionen der Partikel und deren Gewichte berücksichtigt.
Ein Partikel wird dabei umso wahrscheinlicher wieder an seiner vorherigen Position erzeugt, je höher dessen Gewichtung ist. Dadurch werden besonders hoch gewichtete Hypothesen ggf. mehrfach erzeugt. Partikel mit geringen Gewichtungen können hingegen auch ganz entfallen. Dadurch konvergiert die Partikelwolke und die Positionsgenauigkeit wird höher.
Für unseren Anwendungsfall wurde der Selektionsalgorithmus Stochastic Universal Sampling (vgl. Abschnitt 2.3.3.2) gewählt. Dieser Algorithmus ist fitnessproportional und umgeht gleichzeitig die Bias-Probleme einer Glücksradauswahl. Zudem stellt er keine hohen Performanzanforderungen an die Zufallszahlenquelle des Smartphones, wie das bei einer Turnierauswahl der Fall ist. Dies ist vor allem wichtig, da die Zufallszahlengenerierung von der vorhandenen Entropie abhängt und beim Fehlen dieser zum Flaschenhals der Anwendung werden kann.

Ebenfalls positiv am Stochastic-Universal-Sampling-Algorithmus ist, dass die selektierte Menge der Größe n mit einer Zeitkomplexität $O(n)$ berechnet werden kann. Der genaue Resampling-Algorithmus ist in Abschnitt 2.3.3.2 zu finden.

3.3.4 Berechnung der Position des mobilen Empfängers

Die Positionsberechnung bezieht jedes Partikel und dessen Gewichtung ein. Die Position ist dabei der gewichtete arithmetische Mittelpunkt der Partikelwolke. Da die Partikelwolke eine diskrete Wahrscheinlichkeitsverteilung darstellt, wird die Position wie der Erwartungswert dieser berechnet. Dabei sollen Partikel mit einer höheren Gewichtung entsprechend höher gewichtet in die Berechnung eingehen. Der Erwartungswert für eine diskrete Zufallsvariable errechnet sich durch: [6]

$$E(X) = \sum_{x_i \in I} x_i * P(X = x_i) \tag{3.19}$$

Dabei ist I eine abzählbare Indexmenge, welche die Elemente x_i mit den jeweiligen Wahrscheinlichkeiten p_i enthält. Aus dieser Vorschrift ergibt sich die auf unseren Anwendungsfall angepasste Formel zur Berechnung des Erwartungswertes der Position:

$$\vec{Position} = \sum_{(\vec{pos},w) \in Particles} w * \vec{pos} \tag{3.20}$$

3.3.5 Voraussage der wahrscheinlichen nächsten Position

Nach jedem Resampling-Schritt zum Zeitpunkt t muss der Zustand der Partikel zum Zeitpunkt $t+1$ bestimmt werden. Dies ist für die Genauigkeit des Positionsbestimmung entscheidend.
Der zukünftige Zustand wird dabei aus der wahrscheinlichen weiteren Bewegungsrichung und Bewegungsweite des Empfängers bestimmt. Diese werden aus dem bisherigen Bewegungsverlauf berechnet.
Der wahrscheinliche Bewegungsvektor $\vec{v}$ wird dabei folgendermaßen berechnet:

$$\vec{v_{t+1}} = \sum_{t-n}^{t} \vec{v_i} * w_i \tag{3.21}$$

Wobei

$$\vec{v_t} = \vec{pos_t} - \vec{pos_{t-1}} \tag{3.22}$$

Alle Partikel des Monte-Carlo-Filters müssen nach einem erfolgreichen Resampling-Schritt und der darauf folgenden Positionsberechnung, um diesen Bewegungsvektor verschoben werden. Mit diesem Schritt wird der nächste Iterationszyklus vorbereitet und so eine möglichst optimale Positionsberechnung ermöglicht.
Gleichzeitig wird mit der Verschiebung auch die Diversität der Hypothesen erhöht, um eine bessere Abdeckung des Lösungsraumes zu erreichen. Dazu wird der Bewegungsvektor $\vec{v}$ für jeden Partikel bezüglich einer normalverteilten Zufallsvariable modifiziert. Die Standardabweichung wird dabei aus der Varianz $Variance_{\vec{Position}}$ der letzten Position bestimmt.

$$Variance_{\vec{Position}} = \frac{1}{|ParticleFilter| + 1} * \sum_{p \in ParticleFilter} (\vec{p} - \vec{Position})^2 \tag{3.23}$$

[6] www.mia.uni-saarland.de/Teaching/MFI0708/kap67.pdf

Dieser Vorgang wirkt auch einer zu schnellen Konvergenz der Partikelwolke in einen einzigen Punkt entgegen.
Nach diesem Schritt werden neue Informationen der Sensoren dazu verwendet, alle zuvor beschriebenen Schritte (bis auf die Initialisierung) erneut auszuführen und so die Hypothesen zu prüfen und an die neuen Gegebenheiten anzupassen.

3.4 Installationskonzept der BLE-Beacons

Wie bereits in Abschnitt 3.2.3 beschrieben, werden die im Raum installierten Bluetooth-Low-Energy-Beacons sowohl zur Initialisierung des Partikelfilters (Abschnitt 3.2.3) als auch zur Korrektur der Position (Abschnitt 3.2.3), welche vorrangig durch PDR berechnet wird und somit stark fehleranfällig ist, genutzt.
Für die Initialisierung des Systems wird angenommen, dass der Nutzer sich in einem der Ein-/Ausgänge des Raumes befindet. Mit diesem Vorwissen wird an jedem Ein- und Ausgang ein BLE-Beacon installiert. Dadurch kann eine sehr geringe reale Distanz zwischen Sender und Empfänger von möglichst unter einem Meter erreicht werden. Diese geringe Entfernung ermöglicht eine flächenmäßig sehr stark eingeschränkte Umgebung, in der sich der Nutzer befinden kann. Die initial aufgestellten Hypothesen über die Position des Nutzers umfassen damit einen kleinen Lösungsraum, wodurch weniger Hypothesen aufgestellt und später geprüft werden müssen.

Alle weiteren BLE-Beacons werden hingegen sporadisch im übrigen Lösungsraum verteilt. Dabei sollte jedoch möglichst darauf geachtet werden, dass an jeder Position mindestens ein Beacon erreichbar ist. Die Entfernung zwischen Sender und Empfänger sollte dabei möglichst unter 15m - 20m sein, da BLE-Geräte nur eine geringe Reichweite besitzen und bei zu großen Entfernungen Störfaktoren wie Menschen oder statische Objekte wahrscheinlicher werden. Dadurch kann eine möglichst hohe Korrekturhäufigkeit der Position gewährleistet werden.

4. Implementierung

Nach dem Entwurf des Konzeptes und der Auswahl aller grundlegenden Verfahren und Algorithmen muss das vorgeschlagene Indoor-Positionierungssystem evaluiert werden. Für eine experimentelle Evaluierung muss das System prototypisch implementiert und daraufhin in Experimenten untersucht werden.

4.1 Anforderungen an den Prototypen

Der Prototyp muss den Anforderungen des Konzeptes und der Evaluierung genügen, um eine erfolgreiche und korrekte Überprüfung des vorgeschlagenen Indoor-Positionierungssystems zu gewährleisten. Diese sind in den folgenden Punkten zusammengefasst.

Anforderungen des Konzeptes

- Der Prototyp muss Bluetooth-Low-Energy-Geräte in seiner Umgebung suchen und registrieren können.
- Die Suche nach BLE-Geräten muss kontinuierlich ablaufen und darf nicht durch andere Prozesse gestoppt werden.
- Der Prototyp muss Sensorwerte erfassen können. Dazu gehören die 3-Achsen-Beschleunigung des Smartphones und die Werte des 3-Achsen-Magnetometers
- Die kontinuierliche Erfassung der Daten darf nicht durch eine andere Applikation oder das System pausiert oder unterbrochen werden, da sonst keine korrekten Schlussfolgerungen gezogen werden können.
- Der Prototyp muss eine geeignete Karte des betrachteten Raums/Gebäudes verarbeiten können. In dieser Karte müssen auch die bekannten Positionen der BLE-Beacons verzeichnet werden.

Anforderungen der Evaluierung

- Die durch Sensoren aufgenommenen Daten müssen für eine spätere Wiederholung der Berechnung gespeichert werden.
- Die Karte des Raums/Gebäudes muss Informationen zur Ground Truth besitzen. Diese ist eine Ansammlung von Positionen im Raum, die nacheinander angelaufen werden müssen.
- Die Auswertung der errechneten Positionen sollte im Prototypen erfolgen, um leicht nachvollziehbare und reproduzierbare Ergebnisse zu ermöglichen.

4.2 Wahl der Plattform

Für den Prototypen wurde die Smartphone-Plattform Android ausgewählt. Diese wird in den meisten Fällen zur Untersuchung neuer Verfahren, Algorithmen und Konzepte im Bereich der Smartphonenutzung gewählt, da sie sehr verbreitet und offen ist. Bei der Auswahl der Zielversion ist zu beachten, dass BLE erst mit Android 4.3 Einzug gehalten hat. Mit Android 5.0 wurde jedoch die BLE-Schnittstelle überarbeitet und an die Anforderungen angepasst. Bei der Implementierung ist die Android-Version 5.0 als Mindestanforderung festgelegt, da die überarbeiteten Schnittstellen genutzt werden. Dies ist für einen Prototypen ausreichend. In einer Implementierung für den Endanwender muss hingegen auch die Kompatibilität zu älteren Versionen beachtet werden.

4.3 Beschreibung des Implementierungskonzeptes

Der Prototyp ist nativ in Java programmiert und baut auf den androidspezifischen APIs auf. Eine Anpassung des Betriebssystems oder der Gerätetreiber ist nicht notwendig. Dies gewährleistet die Kompatibilität zu einem großen Teil der Android-Smartphones ohne weitere Änderungen.
Das vorgeschlagene Positionierungssystem wird in einem Service-Ansatz[1] implementiert. Dieses Vorgehen gewährleistet, dass das Betriebssystem einen laufenden Positionsbestimmungsprozess nicht ohne dringenden Grund beendet. Zudem wird für jeden Sensortyp ein eigener Thread erstellt, in dem die Datenaufnahme parallel zu sonstigen Berechnungen abläuft. Somit kann eine hohe Samplingrate trotz zeitaufwändiger Berechnungen gewährleiste werden. Ein Watchdog[2] überprüft zudem kontinuierlich die Samplingrate. Sollte diese einen Grenzwert unterschreiten, wird der Samplingprozess für den betroffenen Sensor neu gestartet. Vor allem die Bluetooth-Schnittstelle ist von diesem Problem betroffen[3].
Die Berechnung der Position erfolgt im beschriebenen Service bei jeder neuen Information. Eine Information kann dabei eine berechnete Distanz zu einem Beacon sein oder ein neuer PDR-Vektor, welcher nach einem erkannten Schritt berechnet wurde.

[1] http://developer.android.com/guide/components/services.html
[2] Ein Watchdog ist ein Prozess, welcher kontinuierlich die geforderten Bedingungen prüft und bei hohen Abweichungen einen Prozess neu initialisiert.
[3] https://github.com/iDevicesInc/SweetBlue/wiki/Android-BLE-Issues

Die neu bestimmte Position wird daraufhin an einer Serviceschnittstelle bereitgestellt und kann von anderen Applikationen verarbeitet werden.

Der Evaluierungsprozess nutzt an der Schnittstelle die bereitgestellten Positionsinformationen und gleicht diese mit einer in der Karte gespeicherten Ground Truth ab. Die Evaluierung wird dabei nur offline auf zuvor gespeicherten Sensordaten ausgeführt. Im Datensammelprozess muss der Nutzer festgelegte Punkte im Raum anlaufen. Sobald die Position erreicht wurde, wird ein Zeitstempel gespeichert. Während des gesamten Vorgangs werden die Sensordaten mit dazugehörendem Zeitstempel gespeichert. Bei der Evaluierung wird die berechnete Position zum Zeitpunkt t_i mit der in der Ground Truth gespeicherten Position zum Zeitpunkt t_i verglichen.

