



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

Fakultät für Mathematik, Informatik und Naturwissenschaften



Bachelorarbeit

Sensorgestützte WLAN-Positionsbestimmung durch maschinelle Lernverfahren zur effizienten Indoor Navigation

Daniel Gleim

8gleim@informatik.uni-hamburg.de

Studiengang Wirtschaftsinformatik

Matr.-Nr. 6052405

Fachsemester 7

Erstgutachter: Professor W. Lamersdorf

Zweitgutachter: Professor H. Federrath

Zwei Dinge sind zu unserer Arbeit nötig: Unermüdliche Ausdauer und die Bereitschaft, etwas, in das man viel Zeit und Arbeit gesteckt hat, wieder wegzuwerfen.

– *Albert Einstein*

Danksagung

Die vorliegende Arbeit bildet den Abschluss des ersten Teils meines Studiums an der Universität Hamburg. Während der Anfertigung wurde ich von vielen Menschen persönlich und fachlich unterstützt, bei denen ich mich an dieser Stelle bedanken möchte.

Mein größter Dank gilt meinem fachlichen Betreuer Dirk Bade, welcher mein Interesse an dem Themengebiet *Mobile Computing* geweckt hat und mir zu jeder Zeit freundlich und kompetent zur Seite stand. Desweiteren möchte ich Prof. Dr. W. Lamersdorf und Prof. Dr. H. Federrath für die Möglichkeit der Durchführung dieser Arbeit danken.

Nicht zuletzt gebührt mein Dank meinen Freunden, für ihr entgegengebrachtes Verständnis, wenn ich wieder einmal mehr Zeit zu Hause verbracht habe als mit ihnen und meiner Freundin Cara Rosenkranz, die mich durch Korrekturlesen und liebenswerte Motivation unterstützt hat. Ebenso möchte ich mich bei meinem derzeitigen Arbeitgeber bedanken, für die flexiblen Arbeitszeiten und teilweisen Befreiungen, sodass diese Arbeit entstehen konnte.

Vielen Dank!

Inhaltsverzeichnis

Danksagung	1
1 Einleitung	1
1.1 Motivation	1
1.2 Fragestellung	2
1.3 Aufbau der Arbeit	2
2 Beispielszenarien	3
2.1 Orientierung mithilfe von Gebäudeplänen	3
2.1.1 Ohne eine unterstützende Indoor Navigations App	3
2.1.2 Mit Zuhilfenahme von Indoor Navigation	4
2.2 Clusterung in alphabetischer Reihenfolge	4
2.2.1 Ohne eine unterstützende Indoor Navigations App	5
2.2.2 Mit Zuhilfenahme von Indoor Navigation	5
2.3 Farbmarkierungen auf dem Boden	6
2.3.1 Ohne eine unterstützende Indoor Navigations App	6
2.3.2 Mit Zuhilfenahme von Indoor Navigation	7
2.4 Rückschlüsse aus den Problemen der Beispielszenarien	8
3 Grundlagen	9
3.1 Mögliche Techniken zur Indoor Navigation	9
3.1.1 Global Positioning System	9
3.1.2 Infrarot	13
3.1.3 Bluetooth	14
3.1.4 Ultraschall	15
3.1.5 Geräte-Sensoren	16
3.1.6 WLAN	19
3.1.7 Vergleich der verschiedenen Technologien	23
3.2 Unterstützung durch Maschinelle Lernen	25
3.2.1 WEKA Toolkit	27
3.2.2 Google Prediction	28
3.3 Zusammenfassung	29

4 Verwandte Arbeiten	31
4.1 Active Badge	31
4.2 RADAR	32
4.3 Magic Map	32
4.4 INSFHD	33
4.5 Active Bat	33
4.6 Cricket	34
4.7 Andere	34
4.8 Navigation mittels Maschinellern Lernen	34
4.9 Zusammenfassung	35
5 Entwurf eines Konzepts zur Indoor Positionsbestimmung	37
5.1 Anforderungsanalyse	37
5.1.1 Genauigkeit der Ortsbestimmung	38
5.1.2 Ressourcenschonende Implementierung	38
5.1.3 Frequenz der Messungen	39
5.1.4 Navigation zu einem gewählten Ziel	40
5.1.5 Datenschutz von Kontextinformationen	41
5.1.6 Verbesserung der Genauigkeit durch zusätzliche Inertialnavigation	42
5.2 Systementwurf	42
5.2.1 Anforderungen an die Infrastruktur	42
5.2.2 Ablauf und Systemkomponenten	42
5.2.3 Definition der Schnittstellen	43
5.2.4 Genutzte Technologien und Verfahren	45
5.2.5 Bedienungsablauf als Mock-up	46
5.2.6 Der Gebäudeplan	47
5.3 Zusammenfassung	49
6 Prototypische Implementierung	51
6.1 Entwicklungsumgebung Google Android	51
6.1.1 Aufbau von Android Apps	53
6.2 Das Mobilgerät	53
6.3 Die Testumgebung	55
6.4 Systemkomponenten und Positionsbestimmung	55
6.4.1 Der WifiManager	57
6.4.2 Der SensorManager	57
6.4.3 Datenspeicherung	59
6.4.4 Google Prediction	60
6.5 Zusammenfassung	64

7	Evaluation	65
7.1	Überprüfung der Anforderungen	65
7.1.1	Erreichte Genauigkeit	65
7.1.2	Ressourcenschonende Implementierung und Messungsfrequenz	67
7.1.3	Datenschutz von sensiblen Kontextinformationen	68
7.1.4	Streichungen in der Implementierung	68
7.2	Der entwickelte Prototyp	68
7.3	Nutzenanalyse	70
7.3.1	Alltagstauglichkeit des Prototyps	70
7.3.2	Fachliche Probleme des Konzepts	70
7.3.3	Technische Probleme	71
7.4	Zusammenfassung	72
8	Ausblick und Zusammenfassung	73
8.1	Ausblick	73
8.1.1	Datenbank mit POIs	73
8.1.2	Augmented Reality	74
8.1.3	Barometer	74
8.1.4	Soziale Netzwerke	74
8.1.5	Google Prediction Update-Funktion	74
8.2	Zusammenfassung	75
	Literaturverzeichnis	77
	Eidesstattliche Erklärung	85

Abbildungsverzeichnis

3.1	Positionsbestimmung mittels GPS	10
3.2	(Asynchrone) DGPS Positionsbestimmung, nach [BE05]	12
3.3	Synchronisierte Pseudolit-GPS Positionsbestimmung, nach [BE05]	13
3.4	Vereinfachte Darstellung eines Beschleunigungssensors [Sac10]	17
3.5	Darstellung eines Gyroskops, eines Kompasses und der Koordinatenachsen bei Android Smartphones [gyr12],[Goo12d]	18
3.6	Darstellung des Trilaterationsverfahrens, eigene Abb.	20
3.7	Darstellung eines möglichen Fingerprints, eigene Abb.	21
3.8	Vergleich von unterschiedlicher Hardware und Algorithmen. Zusammen- gestellt aus [GSH07]	22
3.9	Technologien zur (sehr)genauen Positionsbestimmung innerhalb und au- ßerhalb von Gebäuden [Mau09b]	24
3.10	Grundprinzip eines Maschinellen Lernsystems [XZ09]	26
5.1	Beispielgraph mit Gewichtungen und 6 Knoten, eigene Abb.	40
5.2	Systeminfrastruktur, eigene Abb.	43
5.3	Zusammenspiel der verschiedenen Komponenten, eigene Abb.	44
5.4	Zu erwartende Messfehler von Inertialnavigation und WLAN- Fingerprin- ting, einzeln und kombiniert (vereinfacht), eigene Abb.	47
5.5	Mock-up der späteren App, eigene Abb.	48
5.6	Mögliche Gebäudeplan-Umsetzungen. Fix, Aufgeteilt, Scrollbar, eigene Abb. sowie [Unk09]	48
6.1	Smartphone Marktanteil Q3'10 und Q3'11. Quelle: [Gar11b]	52
6.2	Der Aufbau des Android System Stacks. Quelle: [Goo10]	52
6.3	Der Activity Lifecycle[Goo12a]	54
6.4	Die vermessene Testumgebung am Informatikum der Uni Hamburg, eige- ne Abb.	56
6.5	Ausrichtung der verschiedenen Achsen bei Android [Goo12d]	58
6.6	Unterschiedliche Messwerte, je nach Haltung des Mobilgeräts, eigene Mes- sung	59
6.7	Messwerte während des Gehens, eigene Messung	60
7.1	Anzahl der ermittelten Genauigkeiten, eigene Messung	66

Tabellenverzeichnis

3.1	Vor- und Nachteile der verschiedenen Technologien, vgl. [LSW09].	24
5.1	Ermittelte kürzeste Pfade nach 4 Iterationen	40
6.1	Mittelwert und Median der Messwerte des Beschleunigungssensors	58
6.2	Beispielhafte Darstellung der Daten innerhalb der LinkedHashMap	60
6.3	Ausschnitt beispielhafter Trainingsdaten für Google Prediction	62
6.4	Überarbeitete Datenstruktur	62
6.5	Tabellenstruktur der Trainings CSV Datei	63

1 Einleitung

Der Markt für mobile Geräte, wie PDAs, Smartphones und Tablets, verzeichnet in den letzten Jahren ein stetiges Wachstum. Studien gehen davon aus, dass die Ausgaben für mobiles Arbeiten in den nächsten vier Jahren 1,3 Trillionen US-Dollar betragen und der Anteil mobiler Geräte an der IT-Wirtschaft auf 35% ansteigt [Pel12]. Die heutigen Geräte dienen längst nicht mehr nur der E-Mail-Kommunikation, sondern kommen mit WLAN-Funktionalität, Dualcore Prozessoren, mehreren Gigabyte Speicherplatz und verschiedenen Sensoren auf den Markt. Hierdurch scheinen auch komplexe Anwendungsszenarien realisierbar, wie beispielsweise eine Indoor Navigation mit dem eigenen Smartphone.

Diese Arbeit befasst sich mit dem Entwurf eines Konzepts für eine solche Indoor Navigation, ebenso wie der Implementierung eines Prototyps. Hierbei liegt der Schwerpunkt auf der Ortsbestimmung selbst und wie diese durch Kombination verschiedener Verfahren im Hinblick auf Genauigkeit und benötigte Ressourcen verbessert werden kann.

1.1 Motivation

Eine Besonderheit beim sogenannten Mobile Computing, dem Programmieren für Mobilgeräte, ist das Kontextorientierte Programmieren. Aufgrund der vielen Sensoren in einem Smartphone, können unterschiedlichste Daten gesammelt, gespeichert, aggregiert und verglichen werden. Der wichtigste Anwendungskontext ist hierbei der Ort des Nutzers. So ermitteln immer mehr Apps (kurz für engl. *Application*) den Aufenthaltsort um diesen zu integrieren, seien es sogenannte Soziale Netzwerke, bei denen der Nutzer seinen Kontakten mitteilen kann, wo er sich befindet, oder Umkreissuchen nach Bankautomaten, Bus- und Bahnstationen oder Restaurants. Die Angebote sind breit gefächert, sowohl von privaten Entwicklern als auch von kommerziellen Anbietern. Diese nutzen Kontextinformationen häufig für personalisierte Werbung. Diese Location Based Services (LBS) haben in der heutigen Entwicklung von mobilen Anwendungen einen festen Marktanteil eingenommen [Qub11] und sind laut dem Gartner *Hype Cycle* von 2011 bereits in dem Marktreife-Segment angekommen [Gar11c].

Während heutzutage Navigationsgeräte den Straßenatlas mehr oder weniger aus dem Auto verdrängt haben [Har07] und auch für Smartphones zahlreiche Apps zur Navigation vorhanden sind [Gle11], so endet die Navigation doch in den meisten Fällen beim Betreten eines Gebäudes. Grund hierfür ist die abreißende Verbindung zum GPS Satelliten, welcher im Allgemeinen zur Ortsbestimmung genutzt wird und direkte Sichtverbindung benötigt [Gar11a]. Doch gerade in Gebäuden, wo eine GPS-Verbindung aufgrund

der Bauweise nahezu unmöglich ist, bietet sich eine mobile Wegfindungsunterstützung an. Beispiele hierfür sind große Einkaufszentren, Messen, Universitäten oder Flughäfen.

1.2 Fragestellung

Küpper, welcher auf dem Gebiet der LBS bereits einige Forschungsergebnisse verzeichnen konnte nennt in seinem Buch verschiedene Lösungsansätze zur Indoor Navigation, welche in dieser Arbeit näher untersucht werden (vgl [Küp05]). Alle Ansätze haben eine Vielzahl gemeinsamer Probleme, welche teilweise mit der Entwicklung für mobile Geräte einhergehen. Hauptprobleme sind eine genaue Positionsbestimmung, ein beweglicher Empfänger und die vergleichsweise immernoch geringe verfügbare Rechen- und Speicherleistung eines mobilen Geräts.

Die Arbeit soll dem Leser eine Übersicht über die bisher existenten Orientierungshilfen innerhalb von Gebäuden geben und den Mehrnutzen einer hybriden Indoor Navigation verdeutlichen. Zusätzlich soll anhand des entwickelten Konzepts und Prototyps gezeigt werden, dass eine solche Navigation bereits ohne große zusätzliche Kosten realisierbar ist. Die Arbeit beschreibt Hindernisse in der Entwicklung, um deutlich zu machen an welchen Stellen eventuell Handlungs- und Verbesserungsbedarf besteht, indem die Probleme während der Implementierung dokumentiert und analysiert werden.

1.3 Aufbau der Arbeit

In dem folgenden Kapitel werden Beispielszenarien vorgestellt, in welchen eine Indoor Navigation zum Einsatz kommen kann. Es werden die Schwächen derzeitiger Hilfen verdeutlicht und die Anforderungen an eine mobile Navigation entwickelt.

In Kapitel 3 werden die Grundlagen vorhandener Navigationstechnik und ihre Eignung für den Indoor-Bereich analysiert und verglichen. Daraufhin werden in Kapitel 4 verwandte Arbeiten vorgestellt, welche unterschiedliche Methoden und Techniken nutzen. Hierbei wird sowohl auf ältere Arbeiten eingegangen, um die Grundlagen zu verdeutlichen, als auch auf aktuelle Entwicklungen.

Der Entwurf für eine geeignete Indoor Navigation wird in Kapitel 5 erstellt. Hierzu wird aus den Beispielszenarien eine Anforderungsanalyse generiert und der Systemaufbau, sowie die genutzten Technologien beschrieben. In Kapitel 6 wird die Implementierung des Prototyps vorgestellt, hierbei stehen die Besonderheiten und Herausforderungen bei der Entwicklung im Mittelpunkt.

Es folgt anschließend in Kapitel 7 eine Evaluation des Messergebnisses, welche mithilfe des Prototyps erreicht werden konnten. Daraufhin wird in Kapitel 8 ein Ausblick gegeben, über aktuelle Entwicklungen und wie diese eine Indoor Navigation ergänzen können. Kapitel 9 fasst die Ergebnisse dieser Arbeit zusammen.

2 Beispielszenarien

Schon vor Navigationsgeräten für Autos und immer noch, gab und gibt es zahlreiche Hilfsmittel - auch innerhalb von Gebäuden, die die Menschen dabei unterstützen an schwer auffindbare Ziele zu kommen. Einige dieser Mittel werden in den folgenden Szenarien genannt. Der gemeinsame Nenner der folgenden Situationen ist ein Mensch in einer ihm fremden Umgebung, eine Situation also, in der jeder schon einmal war.

2.1 Orientierung mithilfe von Gebäudeplänen

Am geläufigsten ist wohl der Gebäudeplan auf einer Karte in Fluren. Mindestens einmal pro Etage gibt es diesen Plan in größeren Gebäuden, um den Menschen die Wegfindung zum gesuchten Raum, zum nächsten Feuerlöscher oder Fahrstuhl zu erleichtern. Dieser so simpel wirkende Plan hat dennoch einige Vorteile, welche auch heutzutage bei fortschreitender Technisierung weiterhin gelten.

Gebäudepläne folgen meist einer ähnlichen Notation, sodass man nicht bei jedem Plan aufs neue längere Zeit braucht um ihn zu verstehen. Dadurch ist er von jeder Altersgruppe gleich gut zu benutzen.

Nicht von Vorteil ist jedoch, dass der Plan lediglich einen Teilbereich abbildet und nur innerhalb dieses Bereichs aushängt. Verlässt man das Gebiet, so muss ein weiterer Gebäudeplan an anderer Stelle angebracht werden, um das Anschlussgebiet abzubilden. Zusätzlich lässt er sich nicht ohne weiteres in seiner Komplexität anpassen um detaillierter oder genereller die Umgebung darzustellen. Auch mag der Plan eine gute Hilfe beim Hinweg zu einem gesuchten Raum sein, beim Rückweg jedoch, wenn man den gesuchten Raum wieder verlässt, wird in unbekanntem Gebäuden erneut ein Plan gebraucht, welchen man zunächst aufsuchen muss.

2.1.1 Ohne eine unterstützende Indoor Navigations App

Max Mustermann möchte sich an seiner neuen Universität zu einer mündlichen Prüfung anmelden. Er weiß von einem Kommilitonen, dass das zuständige Büro im Gebäude B ist. Auch die Raumnummer B-237 ist ihm bekannt. Max geht also guten Mutes zum Gebäude B. Bereits im Eingangsbereich stellt er fest, dass innerhalb des Gebäudes vier Treppenhäuser sind - eines in jeder Himmelsrichtung. Max verlässt sich auf seinen Instinkt und geht in dem nächstgelegenen Treppenhaus in den zweiten Stock. Dort angekommen muss er feststellen, dass hier bereits die 300er Räume sind, die Nummerierung scheint im

Keller zu beginnen. Max geht wieder ein Stockwerk runter und sieht sich die Raumnummern an. Hierbei stellt er fest, dass alle Nummern in seiner Umgebung kleiner als 220 sind. Genervt sucht er den nächsten Gebäudeplan, um sich zu orientieren. Er findet nur einen Plan in seiner direkten Umgebung und auf diesen sind Kaugummis geklebt und Initialen eingeritzt. Max kann seinen aktuellen Standort nicht ausfindig machen, stellt aber fest, dass das Gebäude in der Mitte geteilt ist, weil der eine Teil scheinbar nachträglich angebaut wurde. Er kann auch erkennen, dass er den falschen Aufgang genommen hat, da der alte Teil des Gebäudes die Räume von 200-220 beinhaltet und der neue Teil die weiteren. Max bleibt nichts anderes übrig, als wieder ins Erdgeschoss zu gehen und eines der neuen Treppenhäuser zu benutzen. Im ersten Stock des neuen Gebäudeabschnitts angekommen, muss er erneut einen Gebäudeplan suchen, zum Glück ist dieser in einem besseren Zustand und er kann sich orientieren. Max macht sich auf den Weg doch auf dem Flur begegnet ihm Michael, der ihn mal wieder nach den Hausaufgaben fragt. Max mag Michael nicht besonders, aber ist hilfsbereit und erklärt ihm die Aufgaben. Danach stellt Max fest, dass er vergessen hat, ob er an der nächsten Ecke links oder rechts weitergehen musste. Ärgerlich geht er zurück zum Gebäudeplan, den er glücklicherweise schnell wiederfindet, kann sich seinen Weg merken und kommt schlecht gelaunt im Prüfungsbüro an. Dort erfährt er, dass das andere Treppenhaus viel näher ist und lässt sich den Weg beschreiben. Auf dem Rückweg kommt er an keinem weiteren Gebäudeplan vorbei, doch zum Glück stimmte die Wegbeschreibung und er konnte sie sich merken.

2.1.2 Mit Zuhilfenahme von Indoor Navigation

Max betritt das Gebäude B und startet seine Indoor Navigations App. Diese erkennt, dass er sich im Eingangsbereich von Gebäude B befindet. Nun gibt Max sein Ziel, Raum B-237 ein und lässt sich den Weg anzeigen. Die App leitet ihn zum hintersten Treppenhaus. Max folgt dem vorgeschlagenen Weg und geht schnellen Schrittes in den 1. Stock. Auf der Höhe angekommen, teilt die App ihm automatisch mit, er solle aus dem Treppenhaus links abbiegen und an der zweiten Ecke habe er das Büro erreicht. In dem Moment begegnet ihm Clara, für die Max heimlich schwärmt und fragt ihn nach seinem letzten Wochenende. Als Clara weitergeht, weiß Max nicht mehr wo oben und unten ist und ob er links oder rechts gehen musste. Max schaut auf sein Smartphone und sieht die aktuelle Route. Er kommt gut gelaunt im Büro an und hat nach seiner Anmeldung auf dem Rückweg noch genug Zeit, um sich zur Mensa navigieren zu lassen und eine Cola zu kaufen.

2.2 Clusterung in alphabetischer Reihenfolge

Bibliotheken haben im Gegensatz zu Gebäudeplänen ein eigenes System, damit sich die Besucher zurechtfinden. Regale werden in Buchstabengruppen angeordnet, sodass man sich mithilfe des Alphabets zurechtfinden kann. Diese Art der Orientierung erfordert je-

doch ab einer gewissen Größe wieder einen Plan zur Übersicht und bietet sich zwar für strukturierte Orte, wie Bibliotheken oder auch Baumärkte an, nicht jedoch für größere Gebäude mit unterschiedlichen Teilabschnitten, wie es bei Flughäfen oder Einkaufszentren der Fall ist.

2.2.1 Ohne eine unterstützende Indoor Navigations App

Max möchte sich für seine bevorstehende Prüfung ein Buch ausleihen. Er kennt Autoren und Titel, weiß jedoch, dass die Bücher in der Bibliothek einem ganz eigenen Ordnungssystem folgen und nicht unbedingt nach den Namen der Autoren geordnet sind. Max betritt die Bibliothek und muss zunächst auf einen freien Platz am PC warten, um seine Suchanfrage zu starten. Als er endlich an der Reihe ist, gibt er den Autoren und den Titel seines Buches ein und bekommt auch das richtige Buch angezeigt. Er sucht auf der angezeigten Seite, welche der vielen Abkürzungen aus Nummern und Buchstaben wohl für das Regal, welche für die genaue Angabe innerhalb des Regals und welche für den Raum stehen könnte und entscheidet sich für Raum 458, Regal P, Bereich B17. Wohl wissend, dass sein Gedächtnis nicht das Beste ist, notiert er sich die Angaben und macht sich auf den Weg. Raum 458 ist schnell gefunden und so beginnt Max nach dem Regal P zu suchen. Als er den Raum betreten hat, war auf seiner linken Seite „Regal F“ zu lesen, also geht Max ein paar Regale weiter bis er meint, auf der Höhe von P angekommen zu sein. Er findet in Blickweite das Regal mit der Aufschrift P-B12 und freut sich, so schnell ans Ziel gekommen zu sein. Guter Hoffnung biegt er in den Gang von dem Regal ein und guckt sich um, ob er nicht Bereich B17 in der Nähe entdecken kann. Leider sind jedoch die Regale zu hoch, als dass er die Kennzeichnungen der anderen erkennen könnte. Also geht er zurück und ein Regal weiter in den Gang. Doch auch hier findet er seinen Bereich nicht, sondern lediglich B14-16. Einen Gang weiter beginnt jedoch schon Regal Q, sodass er sich wundert, ob ihm nicht doch ein Fehler beim Notieren der Daten passiert ist. Er beginnt stärker an sich zu zweifeln und sieht in seiner Nähe einen freien PC, an welchem er erneut sein Buch suchen möchte. Gerade als er vor dem PC steht, fällt ihm auf, dass die Regale auf der anderen Seite des Raumes weitergehen und lediglich von dem breiten Hauptgang unterbrochen sind. Die Bereiche auf der einen Seite sind alle Bereiche mit geraden Zahlen, die auf der anderen Seite für die ungeraden. So findet Max sein Buch doch noch, ohne erneut die Suche nutzen zu müssen. Er ärgert sich über eine solche Struktur, bringt das Buch zur Ausleihe und beginnt zu Hause schlecht gelaunt mit dem Lernen für seine Klausur.