4.4 Repräsentation der Daten

Kartenrepräsentation

Der Prototyp muss eine Karte des Raums/Gebäudes interpretieren können, da diese der Orientierung des Nutzers beim Sammeln von Bewegungsdaten und der Berechnung aller Distanzen nützt. Zudem müssen auch die Positionen der Beacons und die Stationen, welche während einer Datenaufnahme zur Evaluierung angelaufen werden müssen, in der Karte enthalten sein.
Dazu wird die Raumstruktur in einer Liste von Eckpunkten gespeichert. Zusätzlich werden zur Orientierung Fenster und Türen ebenfalls als Eckpunkte eingetragen. Beacons hingegen werden als Positionstupel mit zusätzlicher UUID und Referenzsendeleistung gespeichert.
Die Ground-Truth-Stationen sind als Liste von Positionen in der Karte angegeben.

Sensordaten für spätere Evaluierung

Für eine spätere Evaluierung müssen die Sensordaten mit ihrem Zeitstempel gespeichert werden. Hierzu werden die Datentupel in eine Datei auf dem Gerätespeicher geschrieben und können daraufhin immer wieder genutzt werden.
Für BLE wird das Tupel $(timestamp, UUID, RSSI, MAC)$ gespeichert. Beschleunigungsdaten werden mit dem Tupel $(timestamp, x, y, z)$ gesichert. Magnetometerdaten haben die gleiche Repräsentation wie Beschleunigungsdaten.
Für eine spätere Evaluierung werden zudem Zeitstempel für jede Station der Ground Truth gespeichert, an denen der Nutzer sich an der Station befunden hat. Dazu muss das Tupel $(StationID, timestamp)$ gesichert werden.

4.5 Entwurf der UI

Um eine problemlose Evaluierung des Systems vornehmen zu können, muss die Benutzeroberfläche des Prototypen derart gestaltet sein, dass keine Fehler während der Datenaufnahme und Systemevaluierung durch falsche Benutzung entstehen. Ein Dialog-Assistent hilft dabei, nur sinnvoll aufeinander folgende Aktionen auszuführen. Am Start einer App und zu bestimmten Zeitpunkten wird das Konfigurationsinterface gezeigt, welches durch die Auswahl und Benutzerdialoge weitere Aktionen bereitstellt und passende Interfaces darstellt.

4.5.1 Konfigurationsinterface

Bei jedem Start der Applikation muss diese konfiguriert werden. Dabei werden der untersuchte Raum/das untersuchte Gebäude ausgewählt. Daraufhin kann der gewünschte Modus gewählt werden. Der Modus entscheidet, ob im weiteren Vorgehen Sensordaten aufgezeichnet werden oder ob bereits aufgezeichnete Daten zur Evaluierung des Systems genutzt werden sollen.
Dieses Konfigurationsinterface (siehe Abbildung 4.1) kann jederzeit aufgerufen werden, wenn keine Aufgabe ausgeführt wird.

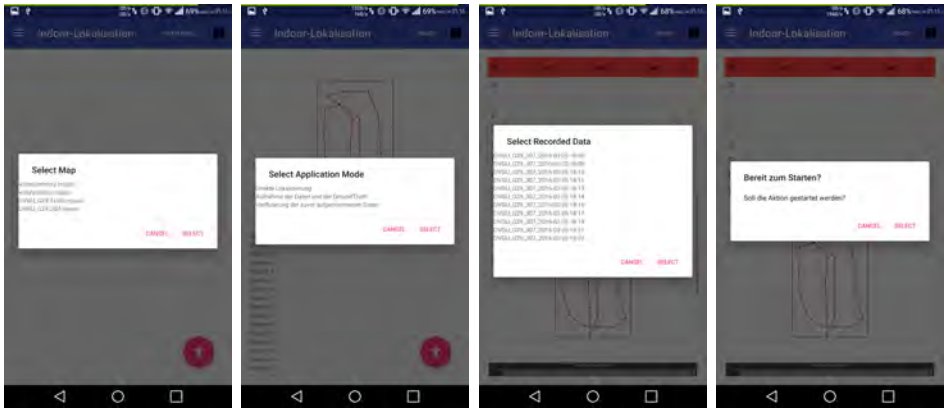


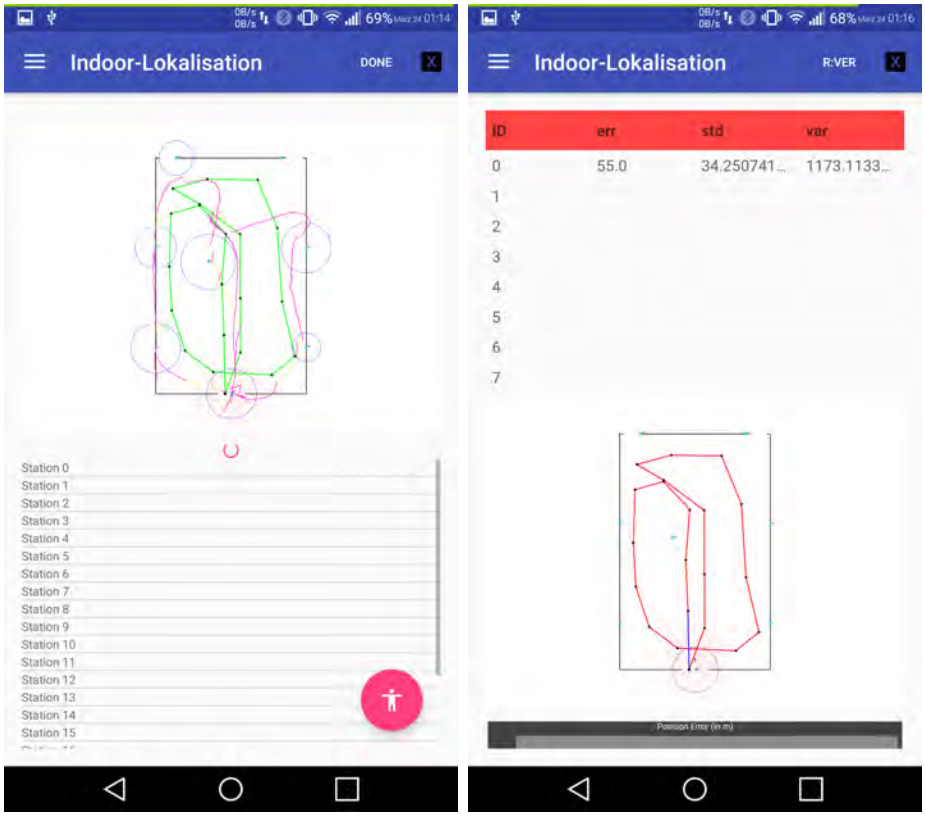
Abbildung 4.1: Konfigurationsdialoge für die Auswahl der betrachteten Räume, des Modus und ggf. der vorhandenen aufgezeichneten Daten

4.5.2 Interface zur Datenaufnahme

Für die Datenaufnahme muss der Nutzer bestimmte Stationen in einem Raum nacheinander ablaufen. Dabei werden alle benötigten Sensorwerte gespeichert.
Um die Orientierung des Nutzers zu vereinfachen, wird die Karte des untersuchten Raumes angezeigt. Zusätzlich werden die festgelegten Stationen in der Karte dargestellt. Die nächste Zielstation ist dabei besonders hervorgehoben.
Sobald ein Nutzer eine Station erreicht hat, muss dieser nur einen Knopf in der App betätigen, damit der Zeitstempel des Besuches an der Station gespeichert wird. Die Datensammeloberfläche ist in 4.2(a) dargestellt.
Zusätzlich sind Bedienelemente zum Neustart und Abbruch der Datenaufnahme verfügbar. Sobald die letzte Station mit einem Zeitstempel versehen wurde, endet die Datenaufnahme. Dies wird dem Nutzer durch eine Statusmitteilung signalisiert.

4.5.3 Interface zur Evaluierung

Das Evaluierungsinterface stellt, wie bei der Datenaufnahme, eine Karte dar, jedoch sind in dieser zusätzlich zu den Stationen auch die kontinuierlich berechnete Position und der bisher zurückgelegte Weg eingezeichnet.
Die errechneten Varianzen während der Positionsberechnung und die Fehlerwerte an jeder Station werden in einem Diagramm und einer Tabelle (siehe 4.2(b)) dargestellt.
Dieser Prozess kann ebenfalls durch Bedienelemente neu gestartet oder abgebrochen werden. Die berechneten Daten müssen nicht manuell exportiert werden, da sie automatisch gespeichert werden.



(a) (b)

Abbildung 4.2: (a) Interface für die Datenaufnahme während eines Versuches. Der magentafarbene Knopf wird betätigt, sobald die oberste Station der Liste erreicht wurde (b) Interface während der laufenden Berechnungen für die Evaluierung

5. Evaluierung

Das vorgeschlagene Konzept für ein Indoor-Positionssystem auf Basis von Bluetooth-Low-Energy-Geräten und Pedestrian Dead Reckoning muss mit Hilfe des implementierten Prototypen evaluiert werden, um dessen Funktionsfähigkeit zu zeigen und die Genauigkeit zu ermitteln.
Dazu wurden mehrere verschieden große Räume mit Bluetooth-Low-Energy-Sendern ausgestattet. Der Raumplan, die Positionen der Beacons und der festgelegte Weg wurden in einer digitale Karte festgehalten und dem Prototypen zur Verfügung gestellt.
Der Nutzer hat während aller Versuche, durch genaues Ablaufen des vorgegebenen Weges, Sensordaten gesammelt, aus denen die jeweilige Position berechnet wurde. Diese wurden mit den Positionen des vorgegeben Weges (Ground Truth) verglichen.

5.1 Verwendete Hardware

Für die Evaluierung wird ein handelsübliches Smartphones des Herstellers LG genutzt. Die Software wird auf einem LG G3 mit Android 6.0.1 evaluiert. Dieses Smartphones besitzt die benötigten inertialen Sensoren und die Fähigkeit, BLE-Signale kontinuierlich zu erfassen.
Als BLE-Beacons werden Beacons der Firma Beaconinside[1] in der Hardwarerevision 2.0 verwendet. Diese können verschiedene BLE-Protokolle ausstrahlen, werden aber derart konfiguriert, dass nur Advertisement-Pakete im iBeacon-Protokoll mit einer Leistung von +0dBm in einem 100ms-Interval ausgesendet werden.

5.2 Verfahren der Evaluierung

Die Evaluierung wird in allen ausgewählten und vorbereiteten Räumen durchgeführt. Es werden mindestens 9 Versuche pro Raum durch dieselbe Person durchgeführt. Die so gesammelten Daten werden im Nachhinein ausgewertet und alle benötigten Kenngrößen berechnet.

[1] http://www.beaconinside.com/beacon/

5.2.1 Auswahl der Räume

Für die Untersuchungen wurden drei verschiedene Raumtypen definiert und Versuchsräume für diese ausgewählt. Die Raumtypen wurde dabei auf Grundlage der häufig in öffentlichen Gebäuden vorkommenden Raumtypen ausgewählt und werden im Weiteren beschrieben:

Raumtyp 1 - Quadratisch mit Raumfläche unter $20\,\mathrm{m}^2$

Der erste Raumtyp ist flächenmäßig relativ klein und schränkt die Bewegungsfreiheit ein. Es sind nur wenige Schritte in eine Richtung möglich, wodurch vor allem eine große Vorlaufzeit für die Schritterkennung nicht gegeben ist. Diese muss sofort zuverlässig Schritte erkennen, welche für eine verlässliche Positionsbestimmung benötigt werden.
Vor allem in kulturellen Einrichtungen werden selbst kleinste Räume genutzt. In diesen sind häufig besondere oder kleine Exponate zu finden, die ein bestimmtes Thema aufgreifen. Auh in solchen kleinen Räume sollte eine gute Positionsbestimmung gewährleistet werden.
Ein flächenmäßig so kleiner Raum ist stark für das Multi-Path-Problem anfällig, wodurch die RSSI-basierte Positionierung gestört werden kann.
Als Beispielraum wurde ein $17{,}05\,\mathrm{m}^2$ großer, nahezu quadratischer Raum in einer Wohnung gewählt. Dieser wurde mit fünf Bluetooth-Low-Energy-Beacons ausgestattet und es wurden sieben Referenzpositionen festgelegt.

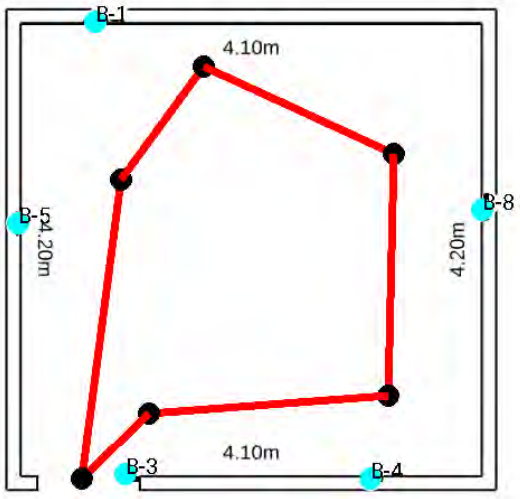


Abbildung 5.1: Raumplan des Versuchsraumes 1 mit installierten Beacons (türkis) und dem Versuchspfad (rote Geraden) zwischen den festgelegten Stationen (schwarze Punkte)

Raumtyp 2 - Langgestreckt mit Raumfläche zwischen $20\,\mathrm{m}^2$ **und** $100\,\mathrm{m}^2$

Als zweite Raumtyp sollte ein sehr langer, aber schmaler Raum untersucht werden, welcher oft als Verbindungsraum zwischen verschiedenen anderen Räumen eines Gebäudes vorkommt. Vor allem Bürokomplexe, große Wohngebäude und alte Schlossanlagen besitzen solche Räume als Flur zwischen Wohnungen, Büros oder anderen

Räumen.
Dieser Raumtyp stellt eine große Herausforderung für das inertiale und radiobasierte Positionierungssystem dar, weil die Abstände des Nutzers zu einzelnen Wänden und Türanlagen sehr gering sind. Wände bzw. die darin verbauten elektrischen Anlagen und elektrische Türschließanlagen erzeugen ein starkes elektromagnetisches Feld, welches das Erdmagnetfeld überlagert. Diese Überlagerung stört die Richtungsbestimmung des inertialen Systems, wodurch Positionsfehler entstehen.
Des Weiteren werden durch die nur wenige Meter auseinanderliegenden Wände des Raumes die BLE-Signale stark reflektiert. Die daraus entstehenden Multi-Path-Effekte können zu starken Fehlern in der Positionsbestimmung führen.
Die Versuche für diesen Raumtyp wurden in einem Flur der Fakultät für Informatik der Otto-von-Guericke Universität Magdeburg durchgeführt. Dieser Raum hat eine Fläche von 59,95 m^2 bei einer Breite von gerade einmal 1,88 m. Zudem besitzt dieser starke elektromagnetische Störfelder, welche durch Stromkästen, Leitungen und einen Serverraum entstehen. Diese Störfelder beeinträchtigen das PDR-System stark und müssen durch die BLE-basierte Positionsbestimmung kompensiert werden.
Für den Versuch wurden acht BLE-Beacons in ähnlichen Abständen zueinander installiert. Zudem wurden dreizehn Referenzpositionen festgelegt. Sowohl die Positionen der Beacons als auch die Referenzpositionen wurden in die digitale Karte des Raumes eingetragen.

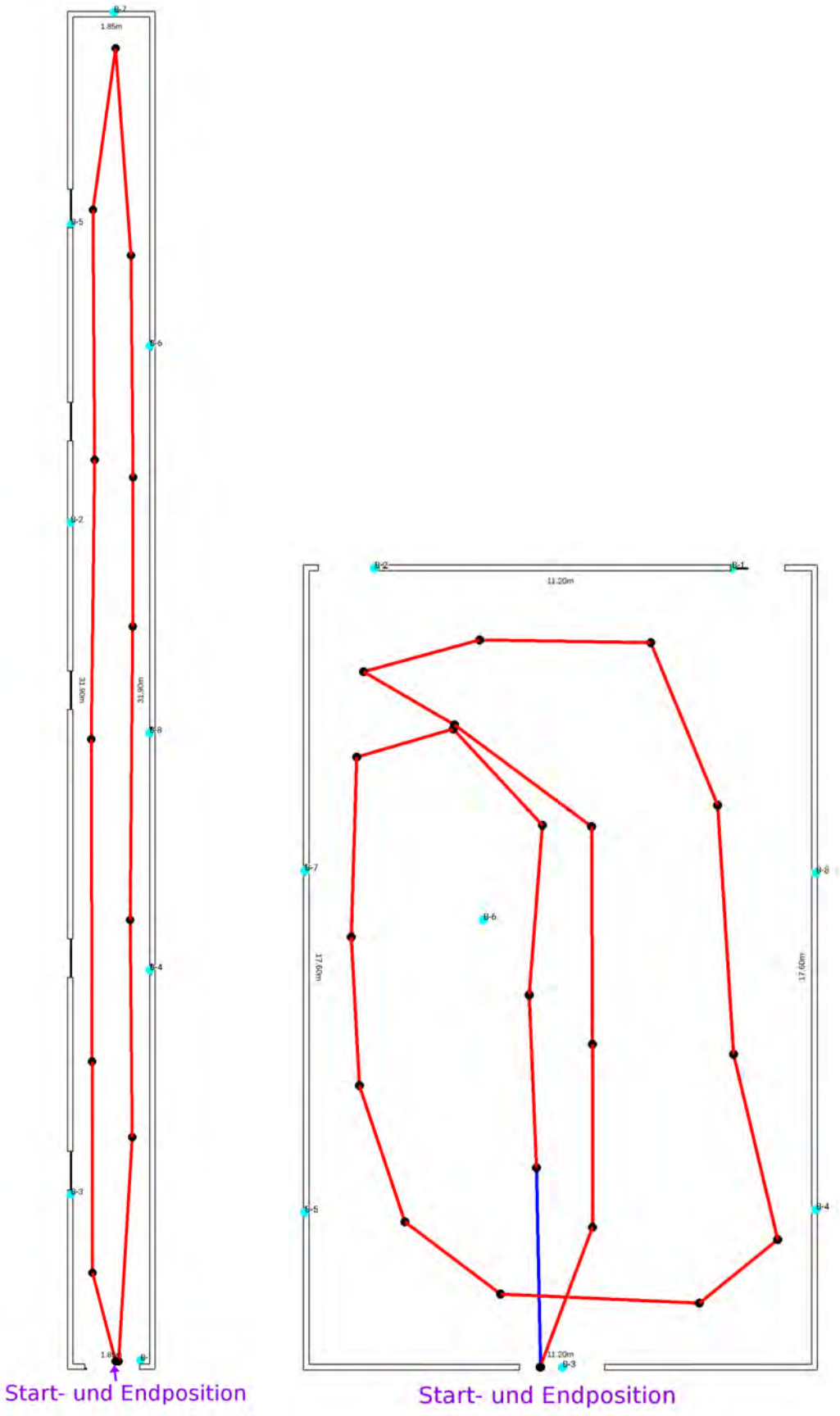
Raumtyp 3 - Raumfläche über 100 m^2

Der dritte Raumtyp ist vor allem in öffentlichen Gebäuden zu finden und ist flächenmäßig sehr groß. Diese Räume werden oft als Ausstellungs- oder Veranstaltungsräume genutzt und sind daher für die Positionsbestimmung und darauf aufbauende Navigation relevante Einsatzgebiete.
Bei diesem Raumtyp sind vor allem Bereiche in der Nähe der Wände und Eingangsbereiche starken Störquellen wie elektromagnetischen Feldern ausgesetzt. Diese Störungen muss das RSSI-basierte Positionsbestimmungssystem korrigieren, um so eine zuverlässige Positionsbestimmung zu ermöglichen.
Als Versuchsraum wurde der Hörsaal der Fakultät für Informatik (G29-307) der Otto-von-Guericke Universität Magdeburg gewählt. Dieser Raum ist mit 197,43 m^2 Fläche und drei Eingängen geeignet, um das vorgeschlagene System ausgiebig zu testen. Für die Versuche wurden acht BLE-Beacons installiert. Dabei wurden drei der Beacons jeweils an einem Ausgang positioniert. Ein Beacon wurde in der Nähe der Raummitte installiert, die restlichen vier hingegen fanden an den zwei langen Wänden des Raumes Platz. Zudem wurden 23 Referenzpositionen bestimmt, welche während eines Versuches nacheinander angelaufen werden mussten. Die Gesamtstrecke eines Versuchs liegt bei mindestens 74,76 m.

5.2.2 Vorbereitung der Räume

Alle Räume wurden durch ein Laser-Distanz-Messgerät[2] zentimetergenau ausgemessen und aus den Messwerten eine digitale Karte erstellt. Zudem wurden die Bluetooth-Low-Energy-Beacons installiert und die Stationen des untersuchten Pfades auf dem Boden markiert. Sowohl die Stationen als auch die Beacons wurden ebenfalls in der Karte mit ihren genauen Positionen eingetragen.

[2]Suaoki S9 mit einer maximalen Reichweite von 60 m und einer Genauigkeit von 1,5 mm



(a) Raum 2 (b) Raum 3

Abbildung 5.2: Raumpläne der untersuchen Versuchsräume 2 und 3 mit installierten Beacons (türkis) und dem Versuchspfad (rote Geraden) über die festgelegten Ground-Truth-Stationen (schwarze Punkte im Versuchspfad)

5.2.3 Durchführung der Datenaufzeichnung

Die Aufzeichnung der Sensordaten erfolgt für jeden Raum separat. Diese Daten werden im späteren Verlauf dazu genutzt, den eigentlichen Algorithmus auszuführen und so die Position zu bestimmen.
Für die Datenaufzeichnung positioniert sich der Nutzer im Eingangsbereich des Raumes und startet die Datenaufzeichnung in der App. Dann bewegt sich der Nutzer im normalen Schritttempo (etwa 2 Schritte pro Sekunde) zur nächsten eingezeichneten Station. Sobald der Nutzer die Station erreicht hat, wird dies durch einen Knopfdruck signalisiert. Der Zeitpunkt wird gespeichert. Der Nutzer bewegt sich daraufhin zur nächsten angegebenen Station und signalisiert dort ebenfalls sein Ankommen durch einen Knopfdruck.
Dieses Verfahren wird für die restlichen Stationen durchgeführt. Sobald die letzte Station bestätigt wurde, endet die Sensordatenaufnahme. Die gesammelten Rohdaten werden auf dem persistenten Speicher des Smartphones gespeichert und so für weitere Verwendung gesichert.