2.2.2 Mit Zuhilfenahme von Indoor Navigation

Noch während Max im Bus auf dem Weg zur Bibliothek sitzt, kann er sein Smartphone aus der Tasche holen und die Indoor Navigation vorbereiten. Da er Autoren und Titel des Buches kennt, kann Max bereits den Standort des Buches raussuchen und zwar direkt in seiner Indoor Navigations App. Diese lädt den Gebäudeplan der Bibliothek auf

das Handy und zusätzlich auch das Buchregister. Max braucht nur noch sein Buch auszuwählen und bekommt auf dem Display angezeigt, wo es sich befindet. Als er die Bibliothek betritt, muss er nicht warten, bis an einem der ständig belegten PCs ein Platz frei wird, sondern kann direkt ins richtige Stockwerk gehen, welches ihm seine App anzeigt. Dort angekommen stellt Max fest, dass keinerlei Überblick möglich ist, aufgrund der hohen Regale. Er verlässt sich also weiterhin auf sein Smartphone und lässt sich Meter für Meter leiten. Als er an dem Regal mit dem Buchstaben O ankommt, schlägt ihm die App vor abzubiegen. Max wundert sich, da er doch Regal P in der Übersicht gelesen hatte und stößt sicherheitshalber manuell eine erneute Positionserfassung an, indem er auf „Aktualisieren“ drückt. Die App bestimmt seinen Standort korrekt und korrigiert die Anzeige, sodass er nun doch bei Regal P abbiegen soll. Bis auf wenige Meter führt ihn das Programm sogar zum richtigen Bereich, sodass Max bereits aus dem Augenwinkel sein Buch erblickt. Voller Vorfreude macht er sich auf den Weg zur Ausleihe und blättert auf dem Rückweg im Bus entspannt in seinem ausgeliehenen Buch.

2.3 Farbmarkierungen auf dem Boden

Hauptsächlich aus Krankenhäusern kennt man die Linien auf dem Boden, die, wenn man ihnen lange genug folgt, einen zum dazugehörigen Bereich leiten. Anders als Gebäudepläne sind diese Linien überall auf dem Weg vorhanden, sodass man jederzeit einen Anhaltspunkt hat.

Die Nachteile dieser Markierungen liegen jedoch auf der Hand. Sie sind lediglich hilfreich, solange sie auf einer stark abstrahierten Ebene leiten und eben nicht für jeden einzelnen Raum vorhanden sind. Desweiteren ist eine Anpassung an zum Beispiel einen Umzug des Bereichs sehr umständlich, da alte Markierungen entfernt, und neue erstellt werden müssen.

2.3.1 Ohne eine unterstützende Indoor Navigations App

Max hat seine Prüfung erfolgreich bestanden und ist in den wohlverdienten Urlaub geflogen. Sein billiges Flugangebot war lediglich mit einem Flugzeugwechsel am Pariser Charles de Gaulles Flughafen zu bekommen, an welchem er von einem anderen Terminal weiterfliegt, als er ankommt. Max ist nervös, da er kein Wort französisch spricht und nur 30 Minuten zwischen den Flügen Zeit hat. Ein Freund hat Max gewarnt, dass er nicht trödeln solle, da der Flughafen sehr groß sei. Kaum gelandet macht Max sich gleich auf den Weg zum nächsten Gebäudeplan, doch er findet keinen. Er weiß, dass er zum Terminal B3 muss, jedoch nicht wo er sich gerade befindet, denn die Ansage im Flugzeug und auch am Flughafen war nicht zu verstehen. Während er umherirrt fallen ihm Markierungen auf dem Boden auf. Diese sind zwar nur auf französisch beschriftet, aber Max erkennt in der Beschriftung sein Terminal B3. Er folgt der Linie eine ganze Zeit lang, bis er aus dem Augenwinkel einen Pfeil zum Terminal B3 sieht, der in die Gegenrichtung zeigt. Max ist

der Linie in die falsche Richtung gefolgt. Er wird nervös und folgt der Linie nun in die Richtung, aus der er gekommen ist. Eigentlich wollte Max noch etwas essen, doch dafür ist nun definitiv keine Zeit mehr. Max kommt an eine Stelle, wo sich die Linie aufteilt und in zwei verschiedenen Richtungen verläuft. Beide Richtungen sind unterschiedlich beschriftet, aber er kann wieder in beiden Beschriftungen sein Terminal B3 ausmachen. Max beschließt der linken Linie zu folgen, da diese neuer aussieht und er eine Abkürzung erwartet. Umso wütender wird er, als er sieht dass die Linie über eine scheinbar neu installierte Rollstuhlrampe führt und seine gewählte Linie nach diesem Umweg wieder mit der anderen Linie zusammengeführt wird. Nun muss Max sich noch mehr beeilen und fängt an zu laufen. Erschöpft erreicht er sein Terminal und ist froh ein paar Gesichter aus dem Flugzeug wiederzuerkennen. An diese muss er den Anschluss verloren haben, als er am Anfang so zielsicher in die falsche Richtung gegangen ist. Max ist erleichtert, dass sich an dem Terminal nichts geändert hat, denn während er sich so konzentrieren musste, hatte er die Flughafenansagen vollkommen ausgeblendet und hätte eventuelle Durchsagen nicht mitbekommen.

2.3.2 Mit Zuhilfenahme von Indoor Navigation

Max sitzt noch im Flugzeug im Landeanflug auf Paris und öffnet seine App. Dank des Offline-Modus kann er bereits jetzt sein Ziel, Terminal B3 eingeben und sobald er im Gebäude ist, die Navigation starten. Max hat Hunger, da ihm das Flugzeugessen nicht geschmeckt hat, also gibt er als zusätzliches Ziel seiner Route einen Baguette-Stand an. Die engen Toiletten in Flugzeugen missfallen ihm ebenso, weshalb er sich vornimmt auch eine solche im Flughafengebäude zu suchen. Als Max den Flughafen betritt schaltet er sein WLAN ein. Sofort sucht die App nach seinem aktuellen Standort und schlägt ihm Terminal C2 vor. Max erinnert sich, dass dieses auch auf seinem Ticket stand und bestätigt den Standort. Die App lädt jetzt seine vorher angelegte Route. Was Max nicht weiß: im Hintergrund berechnet die App seine Bewegungsgeschwindigkeit und in welche Richtung er geht. Dies ist nötig, weil an diesem Flughafen zu wenige Referenzdaten gesammelt wurden, um nur mit dem WLAN-Verfahren eine zuverlässige Navigation zu gewährleisten. Da Max aber seinen Startpunkt angegeben hat, können aus den wenigen Daten in Verbindung mit der Sensormessung zuverlässige Werte ermittelt werden. Max sieht, in welche Richtung er gehen muss und blickt ab und zu auf sein Handy, während er im Vorbeigehen die Läden und ihre Angebote mustert. Er kommt an mehreren Essensständen vorbei, aber ignoriert sie, weil ihm die Schlange davor zu lang ist. Max sieht auf seinem Display schließlich, dass noch weitere Stände auf seinem Weg liegen. Die Leute, die in die selbe Richtung gehen, werden immer weniger, da immer mehr von ihnen sich ein original französisches Baguette holen wollen. So kommt Max schon bald an einen Stand, wo kaum jemand ansteht und kann in Ruhe seinen Snack kaufen. Ein kurzer Blick auf sein Handy verrät ihm, dass um die Ecke ein Waschraum ist und er begibt sich dorthin. Dieser wäre von dem Weg aus nicht sichtbar gewesen, weil er sich hätte umdrehen müssen um

den Eingang zu sehen. Nun kann Max sich ohne Weiteres zum Terminal B3 begeben und kommt ruhig und entspannt in seinen Urlaub.

2.4 Rückschlüsse aus den Problemen der Beispielszenarien

Die Analyse der aktuell verfügbaren Orientierungshilfen innerhalb von Gebäuden zeigt, dass diese zwar hinreichend sind, jedoch nicht optimal die Nutzer unterstützen. Man kann sich mit ihrer Hilfe besser in einem Gebäude zurecht finden, sie schützen aber nicht vor falsch eingeschlagenen Wegen, falls man abgelenkt sein sollte. Speziell wenn man unter Zeitdruck oder aus anderen Gründen im Stress ist, wie beispielsweise an einem Flughafen, kann eine zuverlässige Indoor Navigation auf dem Smartphone, welches man in der Regel die ganze Zeit bei sich hat, sehr hilfreich sein.

Es wurde deutlich, dass zu einer genauen Indoor Navigation mithilfe von WLAN Netzen eine solide Basis an Referenzdaten benötigt wird. Fehlt diese, so kann es zu falschen Positionsannahmen führen würde. Auch wurde die Bedeutung der freien Verfügbarkeit von WLAN Netzen hervorgehoben. Es werden immer mehr Bereiche durch öffentliche Netze abgedeckt und auch private Netze sind oft stärker eingestellt als nötig, weswegen sie auch außerhalb der Gebäude ihrer Betreiber noch zu empfangen sind. Dennoch ist nicht grundsätzlich davon auszugehen, überall und zu jeder Zeit ausreichend viele Netze zu empfangen. Hierbei können zusätzliche Sensordaten helfen, welche von vielen Smartphones bereits zur Verfügung gestellt werden. Beispielsweise durch die Kombination von Beschleunigungssensor zur Schrittmessung und dem Orientierungssensor zur Richtungsbestimmung, kann eine mangelhafte WLAN Positionsbestimmung sinnvoll ergänzt werden. Die Ungenauigkeit der Sensoren und das exponentielle Wachstum der Messfehler verhindert jedoch eine Navigation alleine auf Sensordaten aufzubauen.

Die verschiedenen Techniken und Ansätze zur Positionsbestimmung werden in Kapitel 3 genauer erklärt. Außerdem wird eine Reihe von Projekten vorgestellt, bei denen unterschiedliche Gruppen versuchten eine Indoor Navigation zu verwirklichen - teilweise bereits mit großem Erfolg (vgl. Kapitel 4). Zunächst wird die Ortung außerhalb von Gebäuden erläutert, woraufhin die Probleme dargestellt werden, die auftreten, wenn man die gängigen Techniken an überdachten Orten nutzen möchte. Es folgt eine nähere Betrachtung von alternativen Ansätzen, die teilweise eine Abwandlung oder Ergänzung zu den vorher genannten bestehenden Methoden sind.

3 Grundlagen

Im vorherigen Kapitel wurde gezeigt, dass vorhandene Navigationshilfen in Gebäuden noch verbessert werden können. In diesem Teil der Arbeit werden verschiedene Technologien vorgestellt, welche zur Indoor Navigation geeignet sind oder nach Anpassung an örtliche Gegebenheiten sein könnten. Das Kapitel startet mit dem bekannten Global Positioning System GPS. Anschließend werden Infrarot, Bluetooth und Ultraschall untersucht im Hinblick auf ihre Eignung für Indoor Navigation, woraufhin eine Beschreibung der Positionsbestimmung mittels geräteeigener Sensoren und WLAN-Methoden, der verbreitetsten Technik, folgt. Am Ende des Kapitels wird auf maschinelles Lernen eingegangen, da es sich in verschiedenen Arbeiten als geeignete Unterstützung bei Navigationslösungen erwiesen hat. Anhand von zwei Beispielen wird das Vorgehen bei maschinellem Lernen vorgestellt. Zudem werden weitere Möglichkeiten genannt, welche sich gegebenenfalls zur Unterstützung einer der zuvor genannten Techniken eignen.

3.1 Mögliche Techniken zur Indoor Navigation

Indoor Navigation ist ein rasant an Bedeutung gewinnendes und aktuelles Thema (vgl. [Gar11c]). Geht es um die Navigation außerhalb von Gebäuden, so hat GPS mit weitem Abstand die größte Verbreitung. Bei der Ortung einer Person gibt es jedoch bereits Alternativen. So nutzt die Polizei bei Handyortungen beispielsweise das Mobilkommunikationsnetz GSM (=Global System for Mobile Communication) [WW00], da es eine unbemerkte, wenn auch ungenauere Abfrage des Aufenthaltsortes von außen ermöglicht, während die Positionsbestimmung mittels GPS vom Nutzer selbst angestoßen wird. Andere Technologien werden im Folgenden vorgestellt, wobei jeweils ihre Eignung für eine Indoor Navigation im Fordergrund steht.

3.1.1 Global Positioning System

GPS ist die bekannteste Methode zur Ortung außerhalb von Gebäuden und lässt sich unter gewissen Umständen auch innerhalb von Gebäuden nutzen. Der Grund, weshalb überhaupt in überdachten Bereichen Probleme mit GPS auftreten, liegt an der zu starken Störung durch Gebäude-Strukturen, sodass der Kontakt zu den Satelliten teilweise vollkommen verhindert wird [Gar11a].

Bei der GPS-Ortung werden Signale von verschiedenen Satelliten empfangen und mithilfe der mitgesendeten, genauen Aufenthaltsorte der Satelliten und eines sehr genauen

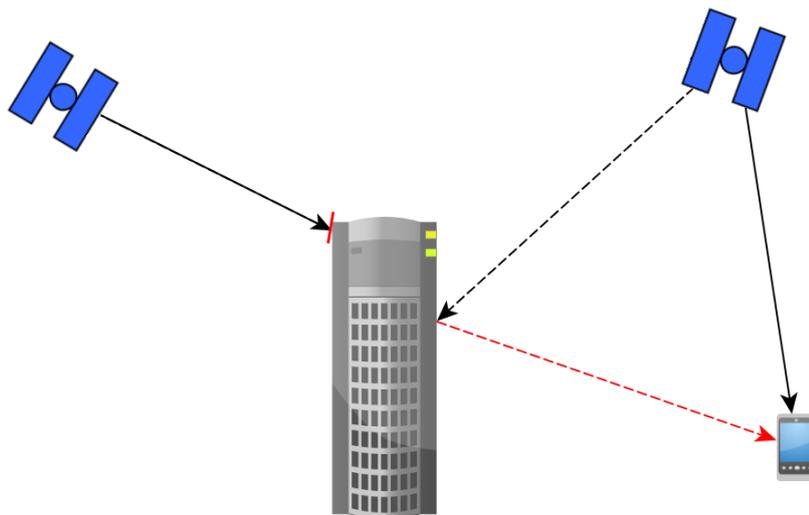


Abbildung 3.1: Positionsbestimmung mittels GPS. Ein blockiertes Signal, Multipathing und das ungestörte Signal, nach [Gar11a]

Zeitstempels die Laufzeit eines Signals bestimmt. Anhand derer lässt sich die Entfernung zu jedem einzelnen Satelliten bestimmen und durch Trilateration der Kreise mit dem Radius der einzelnen Entfernungen wiederum die eigene Position berechnen [BE05]. Abbildung 3.1 zeigt vereinfacht die Positionsbestimmung mittels GPS, sowie die Probleme, welche in dicht bebauten Gebieten auftreten können. Ein mobiles Empfangsgerät kann beispielsweise Signale nicht erhalten, wenn sich hohe Gebäude zwischen ihm und dem Satelliten befinden. Ebenso kann ein Signal an Gebäuden abgefälscht werden, weswegen es zum so genannten *Multipathing* kommt, dem Empfangen von dem gleichen Signal aus unterschiedlichen Richtungen. Ergänzend zeigt Abbildung 3.6 auf Seite 20 den allgemeinen Vorgang bei der Trilateration. Der Empfänger befindet sich an dem Ort, wo die unterschiedlichen Signale der Satelliten sich treffen. Durch Bestimmung eines jeden Senderradius der einzelnen Satelliten, kann nun die Entfernung von Empfänger und Sender ermittelt werden. Auf die auftretenden Probleme wird etwas später genauer eingegangen. Dank der mittlerweile 24 eingesetzten Satelliten und zusätzlicher Technik ist GPS im Außenbereich ausreichen genau und zuverlässig für den Alltagsgebrauch [Gar11a].

Verbessert wird die Genauigkeit von GPS durch *Differential-GPS* [Gar11a]. Sogenannte DGPS-Referenzantennen ermöglichen Korrekturberechnungen von aufgetretenen Abweichungen und gleichen so Störungen und Zeitunterschiede aus. DGPS Stationen erhalten, wie ein normaler GPS Empfänger die Signale der Satelliten, kennen jedoch zusätzliche Informationen. Stellt eine DGPS Referenz-Station auf der Erde eine Abweichung vom erwarteten Wert fest, so sendet sie den umliegenden DGPS Stationen eine Korrekturanweisung, woraufhin diese die korrigierten Daten für sich speichern und in zukünftigen Signalsendeschüben nutzen können, was auch dem einzelnen Nutzer eine höhere Genauigkeit ermöglicht. Eine elementare Rolle bei der GPS Ortung spielen hochgenaue

Zeitangaben. Die Satelliten müssen untereinander eine absolut identische Zeit nutzen, damit die Laufzeit des Signals, welches sich konstant mit Lichtgeschwindigkeit bewegt, nicht verfälscht bemessen wird. Zu diesem Zweck sind in Satelliten Atomuhren verbaut [BE05]. Hersteller von GPS Geräten geben die Ortungsgenauigkeit von DGPS mit einem bis fünf Metern an [Gar11a].

Um GPS auch innerhalb von Gebäuden nutzen zu können, wurde das sogenannte *Pseudolite-GPS* (Pseudo-Satellit-GPS) entwickelt [Ndi94]. Hierbei muss das Gebäude mit mehreren Sende- und Empfangsstationen (den Pseudoliten) ausgestattet werden, welche dann die Satelliten imitieren. Ein großes Problem bei dieser Methode ist, wie bei den echten Satelliten, die Berechnung der genauen Zeit. GPS Satelliten im All nutzen interne Atomuhren, um die höchstmögliche Genauigkeit zu ermöglichen. Diese Uhren sind jedoch zu teuer, als dass man sie in jede Bodenstation einbauen könnte. Dieses Problem kann entschärft werden, indem eine Pseudolite-Referenzstation aufgestellt wird. Dieser Station sind die Standorte der einzelnen Pseudolite-Stationen bekannt. Zusätzlich werden sogenannte *TCXO-Uhren* (Abk. für Temperature Compensated Crystal Oscillators) genutzt, welche temperaturgesteuert und mithilfe von einem Quarzkristall die Zeit ebenfalls sehr genau bestimmen können [BE05].

Pseudolite-Systeme können *synchron* und *asynchron* betrieben werden (vgl. [BE05]). Bei einem asynchronen System erhält zusätzlich zum Empfänger auch die Referenzstation gleichzeitig ein Signal, wann immer eine Pseudolite-Station ein Signal verschickt. Die Referenzstation kann nun aufgrund des bekannten Standorts und der Laufzeit des Signals die Zeitunterschiede berechnen und dem Empfänger mitteilen, sodass dieser dann die Daten korrigieren und Abweichungen ausgleichen kann. Bei dieser Systemstruktur, hat jede Pseudolite-Station ihre eigene interne Zeit, welche unterschiedlich zu den anderen Stationen sein kann. Gleichzeitig benötigt der Empfänger einen direkten Kontakt zur Referenzstation. Abbildung 3.2 zeigt den Aufbau eines asynchronen Systems, dieser gilt für DGPS und Pseudolite-GPS gleichermaßen. Dargestellt sind einzelne Satelliten, die ihr Signal senden, welches sowohl von der Referenzstation, als auch dem Empfänger selbst empfangen wird. Zusätzlich ist das Korrektursignal der Referenzstation zum Empfänger verdeutlicht.

Alternativ zu diesem Ansatz, werden in einem synchronen System die Zeiten der einzelnen Pseudolite-Stationen untereinander synchronisiert. Dies geschieht auch hier über die Referenzstation, die den Pseudolite-Stationen eine gemeinsame Zeit übermittelt, welche zuvor von einer *Master-Pseudolite-Station* übernommen wurde. Bei dieser Methode kann der Empfänger rein anhand der einzelnen Pseudolite-Stationen seine Position berechnen. Ein Kontakt mit der Referenzstation ist für ihn nicht von Nöten. Abbildung 3.3 zeigt den Aufbau eines synchronen Pseudolite-Systems. Zwischen den einzelnen Stationen und der Referenzstation wird regelmäßig die Zeit synchronisiert, sodass der Nutzer nur Signale mit synchronisierten Zeitstempeln erhält. Bei der Positionsbestimmung mithilfe von Pseudolite-GPS kann eine Genauigkeit im Zentimeterbereich erreicht werden

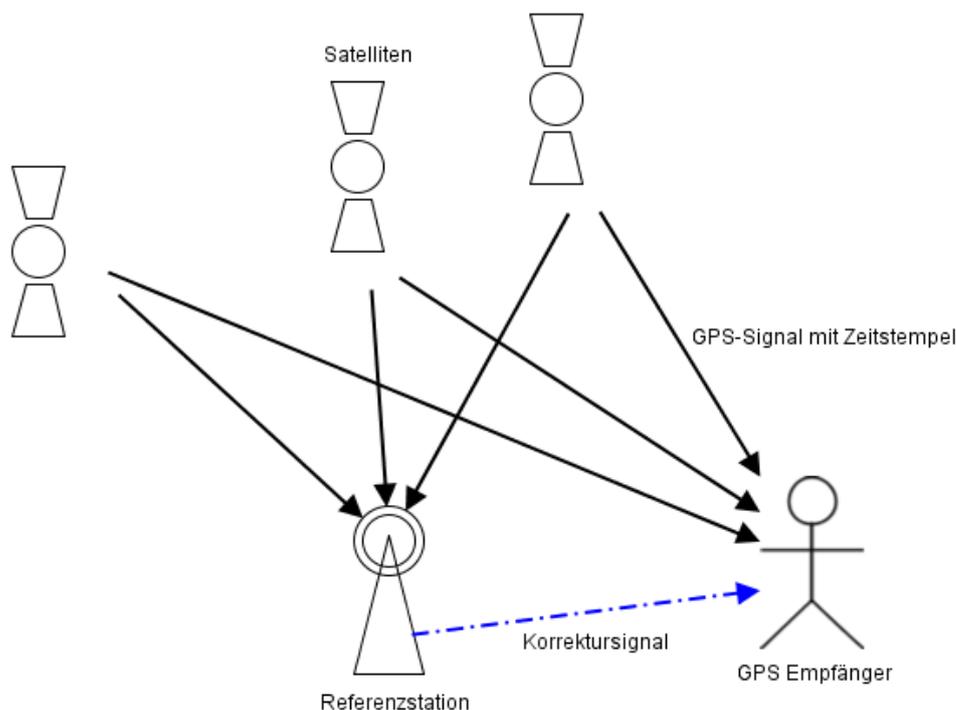


Abbildung 3.2: (Asynchrone) DGPS Positionsbestimmung, nach [BE05]

[KYJ⁺01].

Auch wenn sich GPS und Pseudolit-GPS das Problem des Multipathings aus Abbildung 3.1 teilen, so kommt es bei Pseudolit-GPS zu stärkeren Ungenauigkeiten [BE05]. Multipathing beschreibt die Tatsache, dass ein Signal durch Umwege mehrmals beim Empfänger ankommen kann. Die Ursache von Multipath-Fehlern ist, dass unterschiedliche Satelliten-Signale (teilweise auch das verfälschte eigene) einander beeinflussen, oder an Hindernissen reflektiert oder gebrochen werden. Je geringer der Unterschied zwischen dem echten und dem verfälschten Signal ist, desto schwerer fällt es dem GPS Empfänger festzustellen, ob ein erhaltenes Signal korrekt ist oder nicht. Dies führt dazu, dass aufgrund der falschen Signale die Entfernung falsch berechnet wird und somit die bestimmte Position nicht korrekt ermittelt werden kann. Umgangen oder verringert werden kann dieses Problem durch Benutzung anderer Antennen, wie zum Beispiel Helix- oder Choke-Ring-Antennen [BE05]. Nach entsprechender Ausrichtung, kann die Anzahl an abgefälschten Signalen zumindest reduziert werden.

Ein spezielles Problem von Pseudolit-GPS-Systemen ist das sogenannte *Near/Far* Problem. Wenn eine Sendestation zu nah an einer anderen steht, so beeinflusst sie das Signal der anderen Station und umgekehrt. Ein Nutzer, der in der Nähe beider Stationen steht, erhält also nur überlagerte Signale. Gleichzeitig erhält ein Empfänger ein Signal eventuell gar nicht, wenn er zu weit von einer Station entfernt ist, weswegen die Abstände zwischen den Sendestationen auch nicht beliebig groß sein dürfen. Einer von vielen Lösungsansätzen hierzu ist, die Pseudolit-Stationen abwechselnd senden zu lassen. Hierzu

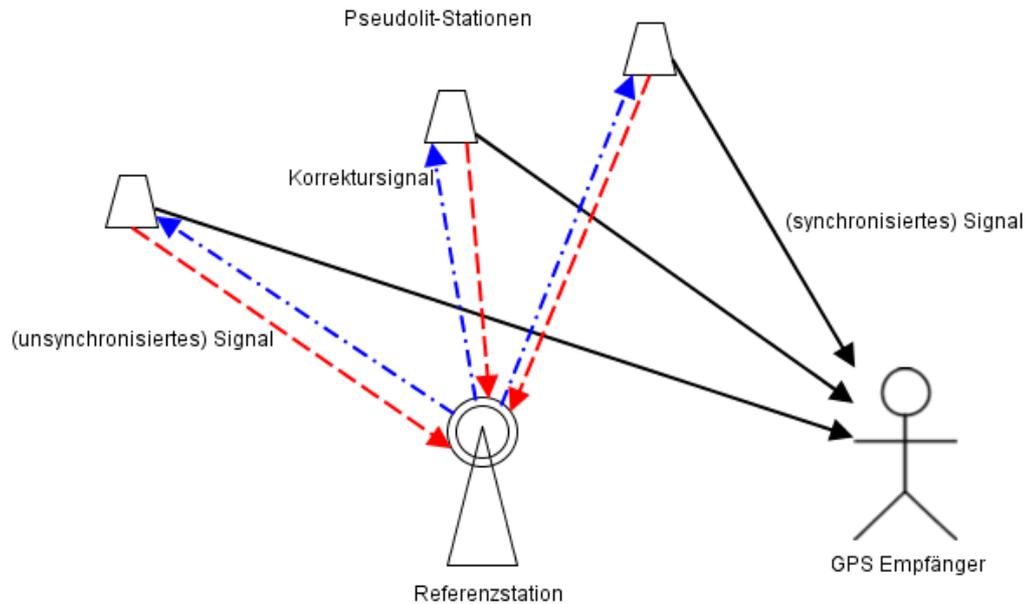


Abbildung 3.3: Synchronisierte Pseudolit-GPS Positionsbestimmung, nach [BE05]

wird jedoch wieder zusätzliche Kontroll-Hardware benötigt, um die Sendezeiten zu koordinieren, was sich wiederum in Kosten niederschlägt.