5.2.4 Berechnung der Kenngrößen

Zur Evaluierung des vorgeschlagenen und implementierten Systems sollen folgende Kenngrößen für jeden Versuch berechnet werden:

Abweichung in m für jede Station i eines Versuches j

Diese Abweichung soll für jede Referenzposition eines Raumes für alle Versuche berechnet werden. Die berechneten Werte finden sich in den Anlagen dieser Arbeit und werden für die Berechnung folgender Kenngrößen benötigt.
Die Abweichung ist dabei der euklidische Abstand zwischen der Referenzposition und der berechneten Position zum selben Zeitpunkt.

$$Abweichung_{j,i} = |\vec{Position}_{Station_i} - \vec{Position}_{Berechnet_{j,i}}| \tag{5.1}$$

Unsicherheit der berechneten Position zur Station i eines Versuches j

Die Unsicherheit einer berechneten Position wird für jede Station i eines Versuches bestimmt. Dazu wird die Varianz der Hypothesen $Var_{Station=i,Versuch=j}$ des genutzten Partikelfilters berechnet und aus dieser die Standardabweichung bestimmt. Diese wiederum gibt die Unsicherheit der berechneten Position an.

$$\sigma_{Station=i,Versuch=j} = \sqrt{Var_{Station=i,Versuch=j}} \tag{5.2}$$

$$Var_{Station=i,Versuch=j} = \frac{1}{|PF|+1} \sum_{\vec{p} \in PF} (\vec{p} - \vec{Position}_{Station=i,Versuch=j})^2 \tag{5.3}$$

Die Varianz muss vor dem Prediction-Schritt(Abschnitt 3.3.5) des Partikelfilters berechnet werden, damit das in diesem Schritt eingebrachte Rauschen die Werte nicht beeinflusst.

Durchschnittlicher und maximaler Fehler eines Versuches j

Für jeden Versuch sollen der durchschnittliche Fehler $\mu_{Fehler,Versuch=j}$ und der maximale Fehler $max_{Fehler,Versuch=j}$ berechnet werden. Diese ergeben sich aus den einzelnen Abweichungen für jede festgelegte Station.

$$max_{Fehler,Versuch=j} = \max(Abweichung_{j,1}, .., Abweichung_{j,n}) \tag{5.4}$$

$$\mu_{Fehler,Versuch=j} = \frac{\sum_{i=1}^{n} Abweichung_{j,i}}{n} \tag{5.5}$$

Durchschnittliche und maximale Unsicherheit eines Versuches j

Parallel zu der Berechnung der maximalen und durchschnittlichen Fehler eines Versuches werden auch die maximalen und durchschnittlichen Unsicherheiten berechnet. Diese ergeben sich aus den Werten der Stationen eines Versuches.

$$max_{\sigma,Versuch=j} = \max(\sigma_{Station=1,Versuch=j}, .., \sigma_{Station=n,Versuch=j}) \tag{5.6}$$

$$\mu_{\sigma,Versuch=j} = \frac{\sum_{i=1}^{n} \sigma_{Station=i,Versuch=j}}{n} \tag{5.7}$$

Durchschnittlicher und maximaler Fehler eines Raumtyps

Damit eine generelle Aussage über die Eignung und Güte des vorgeschlagenen Systems für einen Raum getroffen werden kann, müssen die Werte jedes Versuchs k für einen Raumtyp zusammengefasst werden. Dazu werden die durchschnittlichen und die maximalen Fehler der gemittelten und höchsten Fehler eines Versuches berechnet.

$$max_{Fehler,Durchschnitt} = \max(\mu_{Fehler,Versuch=1}, .., \mu_{Fehler,Versuch=n}) \tag{5.8}$$

$$max_{Fehler,Maximum} = \max(max_{Fehler,Versuch=1}, .., max_{Fehler,Versuch=n}) \tag{5.9}$$

$$\mu_{Fehler,Durchschnitt} = \frac{\sum_{i=1}^{k} \mu_{Fehler,Versuch=j}}{k} \tag{5.10}$$

$$\mu_{Fehler,Maximum} = \frac{\sum_{i=1}^{k} max_{Fehler,Versuch=j}}{k} \tag{5.11}$$

Durchschnittliche und maximale Unsicherheit eines Raumtyps

Damit weitere Aussagen über die Eignung und Güte des vorgeschlagenen Systems für einen Raum getroffen werden kann, müssen zudem die Standardabweichungen jedes Versuchs k für einen Raumtyp zusammengefasst werden. Dazu werden die durchschnittlichen und maximalen Standardabweichungen der gemittelten und höchsten Unsicherheiten eines Versuches berechnet.

$$max_{\sigma,Durchschnitt} = \max(\mu_{\sigma,Versuch=1}, .., \mu_{\sigma,Versuch=n}) \tag{5.12}$$

$$max_{\sigma,Maximum} = \max(max_{\sigma,Versuch=1}, .., max_{\sigma,Versuch=n}) \tag{5.13}$$

$$\mu_{\sigma,Durchschnitt} = \frac{\sum_{i=1}^{k} \mu_{\sigma,Versuch=j}}{k} \tag{5.14}$$

$$\mu_{\sigma,Maximum} = \frac{\sum_{i=1}^{k} max_{\sigma,Versuch=j}}{k} \tag{5.15}$$

5.2.5 Gewählte Parameter für die Algorithmen

Im Folgenden werden alle wichtigen Parameter der genutzten Algorithmen und Filter beschrieben.

Filter der Sensordaten

Für die Filterung der Beschleunigungsdaten werden, wie bereits in Abschnitt 3.1.1 beschrieben, sowohl ein Median- als auch ein Mittelwertfilter genutzt. Der Medianfilter nutzt die letzten sieben Datenpunkte um einen Median zu berechnen und so die Daten von stark abweichenden Werten zu bereinigen. Darauf aufbauend wird ein Mittelwert der letzten 15 Datenpunkte erstellt, welche zuvor vom Medianfilter vorverarbeitet wurden. Der Mittelwertfilter wird vor allem zur Glättung der Daten genutzt, welche trotz Medianfilter lokale unerwünschte Extrempunkte aufweisen können. Des Weiteren wird ein High-Pass-Filter zur Extraktion der Gravitation aus den rohen Beschleunigungsdaten genutzt(vgl. Abschnitt 2.2.1). Die genutzte Filterkonstante wurde empirisch ermittelt und mit dem Wert 0.8 festgelegt. Das bedeutet, dass 20 Prozent eines Datenpunktes in die Änderung der berechneten Gravitation eingeht. Die verbleibenden 80 Prozent werden hingegen aus vorhergehenden Schritten übernommen.
Die Magnetometerdaten werden nicht gefiltert und fließen mit dem berechneten Gravitationsvektor in die Berechnung der Geräteausrichtung ein.
Die für eine Distanzberechnung benötigten RSSI-Werte werden ebenfalls durch einen Mittelwertfilter vom Rauschen bereinigt. Dazu wird aus den letzten drei Werten immer ein Mittelwert gebildet, welcher in die weitere Berechnung einfließt. Der geringe Wert wurde durch die Beobachtung gewählt, dass die Entfernung eines Nutzers zum Sender schnellen und starken Änderungen unterliegt, wenn dieser sich besonders nah am Sender bewegt. Die gleichzeitig geringe Samplingrate von maximal 10 Paketen pro Sekunde bedeutet jedoch bereits bei 3 RSSI-Werten eine Verzögerung von 150 ms.

Pedestrian Dead Reckoning

Für die PDR-basierte Positionsbestimmung mussten Parameter der Schritterkennung und Schrittlängendetektion definiert werden. Zudem wurden Mittelwertfilter für die berechneten Schrittlängen und Ausrichtungswinkel genutzt, welche starke Schwankungen der Werte unterdrücken sollten.
Für die Schritterkennung wurden die Konstanten $\alpha = 2$ und $\beta = 1/3$ gewählt (vgl. Abschnitt 2.2.4.1). Zusätzlich wurde eine feste Schranke definiert, welche für die Differenz der aufeinanderfolgenden Maxima und Minima gilt. Diese legt eine Untergrenze für die Differenz fest und verhindert dadurch die fälschliche Detektion von Schritten in leichten Schwankungen der Beschleunigungsmagnitude. Die Grenze von 0,2 g wurde dabei empirisch ermittelt.
Über die geschätzten Schrittlängen der letzten sieben detektierten Schritte wird bei jedem neuen Schritt ein arithmetisches Mittel gebildet, um Schwankungen der Schrittlängen durch unterschiedliche Auftrittintesitäten auszugleichen. Dieser Wert wurde ebenfalls empirisch ermittelt und kann so hoch gewählt werden, da die Schritte während eines kontinuierlichen Ganges normalerweise nicht stark variieren.
Die kurzzeitigen Ungenauigkeit des berechneten Ausrichtungswinkels wird durch einen drei Werte umfassenden Mittelwertfilter entfernt. Die geringe Anzahl der genutzten Werte wurde durch Beobachtungen im Raumtyp 2 und 3 festgelegt, da in diesen jeweils eine schnelle Richtungsänderung der Bewegung stattgefunden hat, welche in den gefilterten Daten sehr zeitnah wiedergefunden werden musste. Eine höhere Anzahl der genutzten Winkel wirkte sich negativ auf die berechnete Richtung und Position kurz nach der realen Richtungsänderung aus.

RSSI-basierte Distanzbestimmung

Für die RSSI-basierte Distanzbestimmung zwischen Nutzer und Sender werden die letzten RSSI-Werte der jeweiligen Sender durch einen Mittelwertfilter vom starken Rauschen bereinigt. Es konnte jedoch in den Versuchen beobachtet werden, dass nicht regelmäßig Pakete eines Senders empfangen werden. Teilweise wurden gesendete Advertisement-Pakete nicht registriert. Durch diese Effekte kann eine kontinuierliche Filterung der Daten nicht erfolgen. Im ungünstigen Fall können einige der zur Filterung genutzten Werte mehrere Sekunden alt sein und somit den aktuellen Abstand nicht mehr repräsentieren. Alle Pakete, welche über eine Sekunde nicht in eine Berechnung eingeflossen sind, werden entfernt und fließen nicht in die Distanzbestimmung ein. So wird ein zu hoher Distanzfehler durch veraltete Daten vermieden.
Die Dämpfungskonstante n (siehe Abschnitt 2.1.5) wurde durch Versuche der Distanzberechnung bei bekannter Entfernung bestimmt. Diese konnte auf den Wert $n = 2,5$ festgelegt werden.

Partikelfilter

Im Partikelfilter wurden konstant 2500 Partikel zur Berechnung der Position und Fusion der Informationen genutzt. Dieser Wert erwies sich als guter Kompromiss zwischen benötigter Rechenleistung und der Abdeckung des lokalen Lösungsraumes. Im Prediction-Schritt(vgl. Abschnitt 3.3.5) fliest der zuvor berechnete Bewegungsvektor der letzten Positionen zur Hälfte ein. Dies lieferte gute Ergebnisse für die

Voraussage der nächsten Position bei langsamen und schnellen Änderungen der Bewegungsrichtung.

5.3 Kenngrößen und Ergebnisse

Für jeden der hier beschriebenen (Abschnitt 5.2.1) drei Raumtypen wurden mindestens neun Versuche durchgeführt, bei denen Daten gesammelt, ausgewertet und die dazugehörenden Kenngrößen berechnet wurden. Die Kenngrößen und Ergebnisse werden im Weiteren für jeden Raumtyp beschrieben und eingeschätzt. Zusätzlich werden in Abschnitt 5.3.4 allgemeine Ergebnisse dargestellt, welche bei der Durchführung aller Versuche beobachtet wurden. Darauf folgend wird eine allgemeine Einschätzung des vorgestellten Systems gegeben und mögliche Fehlerursachen zusammengefasst.