Bewertung GPS

GPS und DGPS werden außerhalb von Gebäuden häufig genutzt und auch zuverlässig. Innerhalb von Räumen lässt die Genauigkeit und Empfangsstärke jedoch schnell nach, bis hin zum totalen Verbindungsabbruch. Pseudolit-GPS bietet unter Laborbedingungen eine sehr genaue Methode zur Ortsbestimmung. Auch die Fehler die auftreten können, sind durch entsprechende Korrekturberechnungen einigermaßen zu kontrollieren. Jedoch sind die Kosten für die Einrichtung eines Pseudolit-GPS-Systems sehr hoch, weswegen es aktuell kaum Gebäude gibt, bei denen ein solches System genutzt werden könnte. Hinzu kommt, dass die Pseudolit-Stationen einzig zum Zweck der Positionsbestimmung installiert werden müssten und keinerlei sonstigen Mehrwert bringen, sodass diese Technologie sich für diese Arbeit nicht anbietet.

3.1.2 Infrarot

Infrarot Strahlung wird häufig in der Industrie bei Trocknungsprozessen genutzt, da sie eine kontaktlose Wärmeübertragung ermöglicht [BFS09]. Aber auch Heizungen oder Wärmestrahler, Nachtsichtgeräte und Wärmebildkameras nutzen Infrarot. Auch eine Indoor Navigation könnte unter Zuhilfenahme von Infrarot funktionieren, wenn auch nur mit gewissen Einschränkungen [HB00]. Grundsätzlich bietet sich Infrarot als Technologie an, da es sowohl in der Anschaffung als auch im Betrieb aufgrund des minimalen

Stromverbrauchs günstig umzusetzen ist. Hinzu kommt außerdem, dass bereits einige Smartphones Infrarot-Kommunikation unterstützen. Die Datenübertragung mittels Infrarot ist jedoch nicht sehr zuverlässig, da sie sehr anfällig gegen Störungen durch zum Beispiel Sonnenlicht oder andere Geräte ist, weswegen Datenübertragung eher über andere Kanäle stattfindet und ein Infrarot-System einzig zu dem Zweck der Navigation eingebaut und genutzt werden müsste [WM11], [BP00]. Der Aufbau eines Systems zur Indoor Navigation mittels Infrarot könnte aus mehreren Infrarot Sendeleisten bestehen, welche innerhalb eines jeden Raumes eines Gebäudes in ausreichender Höhe angebracht sind und die Referenzpunkte bilden. Eine Infrarot Kamera, welche wiederum an dem Mobilgerät angebracht sein müsste, könnte so über entsprechende Winkelberechnungen den eigenen Standort ausmachen, sofern denn der Ort der Referenzpunkte bekannt ist. Da die Zuverlässigkeit einer Verbindung über Infrarot nur mangelhaft ist, müsste auf dem Mobilgerät die Rechenarbeit erbracht werden. Verschiedene Ansätze werden in [BFS09], [HB00] und [BP00] vorgestellt.

Bewertung Infrarot

Auch wenn das Preis-Leistungs-Verhältnis einer Positionsbestimmung mittels Infrarot angemessen scheint, so überwiegen doch die negativen Punkte und hierbei insbesondere die Anfälligkeit für Störungen. Ein weiteres Defizit von Infrarot-Systemen ist, dass die zuverlässige Sendereichweite nur wenige Meter beträgt. Hinzu kommt, dass wie beim GPS, zwischen Sender und Empfänger keine Störung der direkten Sichtlinie vorhanden sein darf. Einzig eine abgeschwächte, beziehungsweise unterstützende Form der Indoor Navigation könnte mithilfe von Infrarot realisiert werden. Boogaard beschreibt in seiner Arbeit den Ansatz, eine Indoor Navigation durch Infrarot abzulösen, sobald die Sichtverbindung besteht [Boo07]. Auch wäre eine Art „Hier abbiegen“- oder „Hier umdrehen“-Meldung an den Nutzer denkbar, sobald ein Infrarotsender in Reichweite gelangt, welcher nicht zum korrekten Pfad der Navigation gehört. Innerhalb dieser Arbeit wird Infrarot aufgrund der genannten Unzulänglichkeiten jedoch nicht weiter als Technik zur Indoor Navigation verfolgt.

3.1.3 Bluetooth

Bluetooth [WMW05] ist ebenfalls eine Technologie, welche bereits von vielen Smartphones unterstützt wird. Ursprünglich wurde es genutzt um mit PDAs oder Handys untereinander Daten auszutauschen. Mittlerweile kommen immer mehr Zubehörgeräte für Handys auf den Markt, wie zum Beispiel Headsets, welche mittels Bluetooth mit dem Gerät kommunizieren. Bei Bluetooth handelt es sich, ähnlich wie bei Infrarot, um eine Kurzstreckenkommunikation, wobei Bluetoothgeräte ausschließlich im Frequenzbereich zwischen 2,402 GHz und 2,480GHz senden [BS11]. Hierdurch ist es allerdings recht störungsanfällig für WLAN Netze, Schnurlostelefone oder Mikrowellenherde. Um die Störungsanfälligkeit zu reduzieren gibt es den Ansatz des *Frequency Hopping* [WMW05], bei

welchem die Frequenzspanne in 1 MHz Stufen unterteilt wird, und alle Stufen 1600 mal pro Sekunde durchlaufen werden.

Eine Besonderheit von Bluetooth ist die asymmetrische Datenübertragung, das heißt, dass gleichzeitig gesendet und empfangen werden kann. Die Navigation mittels Bluetooth könnte ähnlich wie bei Infrarot erfolgen, indem innerhalb eines Gebäudes ausreichend Sensoren angebracht werden, welche ihren festen Standort dem Gerät mitteilen, woraufhin dieses seinen Standort bei ausreichend vielen erhaltenen Signalen errechnen könnte. Anastasi et al. beschreiben in ihrer Arbeit ein Experiment, bei welchem innerhalb eines Gebäudes mehrere Bluetooth-Sendestationen angebracht wurden und mittels einer Datenbank und eines Master-Servers die Positionsbestimmung vom eigenen, sowie die Navigation zu anderen mobilen Geräten ermöglicht wurde [ABC⁺03].

Bewertung Bluetooth

Für diese Arbeit ist Bluetooth als Indoor Navigationstechnologie uninteressant, aufgrund der durchschnittlichen Sendereichweite von ca. zehn Metern oder im Freien bis maximal 300 Meter je nach Art und Typ der Implementierung [BS11]. Hinzu kommt, dass nicht bei jedem Bluetooth-Typ eine Abfrage der Sendestärke möglich ist, ohne mit dem Sender konkret verbunden zu sein [MT05]. Außerdem unterstützen die meisten in Mobiltelefonen eingebauten Chips lediglich die Verbindung zu einer einzelnen Station, was wiederum eine Trilateration zusätzlich erschwert. Hierdurch entsteht ein unnötiger Zeitverlust beim Auf- und Abbau von Verbindungen, sowie weitere Anfälligkeiten bei der Navigation selbst. Bluetooth könnte, wie Infrarot, lediglich als unterstützende Technik genutzt werden, wird jedoch in dieser Arbeit ebenfalls nicht weiter behandelt.

3.1.4 Ultraschall

Ultraschall-Systeme sind eher unter Laborbedingungen als im Alltag geeignet, will man eine zuverlässige Positionsbestimmung erreichen. Die zwei bekanntesten Systeme sind *Active Bat* [WJH97] und *Cricket* [MIT00]. Diese haben bereits viele Verbesserungsvorschläge und Erweiterungen erfahren [HH06]. Ultraschall bietet sich aufgrund der verhältnismäßig einfach erreichbaren hohen Genauigkeit an. So erreicht das Active Bat-System eine Ortsbestimmung mit Präzision von bis zu 9 Zentimetern [HB01]. Hierbei müssen für das Active Bat-Verfahren in jedem Raum mehrere Empfangsstationen montiert und jedes zu ortende Objekt einen Sender erhalten. Die Empfangsstationen fordern in regelmäßigen Abständen die Sender auf, ein Signal zu schicken und über die Berechnung der Signal-Laufzeit wird die Entfernung mittels Lateration ermittelt. Dieser Systemaufbau ist grundsätzlich vergleichbar bei Systemen, die Ultraschall nutzen. Dennoch unterscheiden sich beispielsweise Active Bat und Cricket in der späteren Umsetzung. Während Active Bat auf fixe Basisstationen angewiesen ist und den Nutzer anhand eines mobilen *Bats* ortet, berechnet Cricket auf dem Mobilgerät selbst den eigenen Aufenthaltsort und ist nicht zwingend auf fest installierte Referenzpunkte angewiesen [MIT00]. Beide Systeme

berechnen anhand der Laufzeit eines gesendeten Signals die Entfernung und ermitteln anhand dieser den Aufenthaltsort des Nutzers, beziehungsweise des zu ortenden Gegenstands. Bei Cricket ist die Genauigkeit aufgrund der Nutzung von sich eventuell bewegenden Referenzpunkten nicht auf dem selben Niveau, wie bei Active Bat [HB01]. Der Vorteil von Cricket ist jedoch, dass auf Basis von Referenzpunkten, ob beweglich oder fix, relative Entfernungen bestimmt werden können. Dies geschieht, indem virtuelle Referenzpunkte erstellt und mit ihrer aktuellen Konstellation gespeichert werden. Später kann dann auf die verschiedenen Punkte zurückgegriffen werden [PBDT05].

Bewertung Ultraschall

Obwohl preiswert und genau gibt es auch schwerwiegende Nachteile bei der Ortung durch Ultraschall. Die bekannten Systeme sind alle lediglich für den Betrieb eines einzelnen Nutzers ausgelegt, da es sonst zu Störungen und Beeinträchtigungen kommt. Ein zentraler Aspekt ist die Sensibilität der Systeme. Diese ermöglicht zwar die verhältnismäßig sehr genaue Positionsbestimmung, alltägliche Tätigkeiten wie das Fallenlassen eines Stiftes oder das Schreiben mit einer Tastatur erzeugen jedoch bereits Ultraschall-Störgeräusche. Ultraschall Ortung mag für Fledermäuse funktionieren, in der Welt der Technik ist sie jedoch noch nicht fortgeschritten genug, um zur Navigation in unserem Kontext angewandt zu werden. Gute Ergebnisse lieferte die Ultraschall-Bildgebung bei Robotern, die ihre Umgebung erfassen konnten und entsprechend Hindernissen ausweichen [Kal96]. Die hohe Störungs-Anfälligkeit und die, mit der Installation eines kompletten Systems verbundenen, Kosten sind jedoch Gründe, weswegen Ultraschall in dieser Arbeit nicht näher betrachtet wird. Hinzu kommt, dass zusätzliche Hardware benötigt würde, welche nicht von Haus aus in Smartphones vorhanden ist.

3.1.5 Geräte-Sensoren

Da viele Smartphones bereits mit mehreren integrierten Sensoren ausgestattet sind, so wie zum Beispiel einem digitalen Kompass, einem Beschleunigungs- und Neigungssensor, ließe sich auch anhand von diesen Sensordaten der zurückgelegte Weg eines Nutzers ermitteln. Um hieraus eine Navigation abzuleiten müsste zunächst jedoch der Startpunkt bekannt sein. In dem Testgerät zu dieser Arbeit, aber auch in vielen anderen Smartphones sind bereits Sensoren zum Messen der *Beschleunigung*, der *Orientierung* und des *Magnetfeldes* um das Gerät herum eingebaut (s. Abschnitt 6.2).

Die Funktion eines Beschleunigungssensors kann man sich als eine Masse vorstellen, die zwischen vielen verschiedenen Sprungfedern aufgehängt ist [Sac10]. Diese Federn sind an einem gemeinsamen Rahmen um die Masse herum befestigt. Wirkt nun eine Kraft innerhalb des Rahmens auf die Masse ein, so werden die Federn unterschiedlich stark gedehnt. Abbildung 3.4 zeigt die verschiedenen Kräfte, die auf einen Beschleunigungssensor wirken können.

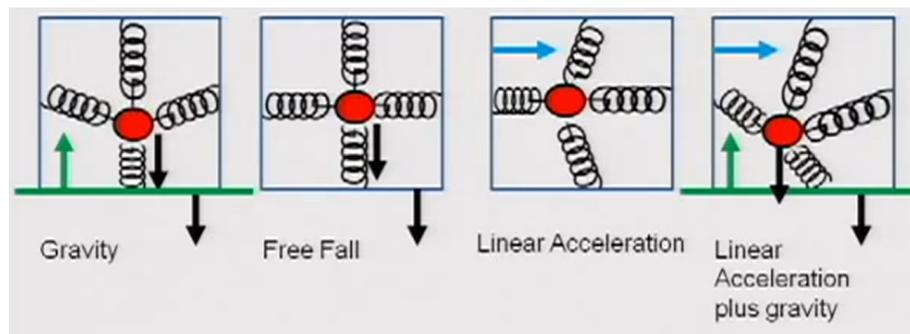


Abbildung 3.4: Vereinfachte Darstellung eines Beschleunigungssensors [Sac10]

Die Werte, welche ein Beschleunigungssensor misst, sind immer absolut und geben Auskunft über die jeweilige Kraft einer Feder, die benötigt würde, um die Masse in der Mitte zu halten. Wirkt also zum Beispiel die Erdanziehungskraft auf ein Smartphone mit Beschleunigungssensor, welches flach auf einem Tisch liegt, so zeigt der Beschleunigungssensor eine Kraft an, die der Erdanziehungskraft entspricht, da der Rahmen des Beschleunigungssensors (das Smartphone) still liegt, die Masse jedoch von der Erdanziehungskraft angezogen wird. Aus dem gleichen Grund zeigt ein Beschleunigungssensor im freien Fall eine Kraft von Null an, da alle Kräfte auf den Rahmen wie die Masse gleichermaßen wirken. Dies bedeutet ebenfalls, dass ein Beschleunigungssensor immer an die Massenträgheit gebunden ist, also immer leicht verzögert reagiert.

Ein Gyroskop, in Smartphones oft realisiert durch einen Orientierungssensor, misst die Rotation eines Gegenstandes um sich selbst, also auf den Koordinatenachsen, die nach dem Smartphone selbst ausgerichtet sind. Anders als ein Beschleunigungssensor reagiert ein Gyroskop schneller und dynamischer, sowie ohne Beeinflussung durch zum Beispiel die Erdanziehungskraft [Sac10]. Die Werte sind immer relativ, zu dem Mobilteil zu sehen - liegt das Gerät still, zeigt das Gyroskop eine Nulllinie. Abbildung 3.5 zeigt ein Gyroskop und die Ausrichtung der Koordinatenachsen bei Smartphones mit Android Betriebssystem.

Ein digitaler Kompass, wie er heutzutage in Smartphones eingebaut ist, misst die magnetischen Felder um ihn herum. Hierbei kann allerdings kein Unterschied gemacht werden, zwischen dem Magnetfeld der Erde oder dem Magnetfeld des eingebauten Akkus oder des Prozessors direkt neben dem Kompass. Hinzu kommt, dass der magnetische Nordpol nicht gleich dem geometrischen Nordpol ist (vgl. [Goo11d]).

Ist der Startpunkt des Nutzers bekannt, so kann über die Abfrage und das Zusammenspiel der Sensoren das Bewegungstempo ermittelt werden und ebenso die eingeschlagene Himmelsrichtung. Hierzu bietet der Beschleunigungssensor die Möglichkeit, die Schritte des Nutzers zu registrieren um daraus die Bewegungsgeschwindigkeit zu ermitteln. Eine kurze Bewegung in positive Richtung der in Abbildung 3.5 dargestellten Z-Achse, gefolgt von einer direkten Gegenbewegung würde somit einen Schritt anzeigen, sofern das Mobilgerät vor dem Körper des Nutzers gehalten wird. Gleichzeitig kann die

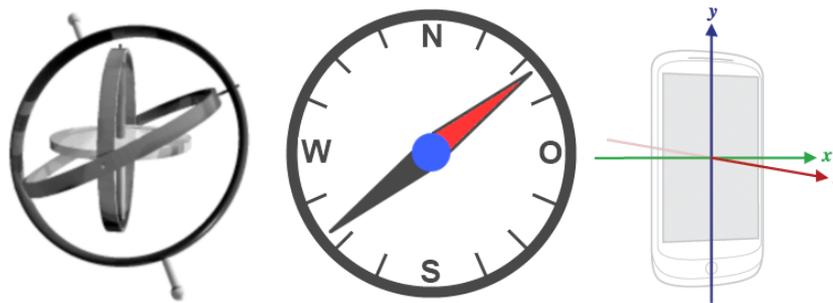


Abbildung 3.5: Darstellung eines Gyroskops, eines Kompasses und der Koordinatenachsen bei Android Smartphones [gyr12],[Goo12d]

Ausrichtung des Mobilgerätes in eine Himmelsrichtungen ermittelt werden. Mit diesen Angaben lässt sich der Pfad des Nutzers ohne zusätzliche Daten rekonstruieren. Eine solche Art der Navigation ohne externe Daten oder Geräte wird *Inertial Navigation* genannt [BF95]. Um hierdurch navigieren zu können, müssen ein möglicher Pfad definiert und dieser mit der Umgebungskarte, meist also dem Gebäudeplan, verknüpft sein.

Bewertung Sensoren

Der Vorteil der Positionsbestimmung über Sensoren ist, dass die benötigte Technik in den meisten Smartphones bereits enthalten ist. Der Nutzer ist somit auf nichts weiter angewiesen als die Karte des Gebäudes. Hinzu kommt eine große Datensicherheit, da alle Berechnungen auf dem eigenen Gerät erfolgen. Störend ist ein sehr hoher Akku-Verbrauch, da ständig auf die Sensoren-Ereignisse reagiert werden muss. Ebenso ist die Genauigkeit der einzelnen Sensoren, aufgrund der Anfälligkeit für externe Einflüsse, teilweise gestört. Eine Möglichkeit, die Genauigkeit zu verbessern ist die so genannte *Sensor Fusion* [Elm02]. Diese beschreibt die Kombination verschiedener Sensoren, um durch ihre unterschiedliche Stärken und Genauigkeiten die der jeweils anderen Sensoren auszugleichen. Im Kontext des Smartphones bedeutet dies das Kombinieren von Beschleunigungssensor und Gyroskop um die relative Beschleunigung, die auf das Gerät ausgeübt wird, zu bestimmen. Ebenso eine Verbindung von Gyroskop- und Kompassdaten, um eine korrekte Berechnung der Ausrichtung in die verschiedenen Himmelsrichtungen zu ermöglichen. Die Zuhilfenahme des Gyroskops dient hierbei der Kompensation von störenden Magnetfeldern. Die Sensor Fusion ist eigenes Forschungsgebiet, weswegen an dieser Stelle nicht weiter auf die technischen Details eingegangen werden soll. Mittlerweile gibt es Firmen, die sich auf dieses Gebiet spezialisiert haben, da die Anwendungsgebiete von Navigation über Bildstabilisation und Spiele reichen [NSM09]. Sensoren eignen sich zwar (noch) nicht als eigenständige Navigationslösung, versprechen jedoch genügend Leistung um unterstützend in einer Navigations App genutzt zu werden. In Kapitel 5 wird erneut näher auf diesen Ansatz eingegangen.

3.1.6 WLAN

Der Begriff der *Wireless Local Area Network (WLAN) Systeme* umfasst im Allgemeinen eine Reihe von drahtlosen Kommunikationsmethoden, welche in dem gemeinsamen Standard IEEE 802.11 [IEE05] erläutert werden. WLAN Netze arbeiten heutzutage hauptsächlich auf einer Frequenz von 2,4-2,4835 GHz und haben sich als Technologie zur Ortsbestimmung innerhalb von Gebäuden etablieren können. Sie sind weit verbreitet, sodass die benötigte Infrastruktur nicht erst durch teure Investitionen bereitgestellt werden muss. Vielerorts sind heutzutage kostenlose Hotspots eingerichtet, Orte, bei denen ein Nutzer sich in freie WLAN-Netze einwählen kann, oder sich zuvor registrieren muss um den Dienst zu nutzen. Zudem gibt es Internetangebote, welche eine Übersicht von diesen freien Hotspots bereitstellen. Ein bekannter Service dieser Art ist *Hotspot Locations* [Loc12], hier sind für Europa knapp 18.000 solcher WLAN Hotspots registriert, für Deutschland sind es 6.400. Auch kommerzielle Angebote gibt es, wie zum Beispiel von T-Online [TO12]. Hinzu kommen die vielen privaten Netze, wovon es besonders in Großstädten ausreichend gibt. Auch bietet mittlerweile jedes Smartphone die Technik für eine WLAN-Verbindung an.

Hauptmerkmal von WLAN im Vergleich zu anderen Funk-Netzwerken ist die hohe Datenübertragungsrate. So sind mittlerweile Übertragungsraten von 54 MBit/s mit dem gängigen Standard möglich. Eine Erweiterung der Technik liefert unter guten Bedingungen sogar 600 MBit/s [Int07]. Für diese Arbeit ist jedoch keine hohe Datenübertragungsrate von Nöten, da es vielmehr um eine genaue Bestimmung der Signalstärke geht. Auch diese wird durch heutige WLAN Implementierungen bereits ermöglicht, denn eine Sendestation schickt in der Regel zehn mal pro Sekunde ein so genanntes „*Beacon*“ mit der niedrigsten Sendeleistung. So wird sichergestellt, dass bei Erhalt eines Beacons auch eine stabile Verbindung hergestellt werden kann, welche dann mit entsprechend höherer Sendeleistung gehalten würde. Anhand dieses Beacons kann jedoch außerdem, ohne sich mit dem jeweiligen Netzwerk zu verbinden, die Signalstärke gemessen werden. Diese wird beim Empfänger als RSS (=Received Signal Strength) protokolliert und nimmt im Normalfall ab, je weiter der Empfänger vom Sender entfernt ist [GSHD07]. Die RSS wird über den Indikator RSSI angegeben, dieser ist ein negativer Wert, zwischen -10 und -96 [Wil02], wobei -10 den stärksten Empfang darstellt und -96 den schwächsten. Die Positionsbestimmung innerhalb von WLAN-Netzen kann durch mehrere verschiedene Arten der Implementierung geschehen. Im Folgenden werden die (Tri-)Lateration und das sogenannte *Fingerprinting* näher betrachtet. Heutzutage findet man häufig eine Kombination der beiden Möglichkeiten vor, dennoch lohnt sich eine klare Differenzierung der einzelnen Ansätze.

Bei der Lateration wird die Entfernung von *Access Point* (=Sender) und *Client* (=Empfänger) berechnet. Der Standort der einzelnen Accesspoints muss hierfür bekannt sein und es müssen mindestens drei Access Points erfasst werden, sodass anhand der errechneten Entfernungs-Radien ein Schnittpunkt gefunden werden kann, welcher die aktuelle

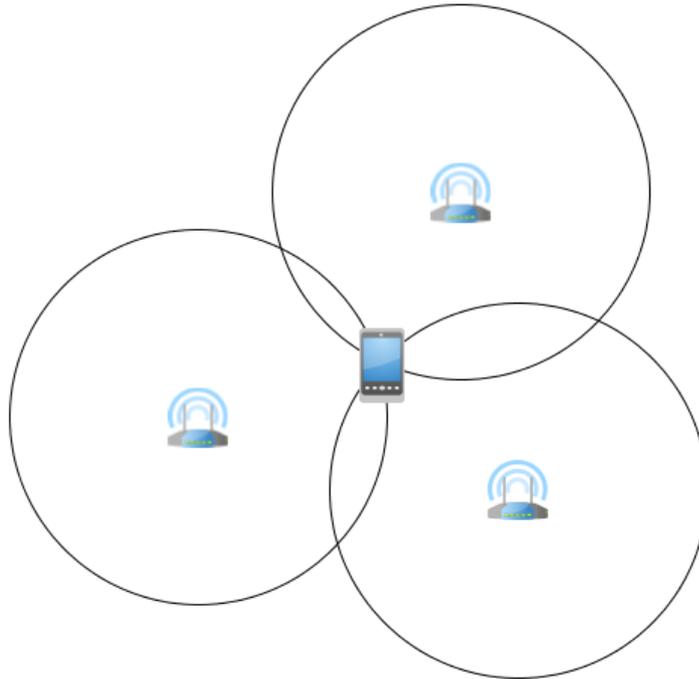


Abbildung 3.6: Darstellung des Trilaterationsverfahrens, eigene Abb.