5.3.1 Raumtyp 1 - Fläche unter $20\,\mathrm{m}^2$

Tabelle 5.1: Kenngrößen für Raum 1 - Raumfläche $< 20\,\mathrm{m}^2$

Versuch #	Mittelwert μ_{error}	Mittelwert σ_u	$\max_{error}$	$\max_{\sigma_u}$
1	103.82 cm	13.6611 cm	214.22 cm	18.8031 cm
2	77.62 cm	14.0077 cm	174.47 cm	21.3158 cm
3	79.01 cm	14.2991 cm	242.00 cm	18.4430 cm
4	114.82 cm	13.4549 cm	214.45 cm	21.6986 cm
5	74.65 cm	14.5757 cm	173.51 cm	19.7149 cm
6	146.11 cm	14.9594 cm	255.41 cm	22.0552 cm
7	115.01 cm	14.5958 cm	228.74 cm	22.6079 cm
8	155.75 cm	13.2355 cm	360.17 cm	20.6954 cm
9	59.68 cm	14.1804 cm	88.14 cm	20.3777 cm
Mittelwert	102.94 cm	14.1077 cm	216.79 cm	20.6346 cm
Max	155.75 cm	14.9594 cm	360.17 cm	22.6079 cm

Die in Tabelle 5.1 dargestellten Ergebnisse des ersten Raumtyps sind zufriedenstellend und zeigen, dass der durchschnittliche Fehler der berechneten Position bei 1,03 m liegt. Dieser Wert wird jedoch nicht dauerhaft erreicht. So hat sich gezeigt, dass der größte Mittelwert eines Versuches bereits bei 1,56 m und damit etwa ein Drittel über dem Raumdurchschnitt liegt.
Diese Schwankungen zwischen verschiedenen Versuchen können an leicht unterschiedlichen Laufgeschwindigkeiten und Auftrittsstärken liegen. Diese beeinflussen die Schrittlängenschätzung, welche wiederum die berechnete Position beeinflusst. Ein weiterer, und weit wichtigerer Faktor, sind jedoch die Ungenauigkeiten bei der Signalstärkemessung. Im Versuchsraum entstehen, bedingt durch die geringe Größe, starke Reflexionen der gesendeten Signale, welche in einigen Versuchen häufiger auftraten und damit stärkere Einflüsse haben als in anderen. Die berechneten Distanzen zu den einzelnen Sendern wurden dabei immer wieder überschätzt und eine falsche Korrektur angewendet. In den Versuchen 9 und 3 wurden hingegen an manchen Positionen keine Korrekturen durchgeführt, obwohl ein BLE-Sender in unmittelbarer Nähe war. Dies ist auf ungünstiges Zusammenwirken von stark gestörten Signalen,

zu wenigen empfangenen Paketen und einer hohen Laufgeschwindigkeit von etwa 1,5 m/s zurück zu führen.
Die in Abschnitt 5.3.4 beschriebene Verzögerung der Signalauswertung durch Filter und damit verbundene verfälschte Korrektur der Position trägt ebenfalls zu den entstehenden Fehlern bei. Das Zusammenwirken der verschiedenen Faktoren führt zu den relativ hohen maximalen Fehlern von bis zu 3,60 m.

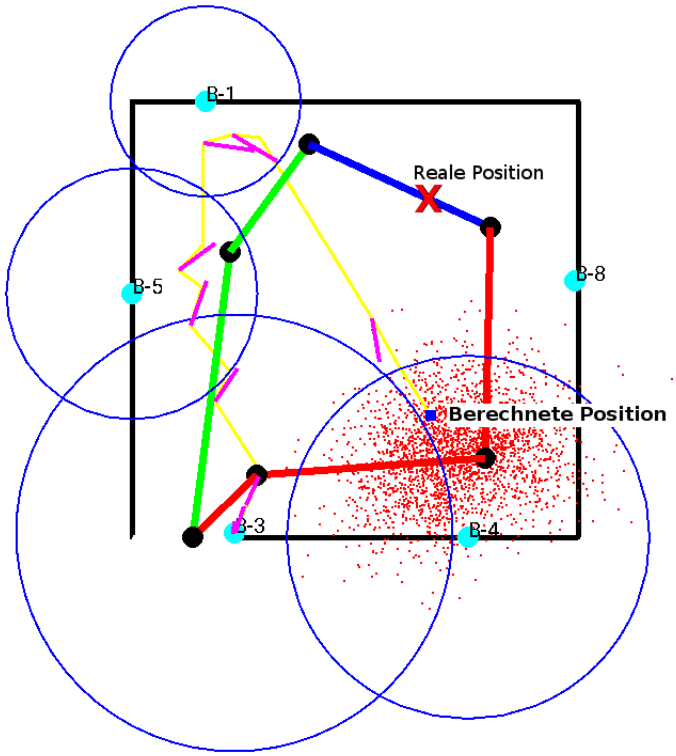


Abbildung 5.3: Fehlende Korrektur durch Beacon 8 (B-8) durch Multi-Path-Effekte. Stattdessen erfolgt eine fehlerhafte Korrektur durch Beacon 4 (B-4), wodurch eine hohe Abweichung zwischen der realen und berechneten Position entsteht. Die falsche Korrektur kommt durch die hohe Rauschanfälligkeiten der RSSI-Messwerte zustande.

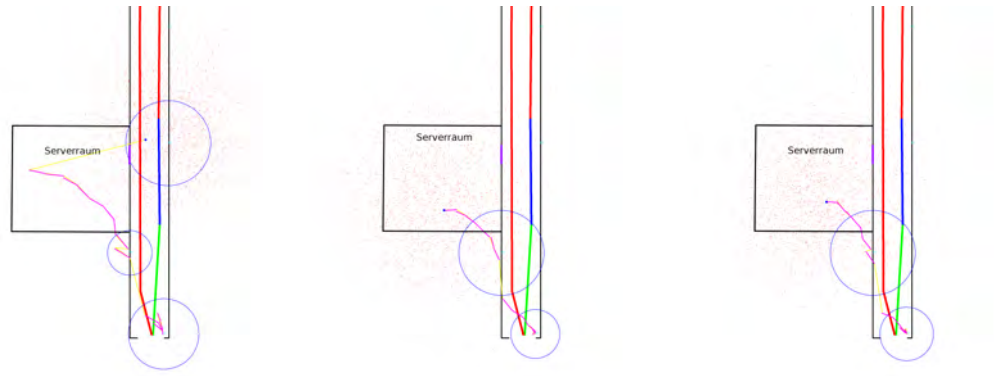
5.3.2 Raumtyp 2 - Fläche zwischen 20 m² und 100 m²

Die Kenngrößen des zweiten Raumtyps in Tabelle 5.2 zeigen am besten die Schwächen und Verbesserungsmöglichkeiten des vorgestellten Positionsbestimmungssystems. In dem sehr schmalen und von starken Störquellen belasteten Versuchsraum konnten große Abweichungen der berechneten Positionen von den tatsächlichen Nutzerpositionen festgestellt werden. Der maximale durchschnittliche Fehler der durchgeführten Versuche liegt bei 2,46 m und ist damit weit unter der angestrebten Grenze von 5 m. Der maximale Fehler der Versuche hingegen liegt bei etwa sieben Metern und befindet sich damit über der festgelegten Grenze. Diese hohen Abweichungen entstanden vor allem durch eine stark gestörte Richtungsbestimmung auf Basis eines Magnetometers. Dieser weist in der Nähe von starken Stromquellen einen Fehler

Tabelle 5.2: Kenngrößen für Raum 2 - Raumfläche zwischen 20 m^2 und 100 m^2

Versuch #	Mittelwert μ_{error}	Mittelwert σ_u	max_{error}	max_{σ_u}
1	173.77 cm	39.3409 cm	385.92 cm	56.2112 cm
2	246.03 cm	39.8964 cm	678.80 cm	57.3307 cm
3	152.06 cm	38.4997 cm	273.72 cm	56.5002 cm
4	143.05 cm	38.6105 cm	235.92 cm	55.9902 cm
5	177.38 cm	38.1978 cm	328.40 cm	56.2912 cm
6	173.66 cm	38.7043 cm	388.86 cm	55.1160 cm
7	175.85 cm	38.5438 cm	351.22 cm	55.7297 cm
8	195.36 cm	36.2335 cm	597.85 cm	55.8175 cm
9	181.70 cm	39.8337 cm	635.14 cm	57.1114 cm
10	237.15 cm	38.5341 cm	693.21 cm	55.9932 cm
Mittelwert	185.60 cm	38.6395 cm	456.90 cm	56.2091 cm
Max	246.03 cm	39.8964 cm	693.21 cm	57.3307 cm

von etwa 25° bis 85° auf und trägt so zu einer, wie in Abbildung 5.4 zu erkennen, stark verfälschten inertialen Position bei. In unmittelbarer Nähe waren zudem kei-



(a) Versuch 2 (b) Versuch 9 (c) Versuch 10

Abbildung 5.4: Visualisierungen der Versuche 2, 9 und 10 des Raumtyps 2. Es ist eine deutliche Abweichung der berechneten Richtung der Bewegungsvektoren (magenta Geraden) von der realen Strecke (blaue Gerade) zu erkennen. Als Verursacher dieser Störung konnte ein Serverraum und ein Stromschaltkasten (violett) an der linken Seite identifiziert werden.

ne Beacons platziert, wodurch eine kontinuierliche Korrektur der falsch berechneten Positionen nicht durchgeführt werden konnte.
Des Weiteren kam der in Abschnitt 5.3.4 beschriebene Effekt der verzögerten Korrekturdaten zum Vorschein. Die Korrektur der zuvor berechneten Positionen wurde durchgeführt, nachdem der Nutzer sich bereits wieder vom Beacon entfernt hatte. Die mittels PDR in diesem Zeitraum berechnete Bewegung wurde teilweise rückgängig gemacht, wodurch die berechnete Position von der Realen stärker abwich.
Des Weiteren wurden in einigen Versuchen keine Korrekturen durch alle Beacons durchgeführt. Einige Beacons, vor allem aber jene, die über einen längeren Zeitraum

in großen Entfernungen empfangen wurden, konnten nicht zur Korrektur der Positionen genutzt werden. Die berechnete Distanz zwischen Nutzer und Sender war dabei so hoch, dass diese den Relevanzgrenzwert von 1,5 m überschritt und damit nicht genutzt werden konnte. Tatsächlich war der Nutzer jedoch nur wenige Dezimeter vom Sender entfernt.

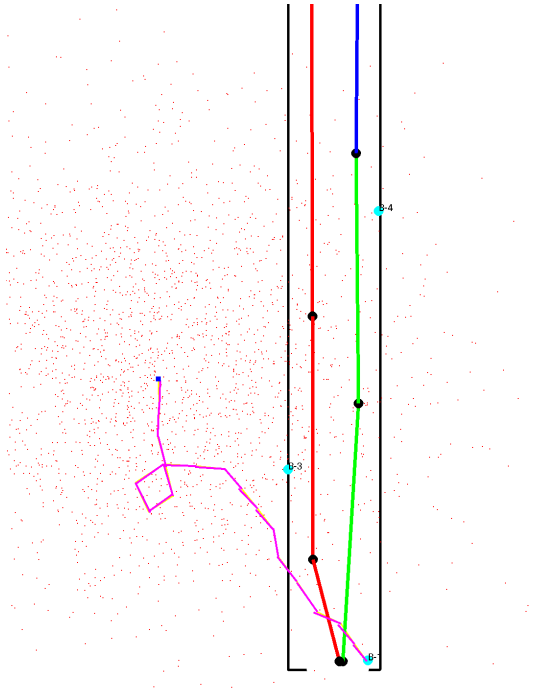


Abbildung 5.5: Die berechnete Position wurde nicht mit Hilfe der Beacons 3 und 4 korrigiert, obwohl der Nutzer diese sehr nah passiert hat. Die grüne Linie stellt den vom Nutzer bereits absolvierten Weg dar. Bei einer Korrektur würde ein blauer Kreis um den Beacon angezeigt und die der Positionsverlauf (magenta) zu einem Beacon hin verschoben werden.

Die in diesem Raumtyp aufgetretenen Fehler zeigen die hohe Fehleranfälligkeit der genutzten Sensoren. In weiteren Untersuchungen sollten daher zusätzliche Sensoren und Informationen über die Umgebung genutzt werden, um die Fehleranfälligkeit weiter zu minimieren und die Genauigkeit der Positionsbestimmung zu steigern.

5.3.3 Raumtyp 3 - Raumfläche über $100\,\mathrm{m}^2$

Der Raumtyp 3 ist vor allem durch die große Raumfläche und die damit verbundene geringere Abdeckung mit Bluetooth-Low-Energy-Sendern interessant. In diesem Raumtyp kann die Idee des vorgestellten Konzeptes einer kontinuierlichen inertialen Positionsbestimmung mit korrigierend wirkenden Radiosendern besonders gut untersucht werden.
Die in Tabelle 5.3 dargestellten Ergebnisse zeigen, dass der maximale durchschnittliche Fehler für diesen Raumtyp unter drei Metern liegt und damit weit besser als der festgelegte Grenzwert von fünf Metern ist. Zudem überschreitet der maximale Fehler

Tabelle 5.3: Kenngrößen für Raum 3 - Raumfläche über 100 m^2

Versuch #	Mittelwert μ_{error}	Mittelwert σ_u	$\max_{error}$	$\max_{\sigma_u}$
1	255.69 cm	23.7279 cm	466.86 cm	39.7930 cm
2	246.99 cm	23.2404 cm	472.00 cm	41.7508 cm
3	201.69 cm	23.0973 cm	298.72 cm	38.0902 cm
4	249.62 cm	23.3878 cm	371.73 cm	39.7053 cm
5	277.30 cm	22.8864 cm	494.52 cm	39.3288 cm
6	227.61 cm	22.6634 cm	433.87 cm	36.2100 cm
7	201.17 cm	22.9669 cm	403.84 cm	39.0831 cm
8	204.91 cm	24.4517 cm	348.92 cm	38.2040 cm
9	212.73 cm	24.4175 cm	424.88 cm	39.9924 cm
10	245.15 cm	23.2990 cm	472.07 cm	43.4846 cm
11	235.27 cm	22.5179 cm	500.88 cm	41.0525 cm
Mittelwert	232.56 cm	23.3324 cm	426.21 cm	39.6995 cm
Max	277.30 cm	24.4517 cm	500.88 cm	43.4846 cm

des Raumtyps über alle Versuche nur minimal diesen Grenzwert. Der durchschnittliche maximale Fehler aller Versuche liegt hingegen deutlich unter der 5 m-Grenze.
So wie bereits bei Raumtyp 2 festgestellt, erzeugt die Richtungsbestimmung durch ein Magnetometer vor allem in Gebäuden ein hohes Fehlerpotenzial. Diese Fehler können durch die Korrektursignale der BLE-Beacons immer wieder minimiert werden, führen jedoch in der Nähe von starken Störfaktoren des Magnetfeldes zu signifikanten Fehlern bei der Positionsbestimmung.
Weiterhin konnte in den Versuchen dieses Raumtyps die Güte des Schrittlängeschätzers beobachtet werden. Der Schätzer lieferte, je nach Schrittlänge, aber auch je nach Auftrittsintensität, unterschiedliche Werte und zudem einige unrealistische Schätzungen. Eine Schrittlänge von über zwei Metern kam dabei vor allem bei sehr intensiven Schritten vor und hat die Positionsbestimmung stark negativ beeinträchtigt.
Es wurde daher eine maximale Schrittlänge eingeführt, welche nicht überschritten werden durfte.

5.3.4 Raumtypunabhängige Beobachtungen

Bei den durchgeführten Versuchen konnten verschiedene Effekte und Störgrößen festgestellt werden, welche die Genauigkeit der Positionsbestimmung negativ beeinträchtigen können. Die im Folgenden beschriebenen Beobachtungen sind dabei vom Raum unabhängig und konnten überall beobachtet werden.

Bestimmung der Referenzleistung bei iBeacon-Sendern

Es wurde in Abschnitt 2.1.2 beschrieben, dass das iBeacon-Protokoll eine Referenzsendeleistung mit jedem Paket überträgt und diese Leistung für die Berechnung der Distanzen genutzt werden kann. Der übertragene Wert konnte bei der verwendeten Hardware jedoch nicht manuell konfiguriert werden. Dieser wird anscheinend über die konfigurierbare Sendeleistung berechnet und stimmt mit realen Werten oft nicht

überein. Die verwendeten BLE-Beacons von Beaconinside haben bei einer konfigurierten Sendeleistung von 0 dBm einen TX-Wert von −61 dBm übertragen. Dieser oder ein ähnlicher Wert müsste in einem Meter Entfernung beim Empfänger festgestellt werden können. Eine Messung hat hingegen bei jedem verwendeten Beacon eine Sendeleistung von −56 dBm bzw. −52 dBm ergeben. Diese Messwerte wurden durch ein zweites Smartphone eines anderen Herstellers (Sony M4 Aqua) überprüft und bestätigt.
Schlussfolgernd kann festgehalten werden, dass die im iBeacon-Protokoll übertragene Sendeleistung nicht zwangsläufig korrekt sein muss. Sie sollte überprüft und die exakten Messwerte sollten gegebenenfalls in einer Datenbank abgespeichert und zur weiteren Distanzberechnung hinzugezogen werden.
Die Beobachtung konnte nicht bei anderen Herstellern überprüft werden, da keine Sender dieser verfügbar waren.

Einfluss der RSSI-Filter auf die Positionsbestimmung

Messungen der Signalstärke ankommender Bluetooth-Low-Energy-Pakete unterliegen starken Störfaktoren und die ermittelten Messwerte sind oft hohem Rauschen ausgesetzt (vgl. Abschnitt 2.1.4). Diese störenden Elemente müssen durch Filter möglichst gut entfernt werden, um die weiteren Berechnungen so fehlerfrei wie möglich durchführen zu können. Die eingesetzten Filter besitzen eine vom Filterkern abhängige Phasenverzögerung $t_{Verzoegerung}$, wodurch aktuelle Messwerte zwar gefiltert aber auch verzögert für die Weiterverarbeitung bereit stehen. Diese Verzögerung macht sich bei der Korrektur der berechneten Position bemerkbar und führt zu verfälschten Ergebnissen.
Sollte ein Nutzer schnell und nah an einem Sender vorbei gehen, so schwanken die RSSI-Werte in diesem kurzen Zeitraum stark und werden durch den Mittelwertfilter in ihrem Einfluss gemindert. Gleichzeitig stellt der Filter einen RSSI-Wert zum Zeitpunkt t bereit, welcher aber für den Zeitpunkt $t - t_{Verzoegerung}$ gültig ist. Weil der Nutzer beim nahen Vorbeigehen am Sender sich schnell wieder von diesem entfernt, ist er im Zeitpunkt t signifikant weiter vom Sender entfernt als die Messwerte dies repräsentieren. Die Korrektur der berechneten Position erfolgt jedoch auf Basis der gefilterten und verzögerten Messwerte. Der so entstehende Fehler beeinflusst die Güte Positionsbestimmung stark und kann je nach Wahl der Filterkerngröße und der Laufgeschwindigkeit des Nutzers mehrere Meter betragen. Dieser Effekt tritt vor allem in kleinen oder schmalen Räumen, wie den Raumtypen eins und zwei auf.

Einfluss des Schrittlängenschätzers

Des Weiteren konnte der Einfluss des Schrittlängenschätzers auf die Positionsberechnung beobachtet werden. Der gewählte dynamische Schätzer von J. Scarlet benötigt eine an den Nutzer angepasste Parameterauswahl, um die Schrittlänge zuverlässig berechnen zu können. Es konnte festgestellt werden, dass der Schätzer nicht auf die Dynamik gleichlanger Schritte mit unterschiedlicher Auftrittsintensität angepasst ist.
Sollte eine Person besonders intensiv die Füße aufsetzen, so wird dies als ein längerer Schritt interpretiert. Bei besonders sanftem Aufsetzen der Füße wird hingegen eine sehr geringe Schrittlänge berechnet. Diese Beobachtung ist nicht direkt in den vorliegenden Daten ersichtlich. sondern wurde bei der Bestimmung der benötigten Parameter festgestellt.

Unsicherheit der berechneten Position

Wie aus den Daten aller Versuche hervorgeht, war die Standardabweichung der Positionshypothesen stets gering und betrug an jeder festgelegten Station maximal 57 cm. Diese Standardabweichungen sind vor allem auf Grund der häufigen Korrekturen durch Bluetooth-Low-Energy-Sender so gering. Die Korrektur wird immer nur bei sehr geringen Entfernungen zwischen Nutzer und Sender durchgeführt, wodurch die möglichen Positionen auf einen stark begrenzten Bereich reduziert werden. Dies ist in Abbildung 5.6 deutlich zu sehen. Die Unsicherheit der berechneten Position sinkt und die Standardverteilung der Partikelwolke wird reduziert.
Zudem konnte festgestellt werden, dass die Messwerte des Magnetometers nur geringe Varianzen aufweisen und somit nicht so stark vom Rauschen betroffen sind. Die Unsicherheit der Position nach einem Schritt steigt auf Grund der geringen Varianz (vgl. Abschnitt 3.3.2) geringer an als angenommen. Die kontinuierlichen Störgrößen, wie externe Magnetfelder, fließen hingegen nicht in die Berechnung der Unsicherheiten ein und bleiben somit nicht betrachtetet.

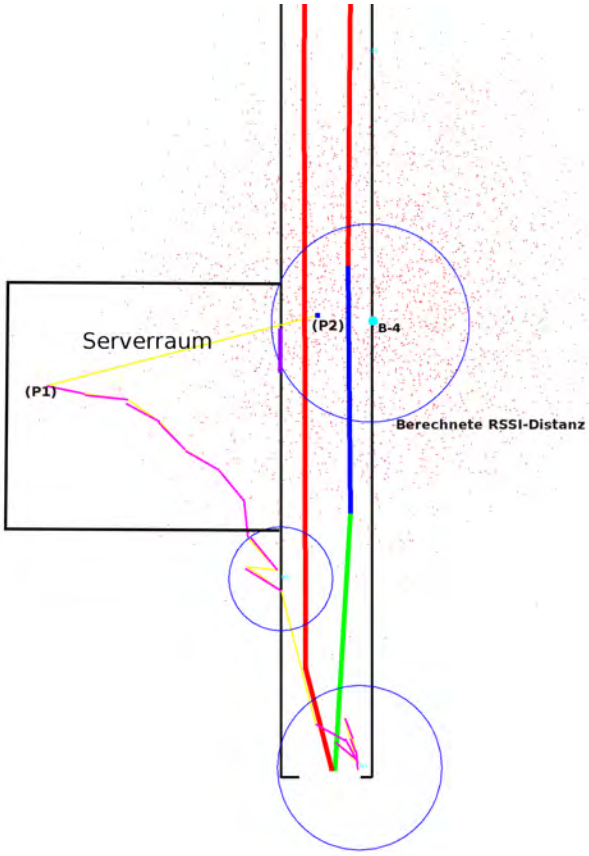


Abbildung 5.6: Korrektur einer stark verfälschten Position (P1) durch einen in der Nähe erkannten iBeacon B-4. Die falsche Position P1 wird an die bessere Position P2 verschoben. Diese Bewegung hat keine PDR-Messung (magentafarbene Geraden) als Grundlagen und wird deshalb als gelbe Gerade dargestellt. Die neue Position P2 ist nach der Korrektur wieder in der Nähe des realen, blau dargestellten, Pfades des Nutzers und repräsentiert die reale Nutzerposition besser.