Aufenthaltsposition eindeutig markiert. Abbildung 3.6 zeigt eine Positionsbestimmung durch Trilateration bei drei Sendestationen. Um jede einzelne Sendestation wird ein imaginärer Kreis gezogen, dessen Radius die Entfernung zum Smartphone ist. Dort, wo sich die drei Kreise treffen ist der Aufenthaltsort des Nutzers. Dieses Verfahren ist sehr anfällig für das Multipath-Problem und ebenso für andere Signalstörungen. Speziell wenn lediglich die Mindestanzahl an Access Points verfügbar ist, kommt es schnell zu großen Ungenauigkeiten, wie in ersten Untersuchungen im Rahmen dieser Arbeit festgestellt werden konnte.

Das Fingerprinting wird innerhalb von zwei Phasen durchgeführt. In der sogenannten *Offline-Phase* wird ein Raster erstellt, mit Punkten in dem Areal, in welchem man später die Navigation nutzen möchte. An jedem dieser Punkte werden Signalwerte der umliegenden WLAN-Netze gemessen und gespeichert - es wird also die RSS der unterschiedlichen Netzwerke in Reichweite, sowie die eindeutige Kennung der Netzwerke selbst gespeichert. Diese Kennung wird *Basic Service Set Identifier* (BSSID) genannt und in dem Standard [IEE05] als *MAC-Adresse* einer Station definiert. Diese Kombination aus verschiedenen BSSIDs und ihrer Signalstärke ist in der Regel einmalig, wie ein Fingerabdruck.

Hat man genügend Vergleichswerte gesammelt, so kann in der *Online-Phase* die Positionsbestimmung erfolgen. Hierzu werden aktuell gemessene Werte an einen Server geschickt, sofern die Daten extern gespeichert wurden, oder innerhalb des mobilen Gerätes selbst verarbeitet. Durch einen Algorithmus, welcher sich von Verfahren zu Verfahren

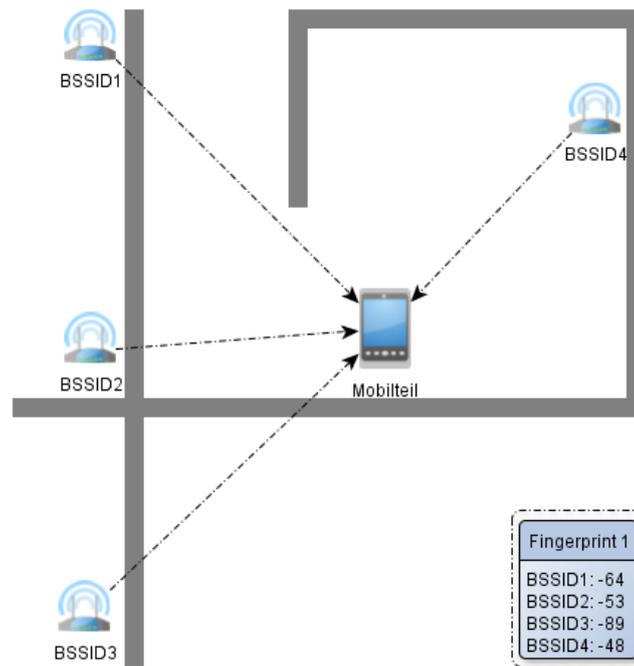


Abbildung 3.7: Darstellung eines möglichen Fingerprints, eigene Abb.

unterscheidet, werden die meisten oder besten Übereinstimmungen der Werte aus den zwei Phasen gefunden und der dazugehörige Punkt im Raster ermittelt. Der häufigste genutzte Algorithmus berechnet die euklidische Entfernung zwischen den unterschiedlichen Werten, welche üblicherweise als Vektoren gespeichert und verglichen werden. Ein geeigneter Vergleichs-Algorithmus ist jedoch die Haupt-Herausforderung bei diesem Verfahren [XLZL10]. Das Fingerprinting ist aufgrund der langen Vorbereitungszeit in der Offline-Phase ein sehr aufwändiges Verfahren, dennoch bildet es die Grundlage für die genaue Ortsbestimmung, da sich bei sehr vielen vorhandenen Vergleichsdaten kurzweilige Messfehler oder Ungenauigkeiten leicht ausgleichen lassen. Abbildung 3.7 zeigt die Erstellung eines Fingerprints, sowie die daraus resultierenden Daten. Die Signalstärken, genauer gesagt der RSSI jeden Netzes wird erfasst und zugehörig zu der BSSID gespeichert.

Eine Ortsbestimmung auf Basis von Fingerprinting ist heutzutage eine oft genutzte Technik, entsprechend wurden bereits viele Arbeiten zu dem Thema eingereicht, in welchem präzisionsverbessernde Methoden vorgestellt wurden. Eine davon ist der Kalman Filter [Cav05], welcher zudem auch in der Sensor Fusion genutzt wird.

Trilateration unterscheidet sich im Wesentlichen von dem Fingerprinting in der Vorbereitungszeit. Während das Fingerprinting viel Zeit in der Vorbereitung benötigt, jedoch wenig Zeit zur eigentlichen Berechnung der tatsächlichen Position, ist es bei der Lateration genau entgegengesetzt und die Vorbereitung ist vergleichsweise schnell geschehen, dafür die Berechnung jedoch aufwändiger.

Table 1: RSSI-values recorded by four PDA under identical conditions

BS	MP	Dell	Fujitsu	HP	MDA
AP1	MP 001	-46,23	-54,97	-44,95	-49,70
AP1	MP 002	-43,60	-49,38	-42,82	-49,12
...					
AP2	MP 001	-49,95	-52,60	-45,75	-52,05
AP2	MP 002	-50,35	-55,72	-50,00	-53,15
...					
AP3	MP 001	-58,35	-61,78	-56,53	-61,68
AP3	MP 002	-55,47	-61,63	-54,43	-63,27
...					
AP4	MP 001	-57,87	-63,42	-59,63	-63,90
AP4	MP 002	-56,53	-65,58	-58,48	-65,43
...					

Table 2: Average difference between measuring and estimated position

	test series		mean (m)	standard deviation (m)
	BAYES	Museum	tour	4,01
Seminar		exhibit	2,03	2,34
Seminar		strolling	2,67	1,40
Seminar		random	2,91	1,56
EUCLID	Museum	tour	4,16	2,51
	Seminar	exhibit	1,47	1,27
	Seminar	strolling	2,91	1,15
	Seminar	random	2,33	1,51
ISOLINE	Museum	tour	3,40	1,71
	Seminar	exhibit	2,90	0,48
	Seminar	strolling	2,00	0,58
	Seminar	random	2,00	1,04

Abbildung 3.8: Vergleich von unterschiedlicher Hardware und Algorithmen. Zusammengestellt aus [GSH07]

Bewertung WLAN

WLAN hat sich mittlerweile als Technologie zur Indoor Navigation etabliert. Nicht nur innerhalb, sondern auch außerhalb von Gebäuden wurden verschiedenste Techniken angewandt, um eine präzise Positionsbestimmung zu erreichen. Ein WLAN-System ist verhältnismäßig günstig und scheint auch nach aktuellen Vergleichen [Mau09b] ganz klar am besten geeignet für diese Arbeit. Besonders von Vorteil ist, dass keinerlei Hardware-Modifikationen vorgenommen werden müssen, um eine funktionierende Ortsbestimmung zu erhalten. Hervorzuheben ist, dass, je nach installierter Hardware, die Messwerte von Accesspoints variieren können, wie Grossmann et al. herausfanden [GSH07]. Auch der zum Vergleich genutzte Algorithmus spielt eine wichtige Rolle, was ebenfalls von Grossmann et al. in ihren Experimenten dokumentiert wurde. Verglichen wurde der „Nächste Nachbar“-Algorithmus von Euklid, die Klassifizierung nach Bayes, sowie die Isoline-Methode [GSH07]. Abbildung 3.8 zeigt den Einfluss von Hardware und genutztem Algorithmus auf die Fingerprints. Grossman et al. kommen in ihrer Arbeit zu dem Schluss, dass es nicht *den einen richtigen* Algorithmus zur Ortsbestimmung gibt, sondern dass die Algorithmen alle unterschiedliche Vor- und Nachteile haben. Letztendlich kommt es also auf ein in sich abgestimmtes System an.

Bahl und Padmanabhan [BP00] ist es gelungen, eine Genauigkeit von bis zu zwei bis drei Metern mithilfe ihres WLAN-Navigations-Systems *RADAR* zu erreichen. Sheu et al. konnten mithilfe von ihren Algorithmen die Genauigkeit auf einige Dezimeter bringen [SLTL10].

3.1.7 Vergleich der verschiedenen Technologien

Es fällt schwer, die ideale Technologie für eine Indoor Navigation zu finden. Alle in diesem Kapitel vorgestellten Ansätze haben ihre Vor- und Nachteile, weswegen sie in unterschiedlichen Kontexten alle genutzt werden können. Für diese Arbeit spielt eine zentimetergenaue Positionsbestimmung eine untergeordnete Rolle, wichtiger sind günstige Installations-, Wartungs- und Betriebskosten, sowie idealerweise ein sonstiger Mehrwert durch den Einbau der Hardware. Auch ist wichtig, das Ziel der Navigation zu betrachten, bei welcher es nicht rein um die Bestimmung des eigenen Aufenthaltsortes geht, sondern ebenso um die Verknüpfung mit einer Karte und mindestens einem anderen Zielort. Ultraschall und Infrarot scheinen eher geeignet für unbemannte Fahrzeuge, um die direkte Umgebung zu analysieren. Auch Bluetooth fällt in diese Kategorie. Pseudolite-GPS ist eine sehr genaue Möglichkeit, jedoch aufgrund der hohen Kosten nicht geeignet für diese Arbeit.

Le, Saragas und Webb haben in [LSW09] einige der zuvor genannten Technologien untersucht und gegenübergestellt. Dabei kamen auch sie zu der Feststellung, dass jedes der einzelnen Systeme spezielle Eigenschaften mit sich bringt. Tabelle 3.1 fasst ihre Ergebnisse und die im Rahmen dieser Arbeit recherchierten und erarbeiteten Beobachtungen zusammen. Dabei signalisiert ein ++ einen starken Vorteil gegenüber den anderen Verfahren, ein - - einen starken Nachteil. + und - sind weniger deutliche Vor- und Nachteile, während ein o weder Vor- noch Nachteil bedeutet. (D)GPS wäre eine gute Navigationslösung, jedoch funktioniert es nicht innerhalb von Gebäuden. Pseudolite-GPS, welches GPS ins Innere von Gebäuden portieren soll, ist extrem teuer und dennoch anfälliger für Störungen als manch andere Technologie. Für Bluetooth und Infrarot würden die geringen Kosten sprechen, jedoch würde durch den Aufbau einer entsprechenden System-Infrastruktur kein sonstiger Mehrwert entstehen, sodass diese niedrigen Kosten relativiert werden. Zusätzlich ist die Zuverlässigkeit und Genauigkeit sehr begrenzt. Bei WLAN kommen zwar ebenfalls Kosten auf einen zu, jedoch dient das System auch dem Datenaustausch. Hinzu kommt dass, bereits viele Orts mit WLAN Netzen ausgestattet sind. Die Sensorfusion bietet für sich alleine keine ausreichende Genauigkeit, wobei dafür keinerlei extra Kosten anfallen. Somit bleibt scheint eine Kombination aus WLAN- und Sensordaten die beste Wahl zu sein, um eine robuste und flächendeckende Indoor Navigation zu ermöglichen.

Abbildung 3.9 zeigt, dass die Genauigkeit der in dieser Arbeit vorgestellten Systeme keineswegs die höchste erreichbare ist. Einige Systeme aus der Grafik wurden ebenfalls im Rahmen dieser Arbeit untersucht und tauchen in dem Vergleich auf. Die anderen, höchstgenauen Systeme sind für eine Indoor Navigation zwar geeignet, jedoch nicht lohnend oder weiträumig umsetzbar [Mau09a].

	Genauigkeit	Verfügbarkeit	Kosten	Reichweite	Anfälligkeit
(D)GPS (nur außen)	+	++	o	++	+
Pseudolit-GPS	++	--	--	+	-
Bluetooth	o	-	+	+	+
Infrarot	o	-	+	-	--
Ultraschall	++	--	--	-	--
WLAN	+	++	+	+	+
Sensorfusion	o	++	++	o	+

Tabelle 3.1: Vor- und Nachteile der verschiedenen Technologien, vgl. [LSW09].

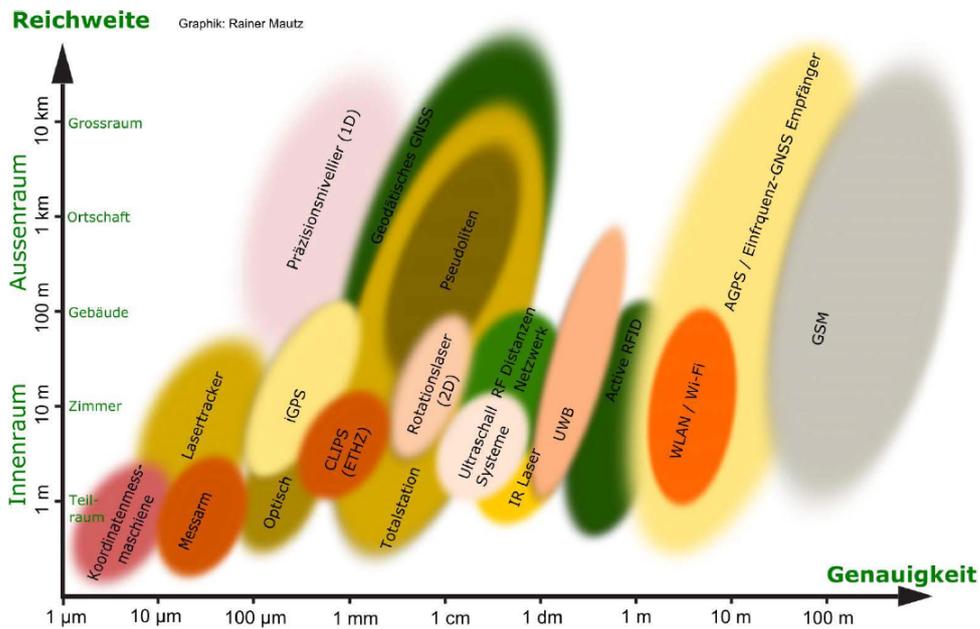


Abbildung 3.9: Technologien zur (sehr)genauen Positionsbestimmung innerhalb und außerhalb von Gebäuden [Mau09b]

3.2 Unterstützung durch Maschinelle Lernen

Maschinelles Lernen (ML) hat in der Wissenschaft verschiedenste Definitionen, in dieser Arbeit sei es in Anlehnung an mehrere gefundene Begriffserklärungen definiert als die *Verbesserung der Leistung eines Computerprogramms mit zunehmender Erfahrung, also im Verlauf der Zeit* (vgl. [Nil98, Dic10]).

Das ML ist einer der Kernforschungspunkte auf dem Gebiet der künstlichen Intelligenz. Man unterscheidet zwischen dem *überwachten Lernen* und dem *unüberwachten Lernen*. Bei ersterem ist dem Nutzer das Muster in den Daten bekannt, sodass die Ergebnisse der Lernalgorithmen direkt kontrolliert werden können. Unüberwachtes Lernen beschreibt einen Vorgang, bei welchem keine Struktur bekannt ist, sondern lediglich die Beispieldaten gegeben werden. Aus diesem Grund fällt es schwer, die Ergebnisse als *richtig* und *falsch* einzuordnen, eine Kategorisierung findet daher eher nach *hilfreich* und *nutzlos* statt. In einigen Papieren findet sich noch die Zwischenstufe des *verstärkenden Lernens*, bei welchem dem System nach einer eigenständigen Entscheidung mitgeteilt wird, ob das Ergebnis korrekt war [Job08]. Ein anschauliches Beispiel zu diesen unterschiedlichen Formen des Lernens findet sich in [RN04]. Hier wird ein Fahrschüler beschrieben, welcher in seiner ersten Fahrstunde vom Fahrlehrer alles beigebracht bekommt - es liegt überwachtes Lernen vor. In den anschließenden Stunden fährt der Fahrschüler immer eigenständiger, weswegen dies als verstärkendes Lernen aufgefasst werden kann, sobald der Fahrlehrer reagiert. Nach der Fahrschulzeit, wenn der Fahrschüler alleine im Auto fährt und auf neue, unbekannte Situationen stößt, liegt unüberwachtes Lernen vor.

Diese Arbeit nutzt eine Kombination aus überwachtem und unüberwachtem Lernen. Das Programm soll nicht eigenständig lernen, sondern wird in einer Trainingsphase gelehrt, auf bestimmte Daten zu reagieren, es soll jedoch automatisch und ohne weitere Betreuung dazu lernen und aufgrund dieses neu erlernten Wissens effizienter im Betrieb werden. Besonders ist hierbei, dass nicht lediglich die bereitgestellten Beispieldaten gespeichert und abgerufen werden, sondern eigenständig eine Gesetzmäßigkeit in den vermeintlich unstrukturierten Daten erkannt werden soll. Anhand dieser erkannten Muster soll das Programm zukünftige Daten verarbeiten und einordnen können.

Es gibt verschiedene Methoden des ML, welche alle demselben Grundmuster folgen, dargestellt in Abbildung 3.10. Über die Umgebung des Systems, oder den Systemadministrator bekommt der Lernalgorithmus neue Daten und verknüpft diese mit den in der Wissens-Datenbank (Knowledge Base) gespeicherten. Bestätigen diese neuen Daten das bisherige Weltbild, wird die Arbeit normal fortgesetzt, andernfalls muss die Arbeitsweise des ML-Systems angepasst werden.

Wie zu sehen ist, bilden die Daten aus der Lernphase das Grundgerüst eines ML-Systems, weswegen sie gründlichst ausgesucht und aufbereitet werden müssen. Es gibt verschiedene Möglichkeiten die Daten zu strukturieren. Beispielsweise können sie im *Comma-Separated Values* (CSV) Format gespeichert werden, ebenso können sie aber auch als Vektoren oder logische Schlussfolgerungen und Formeln vorliegen. Bei der Auswahl

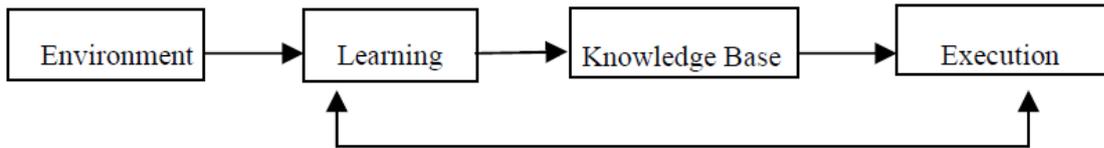


Abbildung 3.10: Grundprinzip eines Maschinellen Lernsystems [XZ09]

des Formats spielen vier Aspekte eine wesentliche Rolle [XZ09]. Die Daten müssen ausdrucksstark sein, leicht nachzuvollziehen und editierbar, sowie ohne Probleme erweiterbar. Im Entwurfskapitel dieser Arbeit wird deutlich, dass nicht immer alle dieser Kriterien gleichzeitig erfüllt werden können, dennoch ist es wichtig, dem Anspruch weitestgehend zu folgen.

Bekommt ein ML-System zusätzlich zu den Daten bereits einen Kontext beigebracht, so kann es zwischen relevanten und irrelevanten Daten unterscheiden und letztere verwerfen, um nicht unnötig Speicherkapazitäten zu beanspruchen. Ein Problem bei zugrunde liegenden Daten ist häufig die Unvollständigkeit, weswegen nicht auf Anhieb zu erkennen ist, ob das System im späteren Gebrauch zuverlässiger wird oder aber sogar im Gegenteil, ein eventueller Fehler sich fortsetzt. Wie die Daten gespeichert werden und welche Schlüsse aus ihnen gezogen werden, ist von außerhalb des Systems oft nicht erkennbar, sodass lediglich durch Testen herausgefunden werden kann, ob das System wie erwartet funktioniert. Sind die Ergebnisse jedoch nicht im Vorfeld eindeutig, so ist auch dies ein großes Problem. Wie gut ein System zu testen ist, hängt ebenso von der Systemstruktur selbst ab. So beschreiben Xue und Zhu die Komplexität, die Transparenz und Feedbackmöglichkeit des Systems als entscheidend [XZ09]. Desweiteren beschreiben Xue und Zhu unterschiedliche ML-Methoden, von denen an dieser Stelle einige für diese Arbeit relevante Methoden vorgestellt seien.

Das *Explanation-Based Learning* (EBL) beschreibt ein Vorgehen, bei welchem dem ML-System zunächst ein Entscheidungsmuster beigebracht wird, woraufhin später die Daten folgen, welche nach dem Muster behandelt werden sollen. Ein solches Vorgehen, bei dem vom Allgemeinen auf das Besondere geschlossen wird, wird *deduktives Lernen* genannt. Diese Methode wird häufig zu Systemoptimierungen genutzt, da bei diesen das Ergebnis klar definiert, der Weg dorthin aber unklar ist [XZ09].

Komplexer sind dagegen das *Erkennen von Analogien* was zum *induktiven Lernen* gezählt werden kann. Hierbei gilt es, vom Besonderen zum Allgemeinen zu schließen. Während bei dem Analogie-Lernen dem System noch ein Konzept beigebracht wird, nach welchem es arbeiten und auf Basis dessen es die Vergleiche der Daten durchführen soll, wird bei einigen induktiven Lernverfahren ganz auf die Intelligenz des Systems vertraut. Das System bekommt Daten und soll diese eigenständig neu arrangieren, um sinnvolle und hilfreiche Schlüsse aus diesen zu ziehen. Das induktive Lernen ist diejenige Methode, bei welcher am wenigsten Beeinflussung von außen auftritt, wenn das System erst einmal gestartet wurde. Aus diesem Grund ist es beim induktiven Lernen möglich, dass ein

System ursprünglich getroffene Annahmen auch wieder verwirft, während es im Unterschied dazu bei *deduktiven* Verfahren üblich ist, getroffene Annahmen in den Bereich der Fakten zu integrieren, um sie bei allen neuen Abfragen einzubeziehen. Wird beim induktiven Lernen eine falsche Annahme genutzt, um andere Annahmen zu bilden, so müssen alle Folge-Annahmen erneut geprüft werden, zu dem Zeitpunkt wo sich die Ursprungs-Annahme als falsch herausgestellt hat.

Als Beispiel hierzu stelle man sich ein System vor, welchem Datensätze über unterschiedliche Bücher gegeben werden. Jeder der Einträge beinhaltet einen Autoren, das Jahr der Veröffentlichung, sowie den Verlag und den Titel. Stimmen nun bei allen Büchern der Verlag und das Jahr überein, so könnte das autonom erstellte Schema eines Buches innerhalb des Systems folgendermaßen aussehen: $\text{BUCH}=\{(\text{Beliebiger Text}), (\text{Beliebiger Text}), 2004, \text{Carlsen}\}$. Ein Buch, welches dem System vorgelegt wird und aus einem anderen Jahr ist, würde nun das ursprüngliche Schema als falsch belegen und zu folgender Struktur führen: $\text{BUCH}=\{(\text{Beliebiger Text}), (\text{Beliebiger Text}), (\text{Beliebige Zahl}), \text{Carlsen}\}$.

Aufgrund der vielen verschiedenen Methoden und Unterscheidungen bei ML, gibt es Implementierungen von ML-Systemen, bei denen dem Nutzer gleich eine Reihe von Lernalgorithmen zur Verfügung stehen. So sind häufig genutzte Verfahren zur Klassifizierung zum Beispiel, Entscheidungsbäume und -Listen, Support Vector Machines oder auch Bayesche Netze. Ein freiverfügbares Toolkit, welches neben diesen Algorithmen noch zahlreiche andere bereitstellt wird im folgenden Abschnitt vorgestellt. Es handelt sich um das weit verbreitete WEKA-Toolkit, welches hauptsächlich zu Data Mining Aufgaben entwickelt wurde. Im anschließenden Abschnitt wird ein neues Application Programming Interface (API) der Firma Google vorgestellt, welche ebenfalls die meisten der gängigen ML-Algorithmen beinhalten soll.

3.2.1 WEKA Toolkit

Das *Waikato Environment for Knowledge Analysis* (WEKA-Toolkit) [WFH11] wurde 1992 an der Universität von Waikato, Neuseeland entwickelt. Es ist ein Softwaretool, welches unterschiedliche Lernalgorithmen zum ML bereitstellt, sowie zusätzlich Techniken zur Datenanalyse im Rahmen des *Data Mining*. WEKA bietet die bekanntesten maschinellen Lernalgorithmen an, darunter viele Methoden zur Klassifizierung von Daten, wie zum Beispiel die Bayes-Klassifikation und Entscheidungsbäume, künstliche neuronale Netze und einige mehr [HFH⁺09].

Ziel der Entwickler war es, vor allem Fachleuten komplexer Themengebiete ein Werkzeug an die Hand zu geben, mit welchem sie Daten aus der Praxis sinnvoll verarbeiten können, ohne sich Gedanken über die ML-Algorithmen selbst machen zu müssen [HFH⁺09]. Erst in zweiter Instanz richtet sich WEKA an Experten aus dem ML Bereich, diese jedoch sollten ohne Fachwissen von Data Mining neue Algorithmen implementieren können, innerhalb von WEKA. Im Mittelpunkt steht dennoch die Anwendung mit

den angebotenen Hilfsmitteln zur Datenverarbeitung, weswegen diese teilweise lediglich einfache Klassifizierungen vornehmen, was jedoch für eine Vielzahl von Anwendungsfällen ausreicht [HFH⁺09]. Zu der Zeit als WEKA entwickelt wurde, wurde aus Sicht von Witten et al. mehr Zeit auf die Entwicklung neuer, leistungsfähiger Algorithmen investiert, als aus den bekannten einen Nutzen zu ziehen. Diesem Trend wollten die Entwickler entgegenwirken und die komplexe Technik hinter den Algorithmen vor dem Endnutzer verbergen [WFH11].

In der Zwischenzeit wurde WEKA mehrfach überarbeitet und ergänzt, sodass zu den ursprünglich acht Lernalgorithmen viele unter anderem Multi-Instanz fähige Algorithmen hinzugekommen sind, sowie *Preprocessing Filter* und unterschiedliche User Interfaces ergänzt wurden [HFH⁺09].

WEKA ist in *Java* [Ora12] geschrieben und unter der *General Public License* [GNU91] frei erhältlich, weswegen es auf den meisten Geräten nutzbar ist, eine solide Basis an Anwendern in der ganzen Welt hat und stetig weiterentwickelt und gewartet wird.

3.2.2 Google Prediction

Google Prediction [Goo11c] ist ein von Google angebotener Service, welcher verschiedene maschinelle Lernalgorithmen in sich vereint und aus ihnen den für die jeweilige Situation am besten passenden wählt. Der Nutzer braucht sich um keinerlei Implementierung zu kümmern, sondern kann Google Prediction als eine *Black Box* nutzen.

Zunächst wird Google Prediction mittels Anfangsdaten *trainiert*, also die Lernalgorithmen mit einen Grundbestand eingestellt, auf dessen Basis die späteren Daten verarbeitet werden. Diese Daten müssen in Googles *Cloud-Storage* [Goo11c] gespeichert werden, damit über das Prediction API auf sie zugegriffen werden kann. Innerhalb der Trainingsphase erkennen die Algorithmen dann Muster und Gesetzmäßigkeiten der Trainingsdaten und wenden diese später auf die Betriebsdaten an. Der Algorithmus, welcher am besten geeignet scheint, übernimmt die späteren Berechnungen aller anderen Daten. Von Google vorgeschlagene Anwendungsszenarien für Google Prediction sind zum Beispiel eine Spracherkennung, die analysiert in welcher Sprache ein Text geschrieben ist, oder ein Spamfilter in E-Mail-Postfächern. Auch Film- und Musikempfehlungen sollen möglich sein, bis hin zu Stimmungsanalysen der Autoren von Nachrichten in sozialen Netzwerken [Goo11a].

Bei den verschiedenen Technologien zur Indoor Navigation, welche in diesem Kapitel vorgestellt wurden, bietet es sich an, wann immer Daten aus durchgeführten Messungen zu verarbeiten sind, hierzu maschinelle Lernverfahren zu nutzen. Der daraus erwachsende Vorteil ist, dass teilweise sehr komplexe Datensätze, verhältnismäßig leicht verarbeitet werden können, sobald eine gewisse Struktur bekannt ist oder von den ML-Algorithmen erkannt wird. In Kapitel 5 wird erneut auf die Kombination aus ML und anderen Technologien eingegangen.

3.3 Zusammenfassung

In diesem Kapitel wurden die bekanntesten Technologien zur Indoor Positionsbestimmung vorgestellt. Es wurde sowohl auf die frühen Technologien eingegangen, sowie auf neueste Entwicklungen. Die Idee WLAN-Netze zu nutzen sticht hierbei hervor, da sie relativ früh zu Beginn der Forschung als interessant galt und auch heutzutage noch in einem Großteil anderer Arbeiten behandelt wird. Zusätzlich wurde das weltweit genutzte GPS vorgestellt und erläutert, welche Schwächen es innerhalb von Gebäuden hat, sodass nach einer Alternative gesucht werden muss. Auch der Ansatz, GPS innerhalb von Gebäuden in Form von Pseudolite-GPS zu nutzen wurde vorgestellt. Desweiteren wurden andere Technologien aus dem Bereich der Robotik erwähnt, nämlich Infrarot, Bluetooth und Ultraschall. Es wurde deutlich, dass bei entsprechendem Aufwand eine Positionsbestimmung mit einer Genauigkeit von unter einem Zentimeter bereits möglich ist. Abschließend wurde in diesem Kapitel auf die Methode des maschinellen Lernens eingegangen.

Im nächsten Kapitel werden verwandte Arbeiten vorgestellt, welche jeweils nach der genutzten Technologie ausgewählt wurden, beziehungsweise wegen ihrer Bedeutung für die weitere Entwicklung auf dem Gebiet der Indoor Navigation. Auch die Relevanz für diese Arbeit spielte bei der Auswahl eine Rolle, sodass nicht nur bekannte Autoren zu finden sind, sondern auch zum Beispiel Bachelorarbeiten von anderen Studenten, aufgrund der Nähe ihrer Ergebnisse zu dieser Arbeit.

4 Verwandte Arbeiten

Indoor Navigation ist keineswegs ein neues Thema. Bereits 1989 haben die ersten Vorreiter angefangen an Lösungen für die Ortung von Personen und Gegenständen zu arbeiten. Damals wurde Infrarot-Technologie im Active Badge-Verfahren genutzt, wohingegen heutzutage WLAN die dominierende Technologie ist.

In diesem Kapitel sollen einige der grundlegenden Ansätze und Arbeiten zur Indoor Navigation und Positionsbestimmung vorgestellt werden. Begonnen wird mit dem Active-Badge-Verfahren als dem ersten System zur Positionsbestimmung innerhalb von Gebäuden, desweiteren werden das Projekt RADAR von Microsoft vorgestellt sowie einige kleinere Projekte, welche ähnliche Ansprüche und Ziele haben wie diese Arbeit. Beendet wird das Kapitel mit Beispielen von maschinellen Lernverfahren und wie diese die Navigation innerhalb von Gebäuden unterstützen können.

4.1 Active Badge

Das *Active Badge*-Verfahren [WHFaG92] wurde 1989 an der Cambridge Universität entwickelt. Es beschreibt den Ansatz, Gegenstände und Personen über Infrarot zu orten. Active-Badge ist das erste Indoor-Ortungsverfahren gewesen und kann somit als grundlegend für alle folgenden Verfahren angesehen werden. Innerhalb der Räume, an welchen man interessiert ist, werden Infrarotstationen angebracht und jeder Gegenstand oder Mensch, dessen Aufenthaltsort bestimmt werden soll, bekommt einen Active Badge-Sender. Dieser Sender schickt im Zehn-Sekunden-Takt ein Signal zu den erreichbaren Basisstationen, welche ihre Daten wiederum an einen zentralen Server senden. Anhand dieser Information kann der Aufenthaltsort des Senders ermittelt und im System gespeichert werden.

Das Active Badge-Verfahren eignet sich aus mehreren Gründen nicht für diese Arbeit. Wie bereits im Abschnitt 3.1.2 erläutert, sind die Nachteile von Infrarot-Ortung die geringe Reichweite und die Anfälligkeit für Störungen, zum Beispiel bei direkter Sonneneinstrahlung. Hinzu kommt die Installation der Rechneinheit, um aus den erhaltenen Signalen die Entfernungen zu berechnen. Zusätzlich müsste der Server die berechnete Position wieder zurück an das Mobilgerät senden, da es in dieser Arbeit nicht um die Ortung eines Objektes von außen geht, sondern vielmehr um die Bestimmung der eigenen Position.

4.2 RADAR

RADAR [BP00] ist ein System um Nutzer innerhalb von Gebäuden lokalisieren und navigieren zu können. Es wurde 2000 von Bahl und Padmanabhan im Rahmen von Microsoft Research entwickelt und ist das erste WLAN-basierte Navigationssystem [Mic00]. Der treibende Gedanke von Bahl und Padmanabhan war, bestehende System-Infrastrukturen zu nutzen, anstatt neue aufwändige Technik zu montieren. Das RADAR-Verfahren nutzt WLAN-Fingerprinting (vgl. Abschnitt 3.1.6) und wird daher von den Autoren in zwei unterschiedliche Phasen unterteilt. Die *off-line* Analyse und die *real-time* Phase. Innerhalb der *off-line* Phase werden Fingerprints erstellt, und innerhalb der *real-time* Phase wird die aktuelle Position mittels Triangulation ermittelt und die gespeicherten Daten mit den aktuell gemessenen verglichen, um zusätzliche Genauigkeit zu erhalten. Besonders an dem Verfahren ist, dass dieses System eine Kombination aus mathematischer Triangulation und empirischen Fingerprint-Daten nutzt. Es eignet sich dennoch nicht für diese Arbeit, da zu der WLAN-Signal-Messung eine Netzwerkkarte im Mobilteil installiert sein muss, welche zusätzlich zum Signal selbst auch die Signal-Störungen (signal-to-noise-ratio) verarbeiten kann [HB00].

4.3 Magic Map

Magic Map [IHS04] greift die bewährten Methoden von Microsofts RADAR auf und verbessert sie mit aktuellen Entwicklungen und neuen Algorithmen. Außerdem wird es sinnvoll ergänzt durch die Einbettung eines Mehrbenutzer Services. *Magic Map* wurde 2004 an der Humboldt-Universität zu Berlin entwickelt. Es vereint die Berechnung des Aufenthaltsortes durch Trilateration und bezieht typische Bewegungsmuster und Signalhindernissen mit in die Berechnungen ein.

Die Besonderheit ist der Austausch von Referenzpunkten durch alle Teilnehmer über ein Peer-to-Peer Netzwerk. So können ständig Korrekturen von Fehlern und Ungenauigkeiten vorgenommen werden, welche aufgrund von schwankenden Sende- und Empfangsleistungen vorkommen können. Ebenfalls wird die Kalibrierungszeit durch diesen Austausch erheblich reduziert. Hinzu kommt, dass nicht nur der Nutzer selbst geortet und auf einer Karte angezeigt wird, sondern dass man innerhalb des App-Netzwerks auch seinen Standort freigeben und anderen Nutzern auf deren Karte anzeigen lassen kann.

Magic Map ist ein durchdachtes und ausgereiftes System, welches jedoch eine leistungsstarke System-Infrastruktur benötigt, sowie eine aktive Community, um aus den genannten Punkten einen Nutzen zu ziehen.

4.4 INSFHD

Indoor Navigation System for Handheld Devices (INSFHD) ist ein Navigationssystem, welches am Worcester Polytechnic Institute im Rahmen einer Masterarbeit dreier Studenten entstand [LSW09]. Es verbindet die Daten aus dem WLAN-Adapter eines Smartphones, dem Beschleunigungssensor und dem Kompass, um die Position des Nutzers zu bestimmen. Anschließend wird der optimale Pfad vom Aufenthaltsort des Nutzers zu dessen gewünschtem Ziel berechnet und ausgegeben. Innerhalb ihrer Tests kamen die Entwickler und Autoren Le, Saragas und Webb auf eine Genauigkeit von ca. 1 Meter.

INSFHD bietet ein stabiles und genaues System zur Positionsbestimmung innerhalb von Gebäuden. Die Schwäche des Systems sind die ständigen Korrekturen der berechneten Position. Diese erfolgt in drei Iterationen, was entsprechende Rechenleistung benötigt. Zunächst wird der grobe zurückgelegte Pfad der Nutzers anhand von WLAN-Fingerprints ermittelt, anschließend wird dieser Pfad bereinigt, unter der Annahme, dass der Nutzer sich stetig in dieselbe Richtung fortbewegt. Danach werden die Werte auf zuvor definierte plausible Punkte angepasst, was den letztendlichen berechneten Pfad ergibt.

Auch wenn die Genauigkeit von INSFHD gut ist, so werden doch eine Reihe von Annahmen getroffen, welche nicht immer mit dem tatsächlichen Verhalten der Nutzer vereinbar sind. Zusätzlich wird über die mehrfachen Korrekturberechnungen des ursprünglich ermittelten Pfades die Akkuleistung des Mobilgeräts aufgebraucht. Eine weitere Vereinfachung der Autoren ist die Bewegungsgeschwindigkeit eines Nutzers. Diese ist konstant im System definiert, was bei unterschiedlichen Altersgruppen, sowie in unterschiedlichen Umgebungen mit Hindernissen, wie beispielsweise anderen Personen, zu Problemen führen dürfte.

4.5 Active Bat

Active Bat nutzt, ebenso wie Fledermäuse, Ultraschallwellen zur Berechnung von Entfernungen. Hierbei werden in jedem Raum mehrere Empfangsstationen montiert und jedes Objekt erhält einen Sender. Die Empfangsstationen fordern in regelmäßigen Abständen die Sender auf, ein Signal zu schicken und über die Berechnung der Signal-Laufzeit wird die Entfernung mittels Lateration ermittelt. Die Genauigkeit bei dieser Methode liegt bei neun Zentimetern, bei guter Abdeckung und wenig Störungen. Nachteile bei diesem Verfahren sind entsprechend die hohen Kosten, da Ultraschallgeräte wesentlich teurer sind als zum Beispiel die für Infrarot. Auch kommen hier Administrationskosten für die Rechner-Infrastruktur hinzu.

4.6 Cricket

Cricket [MIT00] ist ein am Massachusetts Institute of Technology (MIT) entwickeltes weiteres System für die Ortung mithilfe von Ultraschall, jedoch wird Cricket zusätzlich durch Funk-Kommunikation erweitert. Außerdem finden bei diesem Ansatz alle Ortsberechnungen auf Senderseite statt, sodass zwar Informationssicherheit über den Aufenthaltsort gewährleistet ist, jedoch aufgrund von gesendeten Ultraschallsignalen, Funksignalen und anschließender Berechnung der Stromverbrauch auf Seiten des Senders extrem hoch ist. Daher eignet sich dieser Ansatz nur bedingt für mobile Geräte.

4.7 Andere

Weiterhin zu erwähnen sind SpotOn, Ubisense oder Smart Floor als bekannte Systeme, darüber hinaus gibt es viele mehr, welche im Rahmen kleinerer Arbeiten entstanden. Eine gute Übersicht findet sich in [HLCY10] oder [HB01, Tabelle 1].

Auch kommerzielle Anbieter haben bereits den Markt der Indoor Navigation zu erschließen begonnen. Bekanntester und größter Anbieter solcher Angebote ist das Unternehmen *Skyhook* [Sky12], das seine Leistungen frei verkauft und unter anderem von großen Firmen wie Intel, Sony, Samsung oder Apple genutzt wird. Auch Google hat bereits ein eigenes System erstellt, welches jedoch noch nur innerhalb der Google-Dienste genutzt wird und nicht zur freien Verfügung steht [Goo11b].

4.8 Navigation mittels Maschinellen Lernen

Auf dem Gebiet des maschinellen Lernens (ML) wird seit einigen Jahrzehnten aktiv geforscht. Ursprünglich war das ML nur ein Untergebiet der künstlichen Intelligenz, mittlerweile hat es sich zu einem eigenen Forschungszweig entwickelt. In ihrem Artikel [BB05] zeigen Brunato und Battiti bereits einen Ansatz, wie durch die Kombination aus WLAN Fingerprints und maschinellen Lernverfahren eine genaue Positionsbestimmung erreicht werden kann. Hierbei nahmen die Autoren an unterschiedlichen Stellen innerhalb des Testgebäudes Messungen vor und lernten verschiedene Algorithmen mit diesen Referenzdaten an. Sie verglichen den Algorithmus des *Weighted k Nearest Neighbors*, den *Bayesian approach*, ein *Multi-Layer Perceptron* und den Ansatz einer *Support Vector Machine*. Unter anderem kamen sie zu der Beobachtung, dass das Bayessche Modell bereits ab einer Anzahl von zehn Referenzmessungen ausreichend genaue Ergebnisse lieferte, wohingegen andere Modelle mehr als 100 Referenzmessungen benötigten um auf ähnliche Genauigkeit zu kommen. Im Gegenzug benötigte das Bayessche Modell jedoch am meisten Zeit und Zusatzinformationen, weswegen die anderen Algorithmen ihnen geeigneter schienen. Auch vermuten sie, dass eine höhere Dichte an Accesspoints die Genauigkeit aller verglichenen Algorithmen stärken würde [BB05, Kapitel 7].

Anagnostopoulos et al. haben 2009 einen Artikel veröffentlicht, in welchem sie ein Verfahren beschreiben, um die Position eines Nutzers mithilfe von maschinellen Lernverfahren und Nutzerprofilen vorherzusagen [AAH⁺09]. Auch sie verglichen unterschiedliche Algorithmen, wie den *Ensemble-Learning Algorithm*, *Entscheidungsbäume* oder ebenso den Bayesschen Ansatz. Für die Nutzerprofile berücksichtigten sie das bisherige Verhalten und den Grad an zufälligen Abweichungen von ursprünglichen Routen. Zusätzlich schlagen die Autoren vor, weitere ergänzende Kontextinformationen in die Berechnung aufzunehmen, wie zum Beispiel die Uhrzeit, das Bewegungstempo, Begleiter des Nutzers oder den Anwendungsbezug.

Ebenso lieferten Manley und Deogun einen Entwurf für *Smart Homes* [MD07], intelligente Umgebungen, für betreut lebende Patienten. Auch hier wollten die Autoren den Aufenthaltsort des Patienten anhand seines vorherigen Bewegungsprofils identifizieren. Zu diesem Zweck wurden ebenso unterschiedliche Algorithmen verglichen in Bezug auf Genauigkeit und Fehlerquote. Sie kamen zu dem Schluss, dass maschinelles Lernen durchaus ein hilfreiches Instrument in der Bestimmung des Aufenthaltsortes darstellt, jedoch noch weiter unterstützt werden müsste durch zum Beispiel Filterfunktionen von Ausreißern, das Einbeziehen der Orientierung des Patienten oder andere Algorithmen.

4.9 Zusammenfassung

In diesem Kapitel wurden Arbeiten vorgestellt, welche aufgrund ihrer Bedeutung für die weitere Entwicklung auf dem Gebiet der Indoor Navigation ausgesucht wurden, sowie nach Relevanz für diese Arbeit. Es wurde gezeigt, dass mit unterschiedlichsten Mitteln und Technologien erstaunlich gute Ergebnisse bei der Indoor Positionsbestimmung erzielt werden können. Neben den Ergebnissen von größeren Forschungsprojekten aus aller Welt, wurden außerdem Abschlussarbeiten von Studenten vorgestellt, aufgrund der Nähe zu dieser Arbeit. Auch wurde deutlich, dass bei allen existenten Verfahren noch Schwachpunkte zu finden sind, welche einer endgültigen Marktreife im Wege stehen. Diese Arbeit wird sich mit der Fortführung einiger Gedanken der genannten Arbeiten befassen, als auch mit der Kombination aus verschiedenen Einzelaspekten.

Im nächsten Kapitel wird der Entwurf vorgestellt, nach welchem im Rahmen dieser Arbeit eine Indoor Navigations App entworfen werden soll. Hierbei wird die Technologie der WLAN Fingerprints genutzt und kombiniert mit den Daten aus einer Sensor-Fusion, sowie ausgewertet mithilfe von maschinellen Lernalgorithmen.

5 Entwurf eines Konzepts zur Indoor Positionsbestimmung

Im vorherigen Kapitel wurde gezeigt, dass sich einige Gruppen bereits seit gut 20 Jahren mit dem Thema der Indoor Navigation beschäftigt haben. Grundsätzlich wurde im Rahmen der Recherche zu dieser Arbeit Folgendes festgestellt: Die Anforderungen an ein Navigationssystem für Gebäude unterscheiden sich in einigen Punkten von denen, die an ein Outdoor Navigationssystem gestellt werden. Zwar sind die Anforderungen im Kern dieselben, nämlich Ortsbestimmung und Routenberechnung, jedoch bei der technischen Umsetzung und der verfügbaren Hardware stößt man schnell auf Unterschiede.

In diesem Kapitel soll daher der Entwurf einer Indoor Navigations App vorgestellt werden, welche diesen Anforderungen entspricht. Als Technologie zur Ortsbestimmung wird WLAN Fingerprinting genutzt, wobei die einzelnen Fingerprints auf einem externen Server per maschineller Lernalgorithmen verglichen und ausgewertet werden sollen. Außerdem sollen Sensordaten zusätzliche Sicherheit in der Positionsbestimmung liefern. Begonnen wird mit der Auswertung der Beispielszenarien, woraufhin die Systemstruktur skizziert wird.

5.1 Anforderungsanalyse

Die Beispielszenarien haben gezeigt, dass eine Indoor Navigation in vielen Lebensbereichen von großem Nutzen sein kann. Dies setzt jedoch voraus, dass die Position des Nutzers auf wenige Meter genau bestimmt werden kann. Häufig muss dafür im Vorfeld das Areal bekannt sein, in welchem navigiert werden soll, zumindest aber muss ein Gebäudeplan existieren.

Wie genau die Ortsbestimmung sein muss, hängt entscheidend von dem Anwendungsgebiet ab. In einer Bibliothek beispielsweise müsste mindestens eine Genauigkeit von einem Meter angestrebt werden, wohingegen bei einem anderen Gebäude schon Raumgenauigkeit, also zwei bis fünf Meter ausreichen können. An Flughäfen wären sogar noch größere Ungenauigkeiten tolerierbar, aufgrund der großen Areale und Läden.

Zusätzlich zur Genauigkeit kommt bei einem mobilen Gerät die begrenzte Akku-, Speicher- und Rechenkapazität hinzu. Eine App für Smartphones muss mit den zur Verfügung stehenden Ressourcen auskommen, denn diese sind nicht erweiterbar.

Wichtig ist außerdem, die sensiblen Daten des Nutzers vertraulich zu behandeln. Der Aufenthaltsort des Nutzers ist eine der wichtigsten Kontextinformationen und immer

mehr Nutzer entwickeln ein Bewusstsein hierfür [Emn02].

5.1.1 Genauigkeit der Ortsbestimmung

Der im Rahmen dieser Arbeit entwickelte Prototyp soll eine Indoor Navigation in Universitätsgebäuden ermöglichen. Die angestrebte Genauigkeit ist daher Raumgenauigkeit, also drei Meter, mit einer Fehlerquote von maximal 10%. Das bedeutet, dass sich die reale Position in 90% der Fälle innerhalb eines Radius von drei Metern um die geschätzte Position befinden muss, zusätzlich muss der Raum übereinstimmen.

Diese Genauigkeit sollte eigenen Recherchen nach ausreichen, damit sich der Nutzer in einem unbekanntem Gebäude zurechtfindet. Eine Navigation für Roboter oder andere autonome Fahrzeuge müsste genauer sein, ist jedoch nicht Ziel dieser Arbeit. Die angestrebte Genauigkeit sei hierbei auf das endgültige, dem Nutzer angezeigte Ergebnis zu beziehen, nach eventuellen Korrekturberechnungen oder Plausibilitätsprüfungen. Zusätzlich muss sie durch vorhandene Mittel realisierbar sein, wie im nächsten Abschnitt beschrieben.

5.1.2 Ressourcenschonende Implementierung

Eine Besonderheit von Mobilgeräten in verteilten Systemen beschreibt Fuchss in [Fuc09]. Fuchss definiert dort den Begriff des *mobilen Dilemmas*. Dieses beschreibt die beschränkte Rechenleistung von den Mobilgeräten selbst, sowie die der Mobilität geschuldeten, nicht immer vorhandenen Verbindung zu den restlichen Systemteilen. Fuchss schreibt:

„Möglichst viel Funktionalität muss in die mobilen Endgeräte verlagert werden, um die Kommunikation auf ein Mindestmaß zu reduzieren. Abhängigkeiten von stationären Rechnern (Servern) gilt es weitestgehend zu vermeiden.“

Gleichzeitig existiert jedoch parallel ein weiteres Problem, welches Fuchss wie folgt beschreibt.