5.3.5 Einschätzung des Systems

Das vorgestellte Positionsbestimmungssystem erreicht zusammenfassend gesehen gute Ergebnisse bei der Positionsbestimmung. Das angestrebte Ziel, für jeden betrachteten Raumtyp einen maximalen Positionsfehler unter 5 m zu erreichen, wurde jedoch nicht erreicht, obwohl die durchschnittlichen Fehlerwerte deutlich unter dieser Grenze liegen.
Vor allem im Versuchsraum 2 wurden zu hohe Fehler festgestellt, welche auf externe Störungen durch Magnetfelder zurück zu führen sind. Zudem verhinderte der geringe Abstand zwischen den Wänden eine gute Korrektur der fehlerhaften Positionen, da die RSSI-Messungen ebenfalls gestört waren und Korrekturen teilweise nur verzögert ausgeführt wurden (siehe Abschnitt 5.3.4).
Im kleinsten Versuchsraum (unter 20 m^2 Raumfläche) konnte mit dem vorgestellten System eine gute Leistung erreicht werden. Der durchschnittliche Fehler lag nur bei 1,05 m. Der maximale Fehler bei allen Versuchen betrug hingegen 3,60 m und war somit weit unter der gesetzten Grenze von 5 m.
Im flächenmäßig größten Versuchsraum konnten ebenfalls zufriedenstellende Ergebnisse erreicht werden. Der durchschnittliche Fehler lag bei 2,33 m. Der maximale Fehler aller Versuche lag hingegen bei 5,01 m und damit gering über der festgelegten Grenze. Dies kann jedoch als statistischer Ausreißer gesehen werden, da der durchschnittliche maximale Fehler 4,26 m betrug und damit deutlich unter der 5 m-Grenze war.
Das in Kapitel 3 vorgestellte System ist laut den erreichten Ergebnissen nicht für alle Raumarten und Umgebungen gleich gut geeignet und sollte daher durch zusätzliche Sensoren und Informationen erweitert werden. Ein Gyroskop kann dabei, wie von M. H. Afzal et. al. [ARL11a] vorgeschlagen, temporären Störungen durch starke Magnetfelder entgegenwirken, indem eine Veränderung der magnetfeldbasierten Richtung durch die relative Rotation des Gerätes verifiziert wird.
Für den Einsatz eines solchen Indoor-Positionsbestimmungssystems muss zudem eine geeignete Lösung zur Schrittlängenberechnung gefunden werden, welche dynamisch verschiedene Gangarten und Auftrittintensitäten der Nutzer berücksichtigt und so bessere Ergebnisse als der gewählte Scarlet-Algorithmus (vgl. Abschnitt 2.2.4.2) liefert.
Der Vorschlag, RF-Sender als Korrektursignalgeber zu nutzen, lieferte hingegen positive Ergebnisse. Die geringe Reichweite der verwendeten Sender konnte bei jedem Korrekturvorgang die mögliche Position auf wenige Quadratmeter beschränken und somit den durch PDR induzierten Fehler stark reduzieren. Es müssen jedoch weitere detaillierte Versuche durchgeführt werden, bei denen unterschiedliche Filter und Filterkerngrößen verglichen werden. Diese sollen vor allem die Genauigkeit der Korrekturvorgänge erhöhen und die Verzögerungen eliminieren.

6. Diskussion

6.1 Zusammenfassung

In dieser Arbeit wurde ein Indoor-Positionsbestimmungssystem konzipiert und evaluiert welches mit Hilfe eines handelsüblichen Smartphones und installierter Bluetooth-Low-Energy-Beacons die Position eines Nutzers berechnen kann. Dabei wird ein Pedestrian-Dead-Reckoning-Algorithmus zur kontinuierlichen Positionsbestimmung genutzt. Die im Raum installierten Bluetooth-Low-Energy-Beacons werden hingegen zur Initialisierung des gesamten Systems und zur Korrektur der stark störanfälligen inertialen Positionsbestimmung eingesetzt. Diese Kombination ermöglicht eine gute und ausreichend genaue Positionsberechnung, ohne dabei komplexe Installationen in den Gebäuden vorauszusetzen.
In zahlreichen Versuchen wurde das vorgestellte und implementierte System in verschiedenen Umgebungen untersucht und die Güte des Positionsbestimmungssystems bestimmt. Das Ziel, in jedem für Ausstellungen typischem Raum einen maximalen Positionsfehler von 5 Metern nicht zu überschreiten, konnte nicht vollständig erreicht werden.
In einem stark mit Störgrößen belastetem Versuchsraum wurde der gewünschte maximale Fehler in einigen Versuchen überschritten. Die starken Magnetfeldstörungen durch elektromagnetische Felder und häufig auftretende Multi-Path-Effekte haben maximale Positionsfehler von bis zu 6,90 m verursacht. Die erhofften Korrektureffekte durch RSSI-Distanzmessungen konnten dabei nicht in jedem Fall genutzt werden, da Multi-Path-Effekte die korrekte Distanzbestimmung störten.
In weniger von Störeffekten belasteten Räumen konnte hingegen eine gute Positionsgenauigkeit erreicht werden. Der durchschnittliche Fehler lag in dem über 100 m^2-großen Versuchsraum im Bereich von 2,33 m und ist damit weit unter der 5 m-Grenze. Auch bei sehr kleinen Räumen mit weniger als 20 m^2 Raumfläche, konnte eine gute durchschnittliche Genauigkeit gemessen werden. Mit nur einem Meter durchschnittlicher Abweichung und einem maximalen Fehler von 3,60 m hat das System die Erwartungen erfüllt.
Es sollte dennoch festgehalten werden, dass das Positionsbestimmungssystem in der vorgeschlagenen Form nicht für einen produktiven Einsatz geeignet ist. Vor allem die

Störanfälligkeit gegenüber externen Magnetfeldern und die Ungenauigkeiten bei der Schrittlängenschätzung müssen korrigiert werden, bevor ein ausgedehnter Einsatz in kulturellen Einrichtungen in Erwägung gezogen werden kann.
Diese Änderungen sind jedoch mit weiteren Untersuchungen möglich und stellen nicht das gesamte Konzept in Frage. Viel mehr konnte in den Versuchen gezeigt werden, dass die grundlegende Überlegung, die inertiale Positionsbestimmung mit sporadisch installierten RF-Sendern zu verbessern und die entstehenden Fehler zu korrigieren, eine erfolgversprechende Annahme war. Zudem konnte durch die Implementierung des rein offline agierenden Prototypen gezeigt werden, dass heutige Smartphones genügend Leistung für die kontinuierliche Indoor-Positionsbestimmung bereitstellen und damit nicht auf externe Dienste und Server angewiesen sind. Dies ist eine wichtige Feststellung auf dem Weg zu einer kontrollierbaren und datenschutzunbedenklichen Positionsbestimmungslösung in geschlossenen Räumen.

6.2 Ausblick

Das hier vorgestellte Konzept eines offline funktionsfähigen Positionsbestimmungssystems für geschlossene Räumlichkeiten konnte das gesetzte Genauigkeitsziel auf Grund der hohen Anfälligkeit für magnetische Störquellen nicht erreichen. Im Weiteren muss zuallererst die Richtungsbestimmung robuster gestaltet werden. Diese kann durch Verwendung weiterer Sensoren, wie eines Gyroskops[ARL11a], oder durch die Einbeziehung von Gebäudestrukturen[EMN05] erreicht werden.
Die Genauigkeit der Position hängt auch von der Schrittlängenschätzung ab. Diese muss in weiteren Versuchen untersucht und die Eignung des gewählten Scarlet-Algorithmus und weiterer Algorithmen (siehe zum Beispiel [JBS+10] oder [RSL12]) und Optimierungen (siehe [LSVW11]) für eine breite Anwendermasse festgestellt werden. In den bisher durchgeführten Versuchen war aufgefallen, dass die Schrittlängenschätzung nach J. Scarlet[Sca07] für Variationen der Auftrittsintensität anfällig ist und dadurch stark verfälschte Schätzungen liefert.
Die Korrektur der Positionen durch Messung der RSSI-Signale hat größtenteils gute Ergebnisse geliefert, kann jedoch optimiert werden. Vor allem die Filterung der wenigen verfügbaren Messwerte muss weiter verbessert werden, damit Multi-Path-Fehler und das starke Rauschen die Distanzbestimmung nicht im bisherigen Maß beeinflussen.

Literaturverzeichnis

[AQD+02] Bao-Ling Adam, Yinsheng Qu, John W Davis, Michael D Ward, Mary Ann Clements, Lisa H Cazares, O John Semmes, Paul F Schellhammer, Yutaka Yasui, Ziding Feng, et al. Serum protein fingerprinting coupled with a pattern-matching algorithm distinguishes prostate cancer from benign prostate hyperplasia and healthy men. *Cancer research*, 62(13):3609–3614, 2002. (zitiert auf Seite 17)

[ARL11a] Muhammad Haris Afzal, Valérie Renaudin, and Gérard Lachapelle. Magnetic field based heading estimation for pedestrian navigation environments. In *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, pages 1–10. IEEE, 2011. (zitiert auf Seite 20, 64 und 66)

[ARL11b] Muhammad Haris Afzal, Valerie Renaudin, and Gerard Lachapelle. Multi-magnetometer based perturbation mitigation for indoor orientation estimation. *Navigation*, 58(4):279–292, 2011. (zitiert auf Seite 14)

[Bak85] James Edward Baker. Adaptive selection methods for genetic algorithms. In *Proceedings of an International Conference on Genetic Algorithms and their applications*, pages 101–111. Hillsdale, New Jersey, 1985. (zitiert auf Seite 24)

[Bak87] James E Baker. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the second international conference on genetic algorithms*, pages 14–21, 1987. (zitiert auf Seite 23)

[Bea06] Stephane Beauregard. A Helmet-Mounted Pedestrian Dead Reckoning System. In *Applied Wearable Computing (IFAWC), 2006 3rd International Forum on*, pages 1–11. VDE, March 2006. (zitiert auf Seite 13, 16 und 17)

[BH06] Stephane Beauregard and Harald Haas. Pedestrian dead reckoning: A basis for personal positioning. In *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication*, pages 27–35, 2006. (zitiert auf Seite 19)

[BH13] Agata Brajdic and Robert Harle. Walk detection and step counting on unconstrained smartphones. In *HUC*, 2013. (zitiert auf Seite 16 und 17)

[BHS97] Thomas Bäck, Ulrich Hammel, and Hans-Paul Schwefel. Evolutionary computation: Comments on the history and current state. *Evolutionary computation, IEEE Transactions on*, 1(1):3–17, 1997. (zitiert auf Seite 24)

[BLE10a] In *Bluetooth Core Specification 4.0*, pages 52–54. Bluetooth SIG, 2010. (zitiert auf Seite 5, 6 und 10)

[BLE10b] In *Bluetooth Core Specification 4.0 Part G*, pages 521–592. Bluetooth SIG, 2010. (zitiert auf Seite 11)

[Bli00] Tobias Blickle. Tournament selection. *Evolutionary computation*, 1:181–186, 2000. (zitiert auf Seite 24)

[Boe93] Paul Boersma. Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. In *Proceedings of the institute of phonetic sciences*, volume 17, pages 97–110. Amsterdam, 1993. (zitiert auf Seite 17)

[BP00] Paramvir Bahl and Venkata N Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784. Ieee, 2000. (zitiert auf Seite 5 und 12)

[CDDV07] Francois Caron, Manuel Davy, Emmanuel Duflos, and Philippe Vanheeghe. Particle filtering for multisensor data fusion with switching observation models: Application to land vehicle positioning. *Signal Processing, IEEE Transactions on*, 55(6):2703–2719, 2007. (zitiert auf Seite 20 und 21)

[Cha97] Averil Burton Chatfield Chatfield. *Fundamentals of high accuracy inertial navigation*, volume 174. Aiaa, 1997. (zitiert auf Seite 15)

[CR12] Ivan Casacuberta and Adrian Ramirez. Time-of-flight positioning using the existing wireless local area network infrastructure. In *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, pages 1–8. IEEE, 2012. (zitiert auf Seite 28)