„Da mobile Endgeräte ressourcenschwächer, fehleranfälliger und leichter zu kompromittieren sind, sollte man sie am besten als Thin-Clients betreiben. Die zentralen, rechenintensiven und sicherheitskritischen Aufgaben sollte man auf Serverinfrastrukturen im Festnetz verlagern.“

Jedes System, welches mobile Bestandteile innehat, ist inhärent von dem mobilen Dilemma betroffen. Viele Messungen und Berechnungen bedeuten eine starke Belastung für den Akku, diese Belastung ist Tests zufolge größer als der benötigte Aufwand für eine Kommunikation über das Internet [HKD07]. Daher bietet es sich an, die beschränkten Rechenkapazitäten eines Smartphones nicht zusätzlich zu strapazieren, sondern stattdessen die immer weiter ausgebaute Netzkommunikation zu nutzen. In Abschnitt 3.1.6

wurde gezeigt, dass heutzutage schnelle Kommunikation über WLAN möglich ist, weswegen für diese Arbeit der Entschluss getroffen wurde, die maschinellen Lernalgorithmen nicht auf dem Mobilgerät selbst zu implementieren, sondern stattdessen auf einen externen Server auszulagern. Grundsätzlich gilt zwar, dass eine Internetverbindung den Großteil des Akkuverbrauchs ausmacht, jedoch kommen Huber, Kreuzer und Diemer in [HKD07] zu dem Schluss, dass eine Berechnung von komplizierten Algorithmen auf einem Smartphone mehr Leistung benötigt, als das Übertragen kleinerer Datenmengen, wie es im Rahmen dieser Arbeit der Fall sein wird. Somit findet eine Ressourcensubstitution statt, wie sie in [BH12] beschrieben wird, um das Mobilgerät weitestgehend zu entlasten. Die Messungen der Sensorwerte können nicht ausgelagert werden, da die Sensoren ausschließlich im Mobilgerät verbaut sind. Aus diesen Entscheidungen resultiert, dass ein Nutzer, möchte er die Indoor Navigation in vollem Maße nutzen, ständige Verbindung zum Internet benötigt.

5.1.3 Frequenz der Messungen

Betrachtet man die Anforderungen an ein reguläres Navigationssystem für den Außenbereich, so fällt folgendes auf: Die Nutzer befinden sich in der Regel im Auto und bewegen sich mit 30 bis 130 km/h fort. Dies hat zur Folge, dass schon bei 30 km/h für eine genaue Ortung im Zwei-Meter-Bereich vier mal pro Sekunde die Position berechnet werden muss. Bewegt man sich mit 130 Kkm/h, so verändert sich die Aufenthaltsposition um 36 Meter pro Sekunde, entsprechend häufiger muss sie bei gleichbleibender Genauigkeit berechnet werden. Diese Masse an Positionsberechnungen auf einem Smartphone durchzuführen, ist alles andere als ressourcenschonend und nur deshalb mit Smartphones in Autos möglich, da diese dort gleichzeitig geladen werden können. Diese Möglichkeit entfällt bei der Indoor Navigation und eine solche Frequenz an Berechnungen kommt daher nicht für diese Arbeit in Frage. Bei einer Indoor Navigation jedoch, bewegt sich der Nutzer lediglich mit 3,6 bis 5 km/h, sodass für eine Ortung im Meterbereich schon eine einzige Berechnung pro Sekunde ausreicht - Korrektheit der Berechnung vorausgesetzt.

Es gilt also, dass für eine gute Positionsbestimmung die umliegenden Signalstärken im Schnitt einmal pro Sekunde erfasst und verarbeitet werden müssen, sofern die Berechnung korrekt ist. Da die Berechnung jedoch lediglich auf Wahrscheinlichkeiten beruht, empfiehlt es sich mehr als einen Scan pro Sekunde vorzunehmen, um Schwankungen und Störungen auszugleichen. Aus allen durchgeführten Scans kann dann ein Mittelwert der jeweiligen Netze gebildet werden, welcher mit den Bestandsdaten verglichen wird.

Neben der genauen Ortung muss ein Navigationssystem auch Routen berechnen können, damit Nutzer von einem Ort zum anderen geleitet werden zu können. Die Implementierung dieser Algorithmen und ihrer Berechnungen sollen nicht Teil dieser Arbeit sein, vielmehr geht es um die genaue Positionsbestimmung im Vorfeld, dennoch wird im folgenden Abschnitt das grobe Vorgehen bei einer Navigation skizziert.

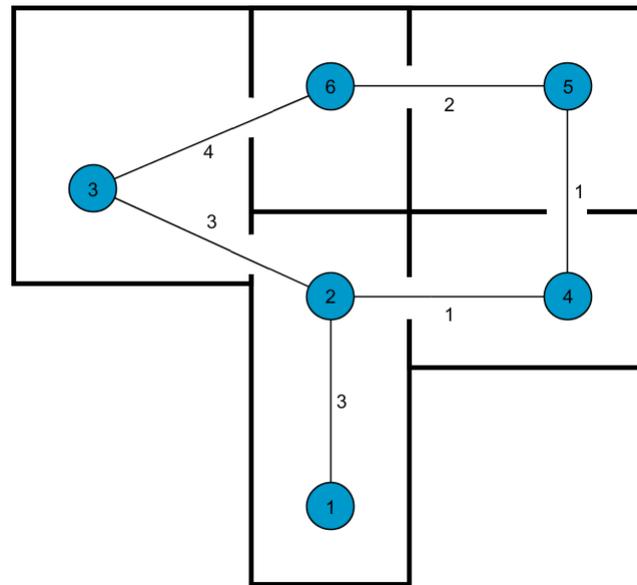


Abbildung 5.1: Beispielgraph mit Gewichtungen und 6 Knoten, eigene Abb.

Kürzester Pfad nach:	2	3	4	5	6
Iteration 1	3				
Iteration 2	3	6	4		
Iteration 3	3	6	4	5	10
Iteration 4	3	6	4	5	7
k. Pfad von 1 nach 7 ist: 1->2->4->5->6					

Tabelle 5.1: Ermittelte kürzeste Pfade nach 4 Iterationen

5.1.4 Navigation zu einem gewählten Ziel

Es gibt viele verschiedene Algorithmen, um den kürzesten Pfad von einem Punkt zum nächsten zu berechnen. In Anlehnung an [Kal96] und [LSW09] sei zunächst der *Dijkstra Algorithmus* genannt. Dieser berechnet ausgehend von einem Punkt in einem zusammenhängenden, gewichteten Graphen die kürzesten Pfade zu einem oder allen anderen Punkten. Dabei wird solange iteriert, bis zu jedem Punkt ein kürzester Pfad gefunden wurde. Der Dijkstra Algorithmus liefert immer ein optimales Ergebnis, sofern keine negativen Gewichtungen in dem Graphen vorkommen. Wählt man als Gewichtung die Entfernung in Metern zwischen zwei Punkten in einem Gebäude, so sind negative Werte nicht existent. Abbildung 5.1 zeigt einen Beispielgraphen und Tabelle 5.1 die dazugehörigen Iterationen und ermittelten kürzesten Pfade. In den ersten Schritten ist von Knoten 1 zu Knoten 7 kein Pfad bekannt, nach drei Iterationen ist einer gefunden, jedoch wird dieser nach der vierten Iteration verworfen, da ein kürzerer Pfad gefunden wurde.

Ebenso ist der *Bellman-Ford Algorithmus* geeignet, wie in [LSW09] näher ausgeführt wird. Dessen Vorteil im Vergleich zum Dijkstra Algorithmus ist, dass er zusätzlich mit negativen Gewichtungen umgehen kann, dafür steigt jedoch die Rechenkomplexität. Da

negative Gewichtungen in dem Szenario nicht vorkommen können, ist dies kein entscheidender Vorteil, also scheidet der Bellman-Ford Algorithmus aus.

In [Kal96] wird zudem der *Algorithmus nach Floyd* betrachtet, dessen Vorteil ist, dass er nicht nur von einem Punkt ausgehend die kürzesten Pfade berechnet, sondern für alle Punkte im Graphen, zu allen anderen Punkten. Dieser Algorithmus scheint aufgrund der hohen benötigten Rechenleistung geeignet, sofern er vor dem Betrieb der App ausgeführt werden kann, um die Entfernungen im Vorfeld zu berechnen. Während des Betriebs könnte dann eine vergleichsweise einfache Abfrage auf dem vorhandenen Datensatz ausgeführt werden, im Gegensatz zu einer neuen Berechnung. Während des Betriebes alle Pfade auf dem Mobilgerät selbst zu berechnen, scheidet wegen der knappen Ressourcen aus.

Als letzter möglicher Algorithmus sei der *A*-Algorithmus* genannt, welcher ebenfalls in [LSW09] Erwähnung findet. Anders als die zuvor genannten findet der A*-Algorithmus nicht immer den optimalen Pfad, ist jedoch dafür schneller in der Berechnung, da eine Heuristik angewandt wird, um die tatsächlichen Berechnungen zu verringern.

Welcher dieser Algorithmen sich tatsächlich am besten eignet, ist unter den verschiedenen Autoren der im Rahmen dieser Arbeit gefundenen Quellen umstritten. Je nach Anwendungsgebiet gibt es Vor- und Nachteile. Zu einer endgültigen Einschätzung müssten weitere Tests durchgeführt werden, um Speicherkapazitäten, benötigte Rechenzeit und andere Faktoren zu berücksichtigen.

5.1.5 Datenschutz von Kontextinformationen

Ein Aspekt, welcher berechtigterweise zunehmend an Bedeutung gewinnt, ist der Datenschutz der Nutzerdaten, inklusive der Kontextinformationen. In einer gemeinsamen Studie von Intel Labs, der Duke University und der Pennsylvania State University fanden die Autoren Enck et al. heraus, dass zwei Drittel der von ihnen analysierten Apps unerwünscht sensible Daten weitergeben [EGC⁺10]. Der Prototyp dieser Arbeit soll keinerlei Zugriff auf persönliche Daten benötigen, ebenso wie Kontextinformationen. Der Aufenthaltsort soll anonymisiert berechnet und rein anhand von eindeutigen Nummern (IDs) identifiziert werden. Der externe Server erhält zu diesem Zweck nur anonymisierte Orts-IDs und Signallevel, nicht jedoch die zugehörigen BSSIDs. Die letztendliche Verknüpfung von Orts-ID zum tatsächlichen Aufenthaltsort erfolgt auf dem Handy des Nutzers, welches als persönlicher und sicherer Raum gesehen werden kann. Mit diesem Ansatz soll gezeigt werden, dass ein datenschutzfreundlicher Dienst, welcher auf Kontextinformationen basiert im Bereich des Möglichen ist, selbst wenn die Daten an einen externen Server geschickt werden. Reinert befasste sich in ihrer Arbeit mit diesem Thema, musste jedoch feststellen, dass sich sensible Nutzerdaten und effektive Navigation nur schwer vereinen lassen [Rei04]. Diese Arbeit soll ihren Gedanken fortführen und jene Kombination aus Nutzen und Datensicherheit ermöglichen.

5.1.6 Verbesserung der Genauigkeit durch zusätzliche Inertialnavigation

In Abschnitt 3.1.5 wurde die Fusion mehrerer Sensoren beschrieben und wie anhand einer Inertial-Lösung ebenfalls eine Navigation realisiert werden kann. Der Prototyp dieser Arbeit soll einen Schritt weitergehen und die fusionierten Sensordaten nutzen, um die durch WLAN-Fingerprinting ermittelten Orte zu validieren. Hierdurch kann eine höhere Genauigkeit erreicht werden, wenn auch auf Kosten höheren Akkuverbrauchs. Eine zusätzlich dadurch ermöglichte Prüfung auf Plausibilität wird im folgenden Abschnitt näher erläutert.

Die Anforderungsanalyse hat die Herausforderungen bei dem Bau eines Prototypen aufgezeigt. Im nächsten Abschnitt soll die Architektur des Systems beschrieben werden, die eine Implementierung der Anforderungen ermöglicht.

5.2 Systementwurf

Im Folgenden soll die Systemarchitektur beschrieben werden, wobei auf die speziellen Anforderungen einer hybriden Navigations App eingegangen wird. Hierzu zählen die Erfassung der WLAN-Signalstärken, der Abgleich von aktuellen Messdaten und den gespeicherten WLAN-Fingerprints, die Bestimmung der aktuellen Bewegungsgeschwindigkeit und die Himmelsrichtung in welche das Smartphone ausgerichtet ist und die Kombination all dieser Daten, um eine konkrete Ortsangabe zu erhalten.

5.2.1 Anforderungen an die Infrastruktur

Ein wesentlicher Anspruch an den zu entwickelnden Prototypen ist, dass keine aufwändige Systeminfrastruktur geschaffen werden muss. Idealerweise soll Vorhandenes genutzt werden, oder durch die Installation von Hardware ein Mehrnutzen jenseits der Indoor Navigation geschaffen werden.

Abbildung 5.2 zeigt den Aufbau der Systeminfrastruktur. Diese besteht lediglich aus dem Mobilgerät und dem externen Server. Während die Messung der WLAN Signalstärken und die Messung der Ausrichtung des Handys auf dem Mobilgerät erfolgen müssen, werden die Referenzdaten und der maschinelle Lernalgorithmus ausgelagert auf einen externen Server. Die Kommunikation von Mobilgerät und externem Server erfolgt über das Internet, mit dem das Mobilgerät über Funk verbunden ist.

5.2.2 Ablauf und Systemkomponenten

Zur Bestimmung des Aufenthaltsortes eines Nutzers ist ein Zusammenspiel mehrerer Komponenten nötig. Der Nutzer selbst bedient lediglich die graphische Oberfläche (GUI) und sieht nicht, welche Prozesse im Hintergrund arbeiten. Somit muss der Prozess der Positionsbestimmung von der GUI aus aufgerufen werden. Abbildung 5.3 zeigt das Zusammenspiel der einzelnen Komponenten in der beschriebenen zeitlichen Abfolge. In-

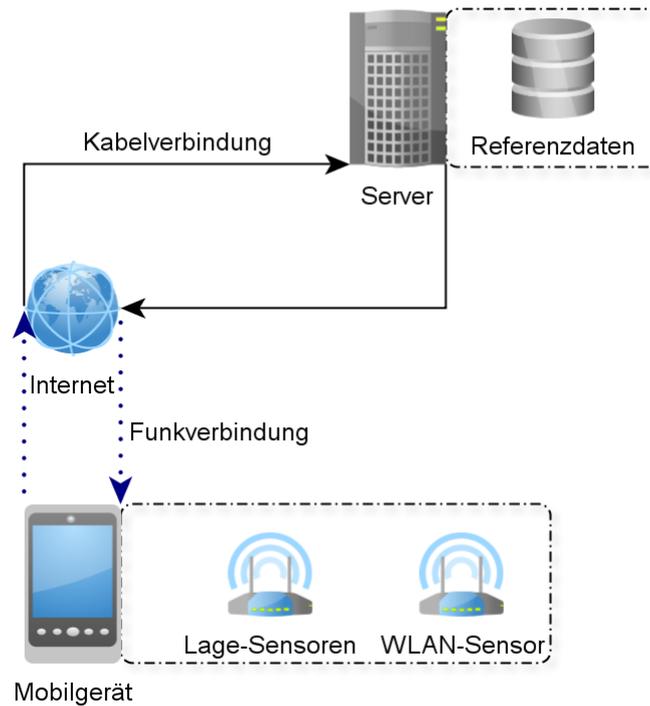


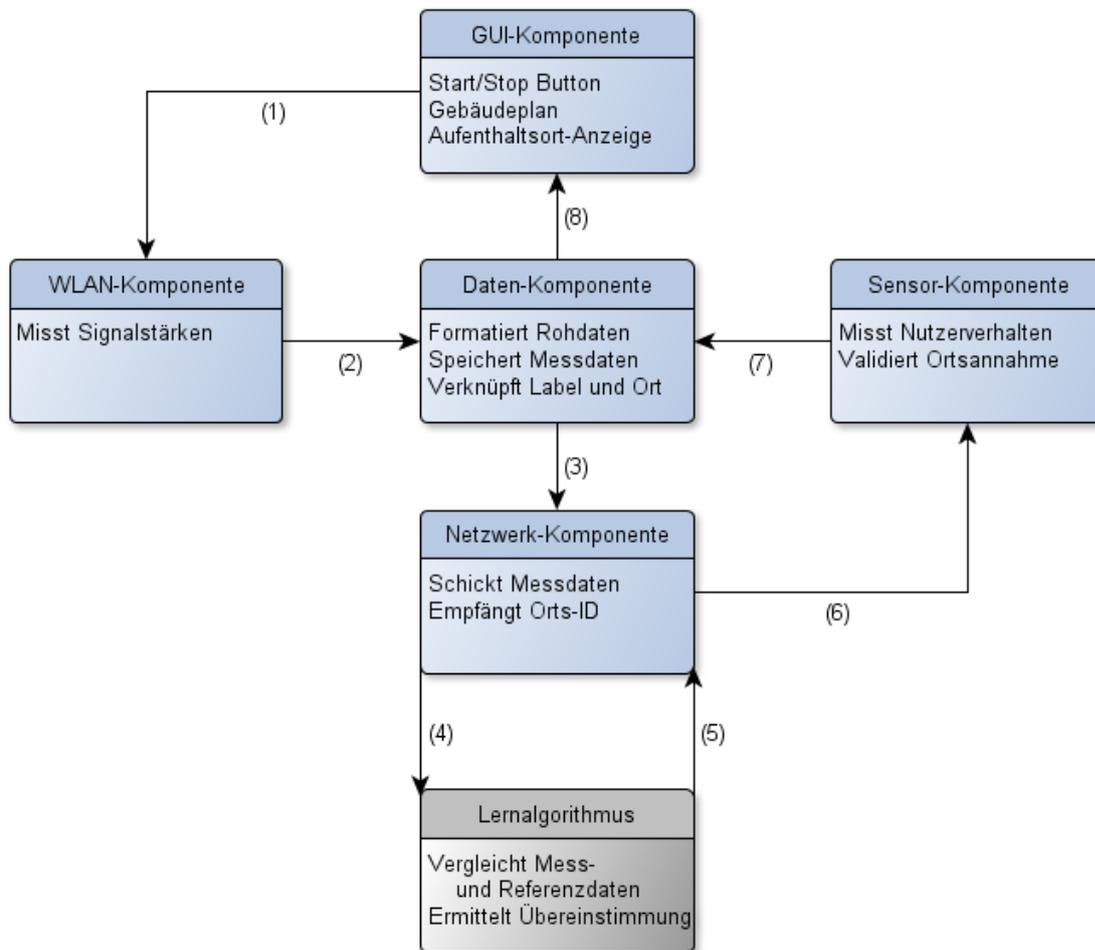
Abbildung 5.2: Systeminfrastruktur, eigene Abb.

nerhalb des Prozesses werden zunächst über die WLAN Komponente die umliegenden Netze erfasst und ihre Signalstärken gemessen. Diese Messwerte müssen aufbereitet und ins richtige Format gebracht werden, um sie sodann mithilfe der Netzwerk-Komponente an den externen Server zu senden. Dort vergleicht ein maschineller Lernalgorithmus die Messwerte mit den vorhandenen Referenzdaten und liefert als Ergebnis an die Netzwerkkomponente zurück, welche Werte die größten Übereinstimmungen haben, genauer gesagt den Ort, welchem diese Werte zugeordnet sind. Zeitgleich werden die Daten der Sensoren erfasst und ausgewertet, woraufhin die Sensordaten mit den Daten aus dem maschinellen Lernverfahren verglichen werden und ein gemeinsames Ergebnis bestimmt wird. Liegt dieses Ergebnis fest, so wird es zurück an die GUI berichtet, welche dann dem Nutzer den Aufenthaltsort auf einer Karte anzeigt.

5.2.3 Definition der Schnittstellen

Damit die Interaktion der verschiedenen Komponenten reibungslos funktioniert, müssen die Anforderungen an die jeweiligen Schnittstellen definiert werden.

Die im zeitlichen Ablauf erste Schnittstelle ist zwischen Nutzer und Mobilgerät. Die graphische Oberfläche soll intuitiv zu bedienen sein und jederzeit auf Eingaben reagieren können. Der aktuelle Aufenthaltsort muss aus der Anzeige hervorgehen. Zusätzlich muss es möglich sein, neue Gebiete zu vermessen und die zugehörigen WLAN-Fingerprints zu speichern, sodass eine Sammlung an Referenzdaten entsteht. Der Gebäudeplan soll angezeigt werden können und das Erfassen der Signalstärken von umlie-



- (1) Der Nutzer stößt den Ortungsprozess an
- (2) Die Signalstärken werden erfasst
- (3) und im passenden Format gespeichert
- (4) Die Messdaten werden an den Server geschickt
- (5) Der Server liefert die Übereinstimmungswerte
- (6) Das wahrscheinliche Label wird weitergereicht
- (7) Der durch Sensoren (inertial) ermittelte Ort wird zusammen mit dem Label weitergereicht
- (8) Nach Abgleich des Inertial ermittelten Ortes mit dem Ort des Fingerprinting-Verfahrens wird der Aufenthaltsort dem Nutzer angezeigt

Abbildung 5.3: Zusammenspiel der verschiedenen Komponenten, eigene Abb.

genden Netze für eine Neuberechnung der Position muss ermöglicht werden.

Die nächste Schnittstelle ist zwischen der GUI und der WLAN-Komponente. Über die WLAN-Komponente muss die Möglichkeit angeboten werden, Signalstärken zu erfassen und zur Aufbereitung zu übertragen, damit diese im Anschluss an den externen Server geschickt und dort gespeichert (im Falle von Referenzdaten) oder verglichen (im Betrieb) werden können.

Damit dies möglich ist, muss die Komponente zur Datenverarbeitung die Signalstärken als Rohdaten annehmen und diese aufbereiten können. Zusätzlich muss die vom Server zurückgelieferte Orts-ID wieder zum tatsächlichen Aufenthaltsort aufgeschlüsselt werden.

Die Kommunikation erfolgt über das Internet, die Netzwerk-Komponente muss hierzu die Authentifizierung gegenüber dem Server durchführen, zusätzlich die Daten an den Server schicken und die Antwort empfangen und verarbeiten können. Es muss davon ausgegangen werden, einen externen Server zu beanspruchen, welcher bestimmte eigene Schnittstellenanforderungen hat.

Zusätzlich müssen die Sensoren über das Programm ansprechbar sein und ihre Ergebnisse müssen fusioniert werden können. Die datenverarbeitende Komponente muss diese Fusion der Sensordaten durchführen und diese fusionierten Daten mit den Daten aus dem Fingerprinting abgleichen.

Abschließend muss die Kommunikation zwischen datenverarbeitender Komponente und GUI erfolgen, sodass der GUI die letztendliche Aufenthaltsposition übermittelt wird und dem Nutzer angezeigt werden kann.

5.2.4 Genutzte Technologien und Verfahren

Im Rahmen dieser Arbeit soll die Ortsbestimmung des Nutzers per WLAN erfolgen. Gründe hierfür sind die sehr hohe Verfügbarkeit von WLAN-Netzen und die große Sendereichweite von WLAN-Accesspoints (vgl. Abschnitt 3.1.6). Lediglich auf dem Land kann es vorkommen, dass kein WLAN-Netz in Reichweite ist. Da jedoch hier auch selten große Gebäudekomplexe anzutreffen sind, bei denen eine Indoor Navigation benötigt werden könnte, fällt dieser Aspekt nicht weiter ins Gewicht. Eine umfassende Übersicht der WLAN-Netzabdeckung bietet der in Abschnitt 4.7 erwähnte Dienst *Skyhook* [Sky12].

Methode zur Bestimmung des Aufenthaltsortes: In Kapitel 3 wurden verschiedene konkrete Möglichkeiten zur Positionsbestimmung mittels WLAN vorgestellt. Für diese Arbeit soll das Verfahren des Fingerprintings genutzt werden, da es den Recherchen nach am besten mit dem ergänzenden Ansatz des maschinellen Lernens vereinbar ist und gute Positionsbestimmungs-Ergebnisse zu erwarten sind.

Unterstützende Methoden zur Verbesserung der Genauigkeit: Zusätzlich zum Fingerprinting soll im Rahmen dieser Arbeit eine Fusion von Sensorinformationen im Test-

gerät zu größerer Genauigkeit bei den Ergebnissen führen. Hierbei werden, wie in Abschnitt 3.1.5 dargestellt der Kompass, und der Beschleunigungssensor genutzt.

Le et al. haben sich 2009 mit der Schrittmessung durch Sensoren beschäftigt und kamen zu dem Ergebnis, dass die in Smartphones verarbeiteten Sensoren zu ungenau sind und man mit ihnen lediglich erkennen kann ob sich der Nutzer bewegt oder still steht [LSW09]. Ursprünglich hatten auch sie vorgehabt, anhand eines Schrittmessers die Bewegungsgeschwindigkeit und Blickrichtung des Nutzers zu bestimmen. Aktuelle Beiträge in Entwicklerforen machen dennoch den Eindruck, als sei eine genaue Schrittmessung mit heutigen Smartphones möglich. Dies wird im Abschnitt 6.4.2 untersucht werden.

Es könnte auch ein zusätzlicher Plausibilitätscheck eingeführt werden, welcher prüft, ob Sprünge zwischen zwei Orten vorliegen. Ein solcher Sprung kann je nach Zeitabstand akzeptiert oder als unrealistisch erkannt werden. Im letzteren Fall müsste eine erneute Messung angestoßen werden. Für einen solchen Plausibilitätscheck müssen die einzelnen Messpunkte zusätzlich in einem gewichteten Graph übertragen werden, um lediglich Sprünge von einem Messpunkt zu dessen direkten Nachbarn zuzulassen oder bei entsprechendem zeitlichen Abstand zu akzeptieren. Wird eine Abweichung festgestellt, so zum Beispiel das Überspringen von zwei oder mehr Punkten, kann automatisch eine neue Messung angestoßen werden. Erst nach erneuter Bestätigung des Wertes sollte dieser als aktueller Standort übernommen werden. Vorkommen kann dies beispielsweise bei kurzzeitiger Unterbrechung der Internetverbindung, sodass der Nutzer sich tatsächlich normal weiterbewegt hat, jedoch keine Aktualisierung auf Serverseite stattfinden konnte. Der Vorteil einer Kombination aus Fingerprinting und Inertialnavigation wird in Abbildung 5.4 skizziert. Während eine reine WLAN Navigation eine verhältnismäßig hohe Ungenauigkeit hat und die Messfehler einer alleinstehende Inertialnavigation exponentiell ansteigen, ist durch eine Kombination beider Methoden ein deutlich genauerer Wert zu erwarten. Eine Inertialnavigation reagiert schneller als das Fingerprinting, welches nur in einzelnen Intervallen zu den Zeitpunkten t_i die Position aktualisiert. Diese einzelnen Aktualisierungen können jedoch als Referenzpunkte dienen und verhindern, dass sich der Fehler der eigenständigen Inertialnavigation fortpflanzt.

5.2.5 Bedienungsablauf als Mock-up

Abbildung 5.5 zeigt das geplante Design der App als Mock-up. Zunächst soll über einen Startbildschirm die gewünschte Aktion gewählt werden. Entweder das Vermessen einer neuen Umgebung um neue Referenzdaten für Fingerprints zu erhalten, oder die Navigation innerhalb des Gebäudes. Wird das Erfassen neuer Daten gewählt, so muss angegeben werden, in welchem Gebäude und wie oft ein Scandurchlauf durchgeführt werden soll. Wird die Navigation gewählt, so soll auf einem Gebäudeplan die aktuelle Aufenthaltsposition markiert werden, hier beispielsweise durch einen hellblau-transparenten Kreis, um eine gewisse Ungenauigkeit zu verdeutlichen. Das Navigationsziel kann wahlweise über eine Texteingabe erfolgen oder aber durch das Antippen des Ziels auf der

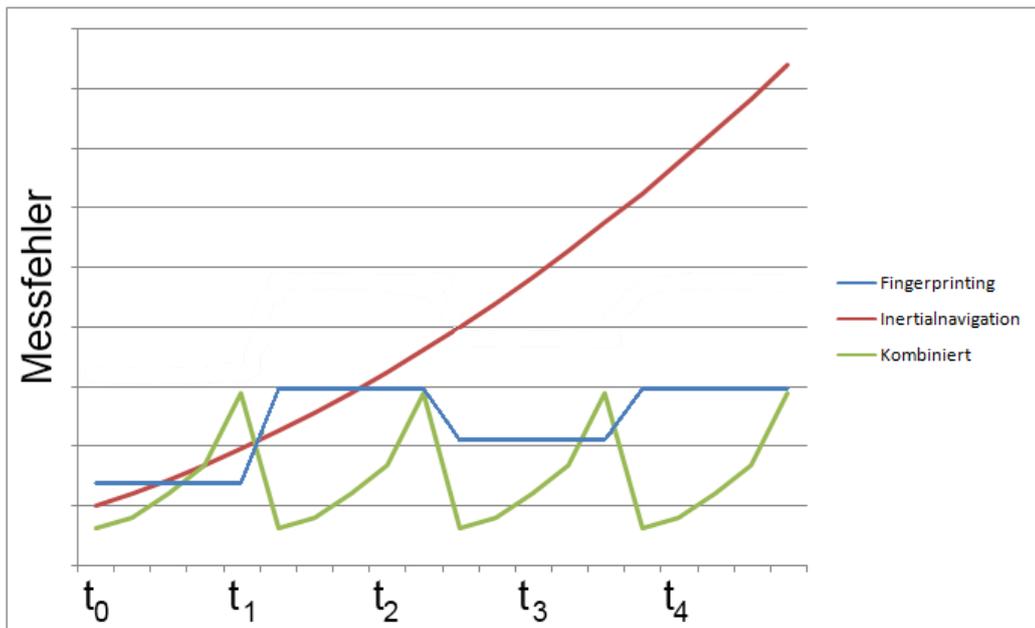


Abbildung 5.4: Zu erwartende Messfehler von Inertialnavigation und WLAN- Fingerprinting, einzeln und kombiniert (vereinfacht), eigene Abb.

Kartenansicht und wird als typisches rotes X auf der Karte markiert. Ein eingeblendeter Pfeil soll die einzuschlagende Richtung anzeigen.

5.2.6 Der Gebäudeplan

Bei dem Gebäudeplan stellt sich die Frage, ob er als feststehendes Bild in Displaygröße eingebunden werden sollte oder als scrollbares großes Bild. Auch wäre eine Gliederung in Raster denkbar, sodass der Plan sich, je nach Aufenthaltsort, aktualisiert und entsprechend bei Bedarf einen neuen Ausschnitt des Gesamtplans zeigt. Letztgenanntes hat sich bereits früh als unpraktikabel erwiesen, da es im Randbereich des Ausschnitts bei Fehlberechnungen des Aufenthaltsortes zu falsch angezeigten Ausschnitten kam. Besonders störend war dies in den jeweiligen Ecken des Ausschnitts, wo teilweise zwischen vier Ausschnitten hin und her gewechselt wurde. Hinzu kommt eine mangelnde Benutzerfreundlichkeit, da der Blick zurück, beziehungsweise der Überblick fehlt. Das Hinzufügen einer zusätzlichen Übersichtskarte, mit angezeigtem aktuellen Aufenthaltsort, um den mangelnden Überblick auszugleichen, bringt einen wieder zurück zur ersten Überlegung. Das Problem hierbei: Bei großen Gebäuden ist ein scrollbares Bild nicht zu vermeiden, denn das zwangsweise benötigte Zoomen geht mit einer scrollbaren Fläche einher. Da gerade in großen Gebäuden eine Indoor Navigation erst benötigt wird und weil die Nutzer eines Smartphones die scrollbaren Pläne aus Gründen der Gewohnheit intuitiv nutzen können, sollte ein scrollbarer Plan angestrebt werden. Abbildung 5.6 zeigt die drei möglichen Umsetzungen der Gebäudeplan-Ansicht auf einem Smartphone anhand des Grundrisses vom Schloss Versailles in Frankreich [Unk09].

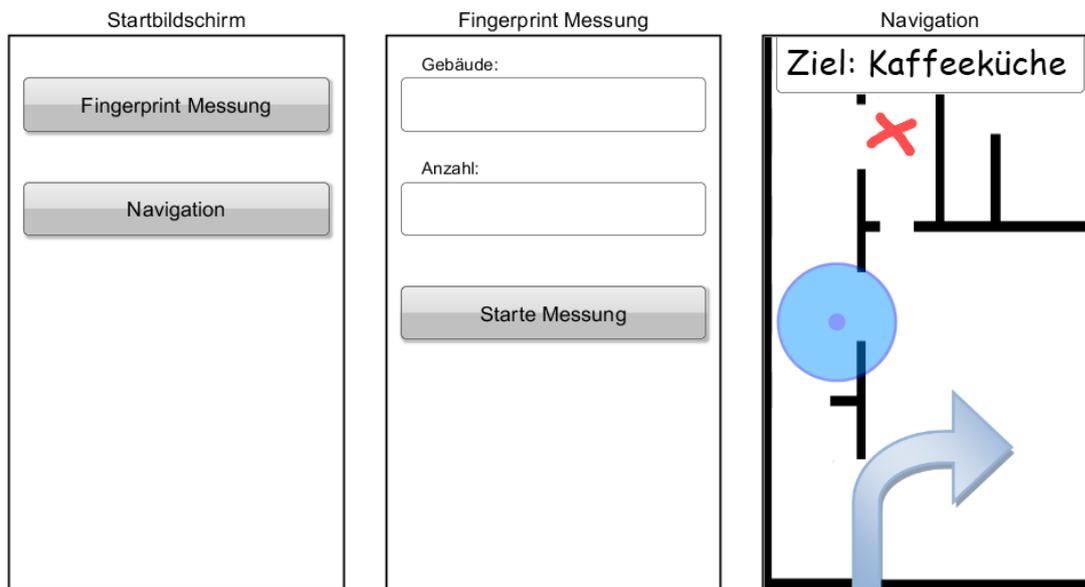


Abbildung 5.5: Mock-up der späteren App, eigene Abb.



Abbildung 5.6: Mögliche Gebäudeplan-Umsetzungen. Fix, Aufgeteilt, Scrollbar, eigene Abb. sowie [Unk09]

5.3 Zusammenfassung

Die zu entwickelnde App soll mittels der WLAN-Fingerprinting-Technik in Verbindung mit Sensordaten den aktuellen Aufenthaltsort des Nutzers ermitteln. Der Vergleich von Bestandsdaten und aktueller Messung soll auf einem externen Server stattfinden, welcher lediglich die übereinstimmende Orts-ID zurückgibt, diese aber nicht mit einem konkreten Ort verknüpfen kann. Da die letztendliche Verknüpfung von Fingerprints einerseits und Aufenthaltsorten andererseits lediglich auf dem Mobilgerät stattfindet, ist die Vertraulichkeit der sensiblen Daten gewährleistet. Alternative Angebote zur Indoor Navigation haben häufig die Algorithmen selbst auf einem zentralen Server gespeichert, welcher die Position bestimmt und zurückgibt. Hierdurch ist jedoch an einer zentralen Stelle die Aufenthaltsposition eines jeden Nutzers zu jeder Zeit bekannt, was ein zu großes Sicherheitsrisiko birgt und nicht mit einer datenschutzfreundlichen Nutzung der sensiblen Kontextinformation vereinbar ist.

In diesem Kapitel wurde das Konzept einer Indoor Navigations App vorgestellt, welche auf Grundlage von WLAN Fingerprints den Aufenthaltsort eines Nutzers durch maschinelle Lernverfahren ermittelt und diesen durch Kombination und Fusion der geräteeigenen Sensoren im Bezug auf die Genauigkeit verbessert. Es wurde anhand der Beispielszenarien in Kapitel 2 eine Anforderungsanalyse durchgeführt und konkretisiert. Hierbei wurde auf die Mobilgerät spezifischen Eigenschaften eingegangen, wie zum Beispiel die begrenzten Ressourcen. Zusätzlich wurde die Infrastruktur des zu entwickelnden Systems vorgestellt und ihre Komponenten und Schnittstellen angesprochen. Zum Ende des Kapitels wurde mithilfe eines Mock-ups die Vorstellung der zu realisierenden App skizziert. Im folgenden Kapitel wird die Implementierung des entworfenen Konzepts vorgestellt, außerdem werden Probleme beschrieben, welche im Laufe der Zeit während der Umsetzung auftraten.

6 Prototypische Implementierung

In Kapitel 5 wurde das Entwurfskonzept für einen Indoor Navigations Prototypen entwickelt, es wurde konkretisiert aus welchen Bestandteilen ein solches System besteht und auf welche Anwendungsfälle geachtet werden muss. Die Systemkomponenten wurden skizziert, sowie die Schnittstellen zwischen ihnen.

Die nächsten Abschnitte befassen sich mit der Implementierung und Programmierung eines Prototyps, welcher den Anforderungen aus dem Entwurf gerecht wird. Es wird die Arbeit an der Lernbasierten Indoor Navigations App *LIN* beschrieben. Hierbei zeigen die einzelnen Unterkapitel in grob chronologischer Reihenfolge die einzelnen Überlegungen und Umsetzungen in der Entwicklung.

Als Betriebssystem der App wurde *Android* [Goo10] gewählt, als Methode zur Berechnung des Aufenthaltsortes kommt WLAN-Fingerprinting zum Einsatz, kombiniert mit maschinellen Lernen durch das *Google Prediction API*. Der Vergleich der aktuellen und zuvor erstellten Fingerprints erfolgt auf Seiten von Google Prediction, die Verknüpfung der Google Prediction Ergebnisse und des tatsächlichen Aufenthaltsortes in einem Gebäude, und auch das Anzeigen der eigenen Position auf Gebäudeplänen erfolgt auf dem Mobilgerät selbst. Die zusätzliche Implementierung einer Inertialnavigation auf Basis einer Sensorfusion musste gestrichen werden, wie in Abschnitt 6.4.2 näher erläutert wird.

6.1 Entwicklungsumgebung Google Android

Grundlegend ist die Frage des Betriebssystems zu klären, auf welchem die App später laufen soll. Da für eine erfolgreiche App eine breite Community benötigt wird, um möglichst viele Gebäudepläne und Messungen von anderen Nutzern zu erhalten, kommt nur ein Betriebssystem in Frage, welches selber weit verbreitet ist. Aktuelle Marktanalysen und Prognosen zeigen, dass das Android Betriebssystem aufgrund seiner Beliebtheit deutlich vor iOS von Apple [App12] und RIM von Blackberry [RIM12] am besten geeignet ist (vergleiche Abb.6.1).

Die Entscheidung für Android bringt eine Reihe an Folge-Entscheidungen mit sich. Das Android Betriebssystem unterstützt Java als Programmiersprache, weswegen die Wahl der Programmiersprache entsprechend ausfiel. Android stellt gleichzeitig ein umfassendes Angebot für Software-Entwickler bereit, aufgeteilt in mehrere Aspekte beziehungsweise Schichten. Dabei steht jedem Entwickler jede Schicht zur Verfügung, von den hardware-nahen Treibern des Linux Kernels bis hin zu fertigen, ausführbaren App in der Applikationsschicht (vergleiche Abb.6.2).

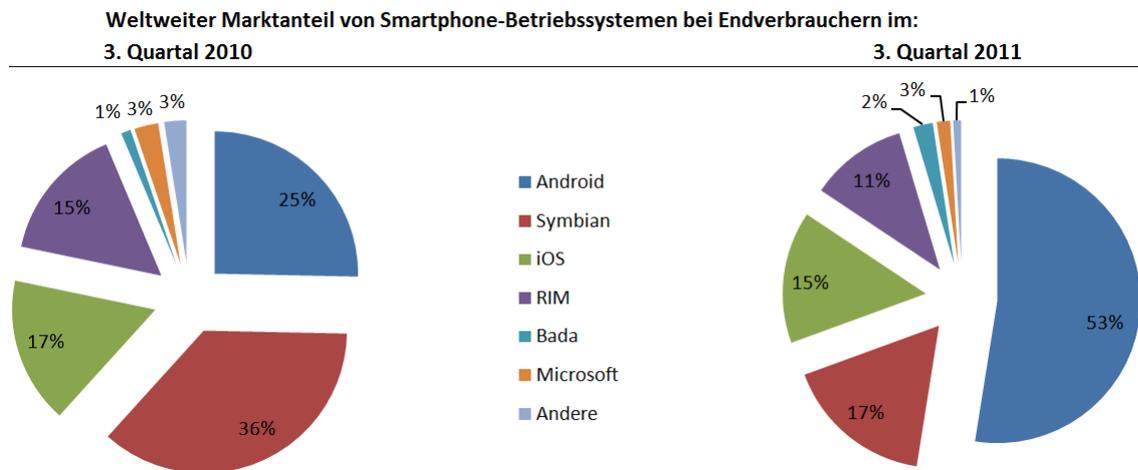


Abbildung 6.1: Smartphone Marktanteil Q3'10 und Q3'11. Quelle: [Gar11b]

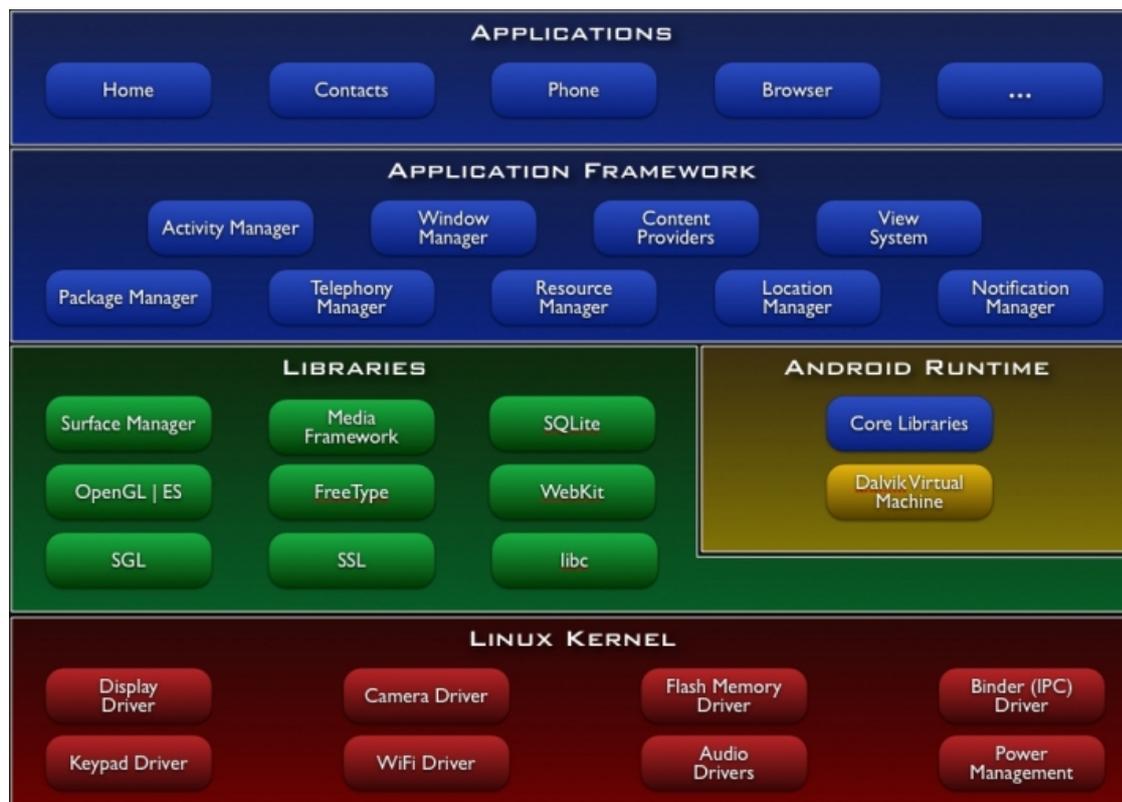


Abbildung 6.2: Der Aufbau des Android System Stacks. Quelle: [Goo10]

Das Android Betriebssystem bietet zudem eine Reihe von weiteren Besonderheiten. So ermöglicht es durch die eigens entwickelte Dalvik Virtual Machine (DVM) echtes Multitasking bei ressourcenschonender Prozessorleistung und durch den Dalvik Debug Monitor Server eine Analyse der benötigten Rechenleistung. Außerdem ist Android nach dem SandBox Prinzip aufgebaut, was zusätzliche Sicherheit bei der Entwicklung und Ausführung von Apps gibt [Goo12c].

6.1.1 Aufbau von Android Apps

Für Android gelten verschiedene grundlegende Architekturvorgaben. Die wichtigsten sollen an dieser Stelle kurz vorgestellt werden [Goo12b].

`Activities` sind das, was ein Nutzer sieht, wenn er die App bedient. Hierbei ist jeder neu angezeigte Bildschirm eine eigene Activity. Als Beispiel sei verwiesen auf Abbildung 5.5. Jede der drei Bildschirmanzeigen ist eine Activity. Klickt der Nutzer einen Button auf dem Startbildschirm an, so öffnet sich die nächste Activity und zeigt beispielsweise den Gebäudeplan. Diese Activities überlagern sich gegenseitig, das heißt, in dem Moment wo ein Nutzer im Menü einen Schritt zurück navigiert, wird die zuvor geöffnete Activity wieder angezeigt, welche die Zeit über im Hintergrund gewartet hat. Der genaue Lebenszyklus einer Activity wird in Abbildung 6.3 dargestellt. Für eine detaillierte Beschreibung der Grafik sei auf die Ursprungsseite verwiesen. Zusätzlich zu Activities sind für diese Arbeit auch `Intents` und `Tasks` von Bedeutung. Ein Intent ist quasi eine asynchrone Nachricht innerhalb des Systems, über Intents werden beispielsweise neue Activities gestartet und beendet. Der Intent fungiert hierbei als eine Art Nachrichtendienst, und beschreibt was eine Applikation mit welchem Objekt tun möchte, zum Beispiel eben eine Activity anzeigen.

`Tasks` sind anders als Activities für den Nutzer nicht sichtbar und verrichten ihre Dienste im Hintergrund in einem eigenen Thread. Android bietet die Möglichkeit von echtem Multitasking, also der Ausführung mehrerer Threads parallel. In dieser Arbeit ist ein Task sehr zentral, da er die Messung der Fingerprintdaten im Hintergrund durchführt, sowie die Daten an Google Prediction sendet, empfängt und verarbeitet. Dies kann nicht innerhalb des Threads der angezeigten Activity geschehen, da während der Rechenzeit eine Aktion auf dem User Interface blockiert würde.

6.2 Das Mobilgerät

Im Rahmen dieser Arbeit kommt als Testgerät ein *HTC Desire* [III12] zum Einsatz¹. Es handelt sich hierbei um ein älteres Smartphone, welches im 2. Quartal 2010 auf den Markt kam. Das Gerät verfügt über einen Single-Core Prozessor Qualcomm QSD 8250 (1.000 MHz) und 576 MByte RAM. Es unterstützt die Datenübertragung via HSDPA mit 7.2

¹Im späteren Verlauf musste der Feldtest mit einem HTC Desire Z durchgeführt werden, aufgrund von Problemen des Android 2.2 Betriebssystems und der genutzten REST-Schnittstelle (s. Abs. 6.4.4).

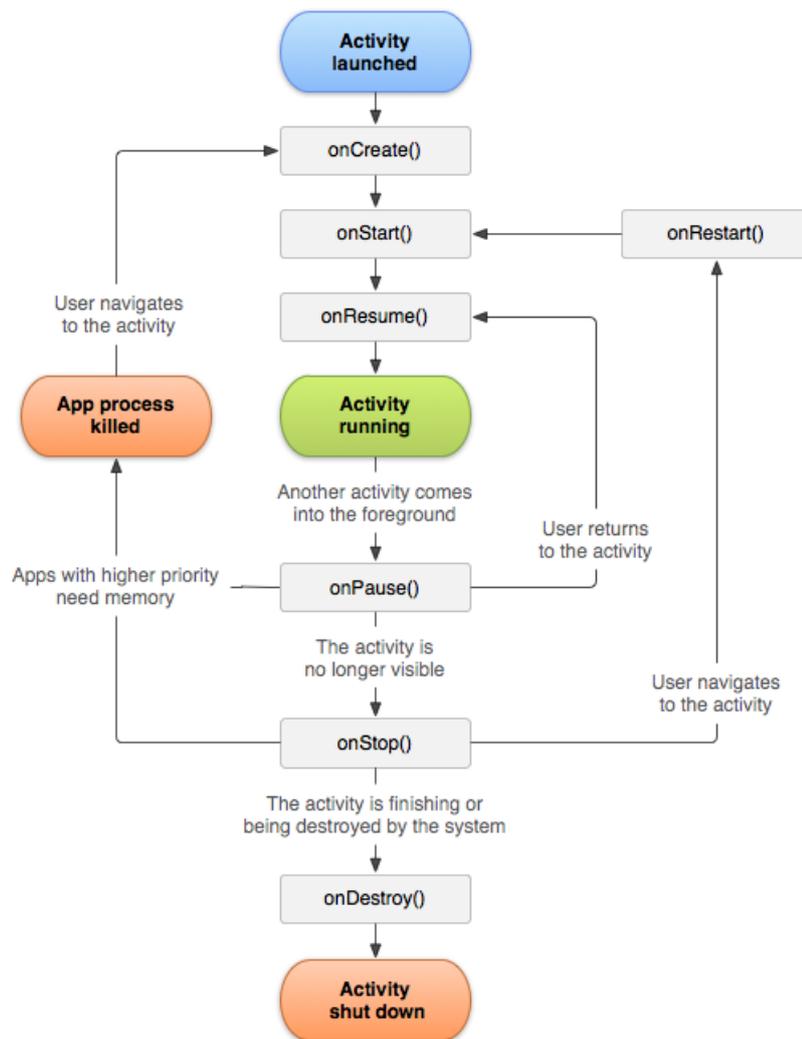


Abbildung 6.3: Der Activity Lifecycle[Goo12a]

MBit/s und die WLAN Standards IEEE 802.11b, IEEE 802.11g und IEEE 802.11n. Zusätzlich ist ein Orientierungssensor (magnetischer Kompass) und ein Beschleunigungssensor vorhanden. Das Betriebssystem ist Android 2.2 und somit auf einem halbwegs aktuellen Stand.