[CTC12] Yohan Chon, E. Talipov, and Hojung Cha. Autonomous management of everyday places for a personalized location provider. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(4):518–531, July 2012. (zitiert auf Seite 17)

[DGA00] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000. (zitiert auf Seite 21)

[DJ09] Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12(656-704):3, 2009. (zitiert auf Seite 21, 22 und 24)

[DM14] Erik Dahlgren and Hasan Mahmood. Evaluation of indoor positioning based on bluetoothr smart technology. 2014. (zitiert auf Seite 5, 12 und 13)

[EM06] Frédéric Evennou and François Marx. Advanced integration of wifi and inertial navigation systems for indoor mobile positioning. *Eurasip journal on applied signal processing*, 2006:164–164, 2006. (zitiert auf Seite 11)

[EMN05] Frédéric Evennou, François Marx, and Emil Novakov. Map-aided indoor mobile positioning system using particle filter. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 4, pages 2490–2494. IEEE, 2005. (zitiert auf Seite 66)

[fM15] Institut für Museumsforschung. In *Statistische Gesamterhebung an den Museen der Bundesrepublik Deutschland für das Jahr 2014*. Staatliche Museen zu Berlin - Preußischer Kulturbesitz, 2015. (zitiert auf Seite 1)

[Gas14] Matthew S Gast. *Building applications with IBeacon: proximity and location services with bluetooth low energy*. O'Reilly Media, Inc., 2014. (zitiert auf Seite 6)

[GGB+02] Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forssell, Jonas Jansson, Rickard Karlsson, and Per-Johan Nordlund. Particle filters for positioning, navigation, and tracking. *Signal Processing, IEEE Transactions on*, 50(2):425–437, 2002. (zitiert auf Seite 20, 21 und 24)

[GLG+12] Michael Gunawan, Bing Li, Thomas Gallagher, Andrew G Dempster, and Günther Retscher. A new method to generate and maintain a wifi fingerprinting database automatically by using rfid. In *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, pages 1–6. IEEE, 2012. (zitiert auf Seite 12)

[HB04] Jeffrey Hightower and Gaetano Borriello. Particle filters for location estimation in ubiquitous computing: A case study. In *In Proceedings of International Conference on Ubiquitous Computing (UbiComp*, pages 88–106, 2004. (zitiert auf Seite 20 und 21)

[iBe15] Proximity Beacon Specification Release R1. https://developer.apple.com/ibeacon/, 2015. (zitiert auf Seite 6)

[Inc] BlueRadios Inc. nBlue Protocol Stack clear. http://www.blueradios.com/images/nBlue_Protocol_Stack_clear.gif. (zitiert auf Seite ix und 6)

[JBS+10] J. Jahn, U. Batzer, J. Seitz, L. Patino-Studencka, and J. Gutiérrez Boronat. Comparison and evaluation of acceleration based step length estimators for handheld devices. In *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, pages 1–6, Sept 2010. (zitiert auf Seite 66)

[JSPG09] Antonio R Jimenez, Fernando Seco, Carlos Prieto, and Jorge Guevara. A comparison of pedestrian dead-reckoning algorithms using a low-cost mems imu. In *Intelligent Signal Processing, 2009. WISP 2009. IEEE International Symposium on*, pages 37–42. IEEE, 2009. (zitiert auf Seite 18)

[JTSW11] Yunye Jin, Hong-Song Toh, Wee-Seng Soh, and Wai-Choong Wong. A robust dead-reckoning pedestrian tracking system with low cost sensors. In *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on*, pages 222–230. IEEE, 2011. (zitiert auf Seite 15, 16, 17 und 20)

[KIK10] Masakatsu Kourogi, Tomoya Ishikawa, and Takeshi Kurata. A method of pedestrian dead reckoning using action recognition. In *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION*, pages 85–89. IEEE, 2010. (zitiert auf Seite 16 und 17)

[KJHP04] Jeong Won Kim, Han Jin Jang, Dong-Hwan Hwang, and Chansik Park. A step, stride and heading determination for the pedestrian navigation system. *Positioning*, 1(08):0, 2004. (zitiert auf Seite 17 und 19)

[KNHL12] Wonho Kang, Seongho Nam, Youngnam Han, and Sookjin Lee. Improved heading estimation for smartphone-based indoor positioning systems. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*, pages 2449–2453. IEEE, 2012. (zitiert auf Seite 20)

[KR77] JF Kaiser and WA Reed. Data smoothing using low-pass digital filters. *Review of Scientific Instruments*, 48(11):1447–1457, 1977. (zitiert auf Seite 14)

[Kün05] Hans R Künsch. Recursive monte carlo filters: algorithms and theoretical analysis. *Annals of Statistics*, pages 1983–2021, 2005. (zitiert auf Seite 21)

[LCL15] Hwan-hee Lee, Suji Choi, and Myeong-jin Lee. Step detection robust against the dynamics of smartphones. *Sensors*, 15(10):27230, 2015. (zitiert auf Seite 14, 17 und 18)

[LL11] Adam Lipowski and Dorota Lipowska. Roulette-wheel selection via stochastic acceptance. *CoRR*, abs/1109.3627, 2011. (zitiert auf Seite 23)

[LSVW11] J. Á B. Link, P. Smith, N. Viol, and K. Wehrle. Footpath: Accurate map-based indoor navigation using smartphones. In *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, pages 1–8, Sept 2011. (zitiert auf Seite 66)

[MA98] Claus B Madsen and Claus S Andersen. Optimal landmark selection for triangulation of robot position. *Robotics and Autonomous Systems*, 23(4):277–292, 1998. (zitiert auf Seite 12)

[MG95] Brad L Miller and David E Goldberg. Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, 9(3):193–212, 1995. (zitiert auf Seite 24)

[MVFB10] Eladio Martin, Oriol Vinyals, Gerald Friedland, and Ruzena Bajcsy. Precise indoor localization using smart phones. In *Proceedings of the international conference on Multimedia*, pages 787–790. ACM, 2010. (zitiert auf Seite 12)

[PPC+12] Jun-geun Park, Ami Patel, Dorothy Curtis, Seth Teller, and Jonathan Ledlie. Online pose classification and walking speed estimation using handheld devices. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 113–122. ACM, 2012. (zitiert auf Seite 19)

[PS10] Sang Kyeong Park and Young Soo Suh. A zero velocity detection algorithm using inertial sensors for pedestrian navigation systems. *Sensors*, 10(10):9163–9178, 2010. (zitiert auf Seite 15)

[PWH12] Azkario Rizky Pratama, Widyawan, and Risanuri Hidayat. Smartphone-based pedestrian dead reckoning as an indoor positioning system. Sep 2012. (zitiert auf Seite 13, 14, 17, 18 und 20)

[QMYL13] Jiuchao Qian, Jiabin Ma, Rendong Ying, and Peilin Liu. Rpnos: Reliable pedestrian navigation on a smartphone. In *Geo-Informatics in Resource Management and Sustainable Ecosystem*, pages 188–199. Springer, 2013. (zitiert auf Seite 13, 15 und 16)

[RSL12] Valérie Renaudin, Melania Susi, and Gérard Lachapelle. Step length estimation using handheld inertial sensors. *Sensors*, 12(7):8507–8525, 2012. (zitiert auf Seite 66)

[RZJM07] Liu Rong, Duan Zhiguo, Zhou Jianzhong, and Liu Ming. Identification of individual walking patterns using gait acceleration. In *Bioinformatics and Biomedical Engineering, 2007. ICBBE 2007. The 1st International Conference on*, pages 543–546. IEEE, 2007. (zitiert auf Seite 17)

[Sca07] Jim Scarlett. Enhancing the performance of pedometers using a single accelerometer. *Application Note, Analog Devices*, 2007. (zitiert auf Seite 18, 19 und 66)

[SHNR10] Isaac Skog, Peter Händel, John-Olof Nilsson, and Jouni Rantakokko. Zero-velocity detection—an algorithm evaluation. *Biomedical Engineering, IEEE Transactions on*, 57(11):2657–2666, 2010. (zitiert auf Seite 15)

[Tal09] El-Ghazali Talbi. *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons, 2009. (zitiert auf Seite 23 und 24)

[VK95] Martin Vetterli and Jelena Kovacevic. *Wavelets and subband coding*. Number LCAV-BOOK-1995-001. Prentice-hall, 1995. (zitiert auf Seite 16 und 17)

[VLB+08] A. Vlavianos, L. K. Law, I. Broustis, S. V. Krishnamurthy, and M. Faloutsos. Assessing link quality in ieee 802.11 wireless networks: Which is the right metric? In *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, pages 1–6, Sept 2008. (zitiert auf Seite 10)

[WBLP09] Xinwei Wang, Ole Bischoff, Rainer Laur, and Steffen Paul. Localization in wireless ad-hoc sensor networks using multilateration with rssi for logistic applications. *Procedia Chemistry*, 1(1):461–464, 2009. (zitiert auf Seite 11)

[Wei02] Harvey Weinberg. Using the adxl202 in pedometer and personal navigation applications. 2002. (zitiert auf Seite 19 und 20)

[WF10] Neeti Wagle and Eric W Frew. A particle filter approach to wifi target localization. In *AIAA Guidance, Navigation, and Control Conference*, pages 2287–2298, 2010. (zitiert auf Seite 21 und 22)

[WGH+11] Wei-zhong Wang, Yan-wei Guo, Bang-Yu Huang, Guo-ru Zhao, Bo-qiang Liu, and Lei Wang. Analysis of filtering methods for 3d acceleration signals in body sensor network. In *Bioelectronics and Bioinformatics (ISBB), 2011 International Symposium on*, pages 263–266. IEEE, 2011. (zitiert auf Seite 13 und 14)

[WH08] Oliver Woodman and Robert Harle. Pedestrian localisation for indoor environments. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 114–123. ACM, 2008. (zitiert auf Seite 21)

[WKM08] Carl Wong, Richard Klukas, and Geoffrey Messier. Using wlan infrastructure for angle-of-arrival indoor user location. In *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*, pages 1–5. IEEE, 2008. (zitiert auf Seite 12)

[Woo07] Oliver J Woodman. An introduction to inertial navigation. *University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696*, 14:15, 2007. (zitiert auf Seite 13, 15, 16, 20 und 24)

[WT04] Jan Wendel and Gert F Trommer. Tightly coupled gps/ins integration for missile applications. *Aerospace Science and Technology*, 8(7):627–634, 2004. (zitiert auf Seite 15)

[WYZ+13] Yapeng Wang, Xu Yang, Yutian Zhao, Yue Liu, and L. Cuthbert. Bluetooth positioning using rssi and triangulation methods. Jan 2013. (zitiert auf Seite 5 und 11)

[XLL+10] Jiuqiang Xu, Wei Liu, Fenggao Lang, Yuanyuan Zhang, and Chenglong Wang. Distance measurement model based on rssi in wsn. *Wireless Sensor Network*, 2(08):606, 2010. (zitiert auf Seite 11)

[YA05] Moustafa Youssef and Ashok Agrawala. The horus wlan location determination system. In *Proceedings of the 3rd international conference on*

Mobile systems, applications, and services, pages 205–218. ACM, 2005. (zitiert auf Seite 5)

[YBM08] Xiaoping Yun, Eric R Bachmann, and Robert B McGhee. A simplified quaternion-based algorithm for orientation estimation from earth gravity and magnetic field measurements. *Instrumentation and Measurement, IEEE Transactions on*, 57(3):638–650, 2008. (zitiert auf Seite 20)

[ZZZZ08] Giovanni Zanca, Francesco Zorzi, Andrea Zanella, and Michele Zorzi. Experimental comparison of rssi-based localization algorithms for indoor wireless sensor networks. In *Proceedings of the workshop on Real-world wireless sensor networks*, pages 1–5. ACM, 2008. (zitiert auf Seite 12)

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Magdeburg, den 29. März 2016