Damit erfüllt das HTC Desire die benötigten Grundvoraussetzungen, um als Testgerät zu fungieren.

6.3 Die Testumgebung

Die Tests des Prototypen erfolgen am Informatikum der Uni Hamburg, auf dem Flur im vierten Stock von Haus F. Hierzu wurde die Testumgebung vermessen und an verschiedenen Orten Referenzdaten gesammelt. Abbildung 6.4 zeigt die einzelnen Orte, jeder blaue Punkt markiert einen Messpunkt. Der Abstand der Punkte auf dem Flur beträgt zwei bis drei Meter, je nach örtlichen Gegebenheiten, wie zum Beispiel Brandschutztüren, Pflanzen, Tischen et cetera. Zusätzlich wurde in jedem Raum eine Messung durchgeführt, jeweils in der Mitte des Raumes, sofern dies möglich war. Ausgegraute Flächen waren leider zum Zeitpunkt der Messung nicht begehbar. An jedem Punkt wurden 50 Referenzmessungen durchgeführt. Die Recherche im Rahmen dieser Arbeit hat ergeben, dass häufig schon weniger Referenzdaten ausreichen, je nach gewähltem maschinellen Algorithmus, jedoch gibt Google Prediction nicht den gewählten Algorithmus aus. Zu diesem Zweck wurden 50 Referenzmessungen gewählt und in ersten Feldtests als guter Kompromiss zwischen zeitlichem Aufwand der Messung und Genauigkeit der Ortsvorhersagen bestätigt.

6.4 Systemkomponenten und Positionsbestimmung

Bei der Bestimmung des Aufenthaltsortes soll sich aufgrund der in Abschnitt 3.1.6 genannten Schwachstellen nicht auf das Fingerprinting alleine verlassen werden. Zusätzlich kommen die eingebauten Sensoren in dem Smartphone zum Einsatz. Über den Orientierungssensor kann festgestellt werden, in welche Himmelsrichtung das Smartphone gedreht ist, also in welche Richtung der Nutzer sich bewegt. Zusätzlich kann über den Beschleunigungssensor die Gehgeschwindigkeit ermittelt werden. Eine entsprechende Fusion der Sensoren, wie sie in Abschnitt 3.1.5 beschrieben ist, liefert somit hilfreiche, unterstützende Daten.

Aufgrund der unterschiedlichen Technologien, welche in dem Prototyp dieser Arbeit einander ergänzend genutzt werden, sind unterschiedliche einzelne Komponenten miteinander zu kombinieren. In den folgenden Abschnitten werden die genutzten Dienste näher vorgestellt.

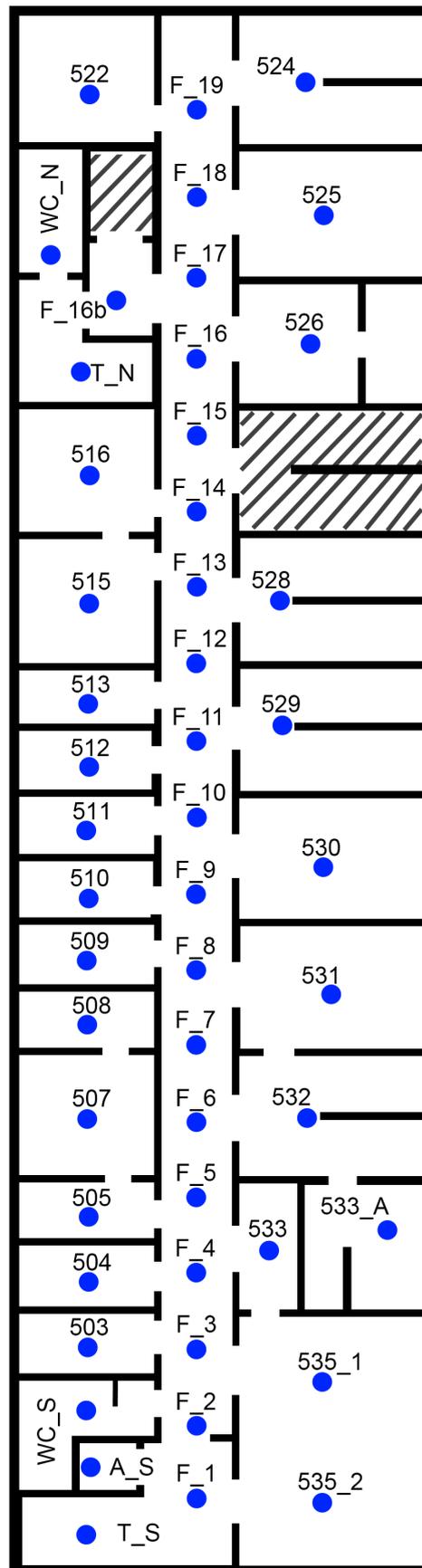


Abbildung 6.4: Die vermessene Testumgebung am Informatikum der Uni Hamburg, eigene Abb.

6.4.1 Der WifiManager

In Abschnitt 3.1.6 wurde das Verfahren des Fingerprintings bereits vorgestellt. Um dieses auf Mobilgeräten mit dem Android Betriebssystem umzusetzen, benötigt man den von Android bereitgestellten `WifiManager`. Dieser ermöglicht es, die Netze um das Smartphone herum zu erfassen und unter anderem deren Signalstärken zu ermitteln. Zusätzlich können die Sendefrequenz und verschiedene Angaben zur Netzverschlüsselung erfasst werden, welche jedoch für diese Arbeit nicht weiter von Interesse sein werden.

Aus den Signalstärken werden Fingerprints gebildet und serverseitig gespeichert. Ebenso können aber auch aktuelle Messungen als Fingerprint dargestellt und mit den Referenzdaten verglichen werden. Um eine höhere Trefferquote und somit Genauigkeit der Ortsbestimmung zu erzielen, werden zusätzlich die Daten aus dem Orientierungs- und Beschleunigungssensor hinzugezogen. Eine detaillierte Beschreibung hierzu findet sich im anschließenden Abschnitt.

6.4.2 Der SensorManager

Die Klasse `SensorManager` ermöglicht einem den Zugriff auf die Sensoren des Smartphones [Goo11d]. Hierzu werden zunächst die einzelnen Sensoren in der Activity deklariert, welche genutzt werden sollen und zusätzlich eine Instanz des `SensorManagers` bereitgestellt. Relevant für diese Arbeit sind die Himmelsrichtung, die Richtungsachse der Bewegung (s. Abb. 6.5) und das Bewegungstempo. Die Achsenausrichtung des Koordinatensystems ist hierbei so, dass die Z-Achse vom Erdmittelpunkt weg zeigt, also dem Anheben des Smartphones entspricht. Die Y-Achse zeigt nach Norden, sodass die X-Achse nach Osten ausgerichtet ist. Für eine App, welche auch im *LandscapeModus*, also wenn das Handy 90° gedreht ist, betrieben werden kann, müsste man zusätzlich die Orientierung des Smartphones ermitteln. Für den Prototyp im Rahmen dieser Arbeit soll jedoch davon ausgegangen werden, dass das Mobilgerät stets in einer Hand aufrecht vor dem Körper gehalten wird.

Liefert Google Prediction einen Wert zurück, welcher in entgegengesetzter Richtung zur Blickrichtung liegt, was bedeuten würde dass der Nutzer umgekehrt ist oder rückwärts gelaufen ist, so kann dieser Wert als falsch angesehen und eine neue Messung gestartet werden. Ebenso können die Werte von Google Prediction verworfen werden, wenn die Sensoren feststellen, dass sich der Nutzer nicht bewegt, Google Prediction jedoch unterschiedliche Werte zurückgibt, typischerweise abwechselnd zwischen zwei Annahmen schwankt. Dies könnte der Fall sein, wenn der Nutzer exakt zwischen zwei Referenzpunkten stehen bleibt, sodass die Übereinstimmungen abwechselnd größer bei dem einen oder anderen sind.

Die Genauigkeit des eingebauten digitalen Kompasses ist nicht mit einem mechanischen zu vergleichen. So konnten regelmäßig Abweichungen von bis zu 30° festgestellt werden. Die Konsequenz ist, den Mittelwert aus mehreren Messungen kurz hintereinan-

Achse	liegend		gehalten	
	Mittelwert	Median	Mittelwert	Median
X	0,1496059	0,14982383	0,064288042	0,040861044
Y	-0,176846598	-0,19068487	1,985410853	3,3778462
Z	10,35309874	10,337844	6,770475197	9,343558

Tabelle 6.1: Mittelwert und Median der Messwerte des Beschleunigungssensors während das Mobilgerät still liegt / gehalten wurde, $n=375$, Werte in m/s^2

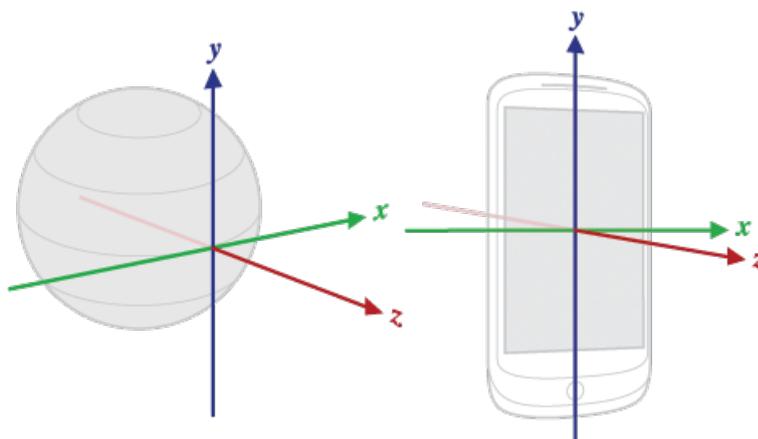


Abbildung 6.5: Ausrichtung der verschiedenen Achsen bei Android [Goo12d]

der zu bilden, also die häufigste gemessene Richtung zu bestimmen und mit dieser zu arbeiten.

Um die Schritte des Nutzers zu registrieren, muss zunächst definiert werden, wie ein Schritt festgestellt werden kann. Hierzu wurden die Daten des Beschleunigungssensors ausgewertet, während das Testgerät flach und still auf dem Tisch lag, es ruhig in der Hand gehalten wurde und während es im 45° Winkel still gehalten wurde. Abbildung 6.5 zeigt die Orientierung der X-, Y- und Z-Achse bei Android, dargestellt mit Blick auf den Globus und das Mobilgerät selbst.

Tabelle 6.1 zeigt die Mittelwerte und den Median aus 375 Messwerten. Alle Werte werden in m/s^2 angegeben. Wie in Abschnitt 3.1.5 beschrieben, kann man deutlich die Wirkung der Erdanziehungskraft auf der Z-Achse erkennen.

Auffällig war jedoch auch, dass sobald das Mobilgerät nicht flach lag, die Erdanziehungskraft auf Z- und Y-Achse wirkte, jedoch nicht mit einer Kraft, die in Summe ca. zehn m/s^2 war, sondern vielmehr ca. 13 bis 14. Abbildung 6.6 zeigt die entsprechenden Messungen.

Eine Bewegung in Richtung der X-Achse gibt keinerlei Informationen zu der Schrittmessung, daher wird sie an dieser Stelle vorerst vernachlässigt. Die Messungen haben gezeigt, dass ein Schritt definiert werden kann, als Punkt zu dem ein definierter Grenzwert überschritten ist. Es wurden verschiedene Grenzwerte ausprobiert, die besten Ergebnisse lieferte die App wenn die Werte der Y- und Z-Achse addiert wurden und als

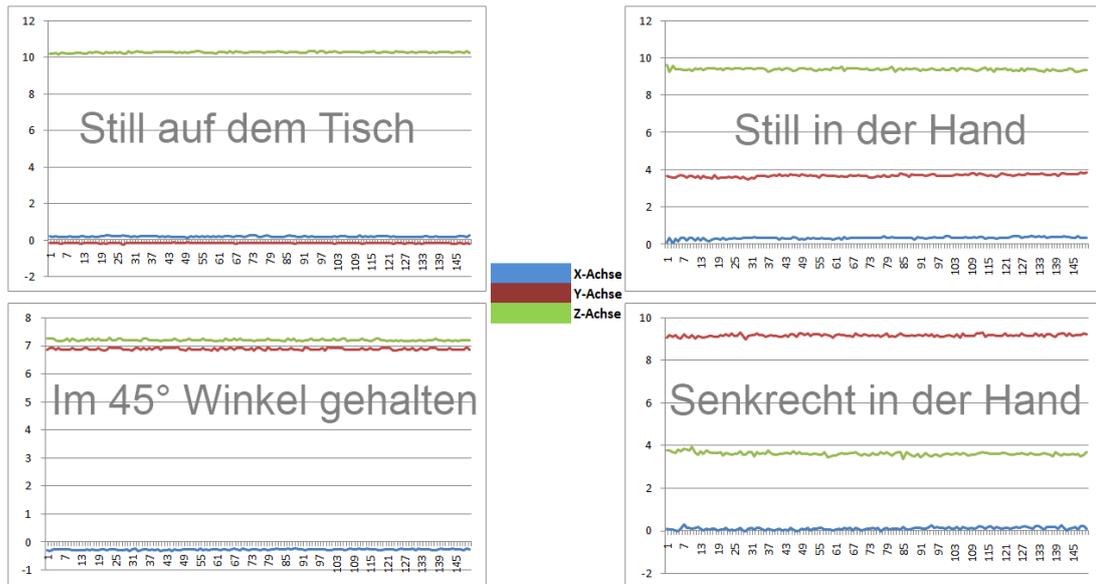


Abbildung 6.6: Unterschiedliche Messwerte, je nach Haltung des Mobilgeräts, eigene Messung

Grenzwert 14.7 m/s^2 festgelegt. Zusätzlich wurde ein Messintervall von 500 Millisekunden definiert, um jeden Schritt nur einmal zu erfassen. Mit dieser Kalibrierung wurde in 8 von 10 Fällen die richtige Schrittzahl gemessen. Hierbei wurden sowohl gerade Strecken, als auch Kurven und Kehrtwenden getestet.

Abbildung 6.7 zeigt abschließend die addierten Messwerte von Y- und Z-Achse während des Gehens. Zu Beachten ist hier, dass lediglich die Ausschlagswerte nach oben betrachtet werden, eine höhere Genauigkeit könnte erreicht werden, indem die Abweichungen in den niedrigeren Bereich ebenfalls berücksichtigt werden, ein Schritt also definiert sei als ein Übersteigen des Wertes 14.7 gefolgt von einem Fallen unter einen Grenzwert von ca. 10.5 m/s^2 . Für diese Arbeit jedoch reicht diese Genauigkeit aus.

Aufgrund der Vorgaben, die den Rahmen und die Bearbeitungszeit dieser Arbeit regulieren, konnte zwar die Schrittmessung und das Ermitteln der Blickrichtung realisiert werden, auf eine Inertialnavigation, welche diese Sensordaten nutzt um die Werte von Google Prediction zu validieren oder zu korrigieren, musste jedoch verzichtet werden.

6.4.3 Datenspeicherung

Wie im Entwurfskapitel begründet, wird bei LIN der Gebäudeplan sowie eine Lookup-Tabelle der Referenzpunkte auf dem Handy selbst gespeichert, die Fingerprints aus der Offline-Phase hingegen in der Google-Cloud, wo sie für das Training der maschinellen Lern-Algorithmen genutzt werden und innerhalb des Betriebs zum Vergleich benötigt werden.

Innerhalb der App werden die Fingerprints als `LinkedHashMap` gespeichert. Eine `HashMap` speichert die Datensätze als Key-Value Paare. Ein Wert, welcher zu einem be-

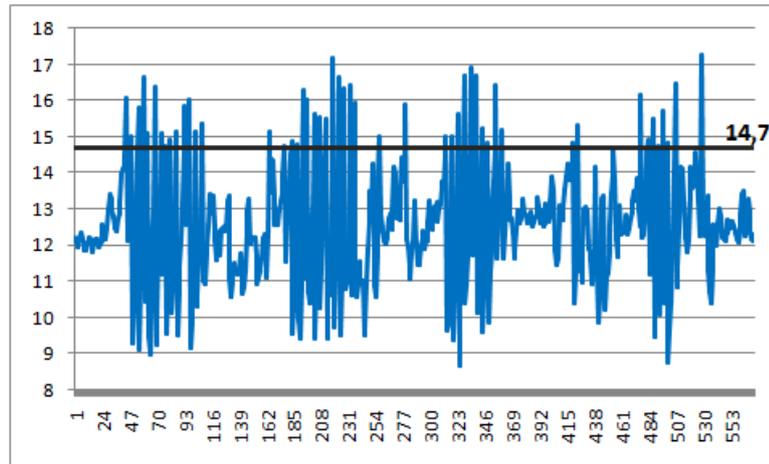


Abbildung 6.7: Messwerte während des Gehens, eigene Messung

Key (BSSIDs)	Value (Signalstärken)
bd:05:43:ef:5c:b9	-100,-100,-85,-93,-100,-100,-85,-93
88:2a:2c:5d:8c:69	-100,-100,-64,-100,-100,-100,-64,-100
14:da:3b:de:43:ed	-95,-75,-100,-83,-95,-75,-100,-83
5d:92:d2:6a:96:55	-61-100,-100,-42,-61-100,-100,-42
1a:7c:d5:20:65:f3	-37,-76,-100,-100,-37,-76,-100,-100

Tabelle 6.2: Beispielhafte Darstellung der Daten innerhalb der LinkedHashMap

stimmten Schlüssel hinterlegt wird, kann also anhand des Schlüssels leicht identifiziert und wieder abgefragt werden. Der Schlüssel ist hierbei beliebig wählbar, im Rahmen dieser Arbeit wurde er als Text implementiert und repräsentiert eine eindeutige Kennung eines Messpunktes. Die Werte in der LinkedHashMap liegen alphabetisch sortiert vor. Die Datensätze, welche zu den Keys gespeichert werden, sind Arrays vom Typ Integer. Somit können für jeden Key mehrere Werte aus unterschiedlichen Messungen gespeichert werden. Tabelle 6.2 zeigt exemplarisch den Aufbau der LinkedHashMap.

Im folgenden Abschnitt wird genauer beschrieben, wie Google Prediction in den Prototyp dieser Arbeit eingebunden wurde.

6.4.4 Google Prediction

Google hat im vergangenen Jahr das Projekt *Google Prediction* öffentlich zugänglich gemacht. Dieses ermöglicht es, wie in Abschnitt 3.2.2 beschrieben, eigene gesammelte Daten von Google Prediction auswerten zu lassen, sodass lediglich die Anfrage und die Antwort selbst auf dem Mobilteil verarbeitet werden müssen. Google Prediction bietet damit eine leichtgewichtige Möglichkeit, komplexe Rechenarbeiten, wie maschinelle Lernalgorithmen auszulagern und die Ressourcen des Mobilgerät-Prozessors gegen die der Netzkommunikation zu tauschen.

In Abschnitt 3.1.6 wurde beschrieben, dass der Prozess beim Fingerprinting in zwei Phasen unterteilt ist, die Lern- bzw. Trainingsphase und die Betriebsphase. Erstere beschreibt das Messen und Aufbereiten der Referenzdaten, sowie das Trainieren der maschinellen Lernverfahren mit diesen. Betriebsphase meint die Zeit nach der Trainingsphase, in welcher der Algorithmus einsatzfähig ist.

Innerhalb der Trainingsphase wird nun aus den unterschiedlichen gemessenen Signalstärken ein Bestand an Referenzdaten erstellt. Diese müssen in dem Speicherplatz der *Google Cloud* gespeichert werden, nur so hat Google Prediction Zugriff auf sie. Ein Datensatz besteht dabei aus einer Comma-Separated Values (CSV) Datei, einer Darstellung von Tabellen, in welcher einzelne Zellenwerte durch Kommata separiert sind. Listing 6.1 zeigt ein Beispiel einer CSV Datei, mit welcher Google Prediction trainiert werden kann. Die erste Spalte (der erste Wert in jeder Zeile) ist hierbei das sogenannte *Label*, das Erkennungsmerkmal für den gesamten folgenden Datensatz, also den Rest der Zeile. Google Prediction vergleicht anhand der Datensätze und liefert das Label mit der höchsten Übereinstimmung zurück. Zusätzlich wird die relative Übereinstimmung der einzelnen Label in Prozent ausgegeben, also die Wahrscheinlichkeit, mit der die verglichenen Datensätze tatsächlich zu dem jeweiligen Label gehören. Der Aufbau und die Entstehung der Datensätze in dieser Form wird im Folgenden näher erläutert.

Listing 6.1: Auszug aus Training.csv

```
1 "A1", -100, -100, -95, -61, -37
2 "B2", -100, -100, -75, -100, -76
3 "C3", -85, -64, -100, -100, -100
4 "D4", -93, -100, -83, -42, -100
```

In der Betriebsphase wird bei einer Ortsabfrage der Mittelwert aus mehreren Scans berechnet, um Schwankungen und Störungen auszugleichen. Die durchschnittliche Signalstärke eines jeden erfassten Netzes wird nun an Google Prediction gesendet, welches die Werte mit den gespeicherten Fingerprints aus der Trainingsphase vergleicht und das Label mit der höchsten Übereinstimmung zurückliefert, zusätzlich mit der dazugehörigen Gewichtung.

Google Prediction bietet keinerlei hierarchische Struktur in den Daten an und das einzige akzeptierte Dateiformat ist das CSV Format. In dieser Datei können zwar beliebig viele Spalten sein, es gibt jedoch keine Gewichtung einzelner. Das heißt, die BSSID in der ersten Spalte (das Label wird bei der Anfrage nicht übertragen) wiegt genauso schwer wie der Wert in der letzten (vgl. [Goo11e, Structuring the Training Data]).

Angenommen Prediction wurde mit Daten nach dem Muster in Tabelle 6.3 trainiert, dann wiegt die BSSID, welche für eine Übereinstimmung zweier Werte bei beiden Werten die gleiche sein *muss*, nicht stärker als die verschiedenen Signalstärken. Bei 99 Messungen der Signalstärke, fließt die BSSID in die Entscheidung mit einem Gewicht von $1/100$ ein, es würden also faktisch nur die verschiedenen Signalstärken verglichen. Für Tabelle 6.3 hieße dies, dass die erste und zweite Zeile fast übereinstimmen würden (eben bis auf die

BSSID), obwohl es sich um unterschiedliche Räume und BSSIDs handelt.

Ort	BSSID	Signalstärke							
Raum 1	BSSID3	-66	-63	-62	-60	-66	-63	-62	-60
Raum 2	BSSID4	-66	-63	-62	-60	-66	-63	-62	-60

Tabelle 6.3: Ausschnitt beispielhafter Trainingsdaten für Google Prediction

Eine Lösung für dieses Problem ist der folgende Ansatz. Prediction speichert nicht, wie anfänglich geplant einen Satz von mehreren 100 Levelstärken, sondern bildet einen Fingerprint als ein Key-Value-Paar ab, mit der BSSID als Key und dem Durchschnittswert der gemessenen Signalstärken als Value. Somit hätte die BSSID im späteren Vergleich wieder ein ausschlaggebendes Gewicht - nämlich 50% und gleichzeitig würde der Speicherbedarf gesenkt, wenn auch auf Kosten der Rechenleistung. Tabelle 6.4 zeigt beispielhaft die überarbeitete Struktur der verschiedenen Fingerprints.

Ort der Messung	Empfangene BSSID	Durchschnittswert der Signalstärke
Raum 1	BSSID3	-62,75
Raum 2	BSSID4	-62,75

Tabelle 6.4: Überarbeitete Datenstruktur

Durchgeführte Tests zeigten, dass mit diesen Daten kein zuverlässiges Ergebnis erreicht werden konnte, weswegen eine erneute Umstrukturierung stattfinden musste. Der endgültige Aufbau des Datensatzes ist in Tabelle 6.5 dargestellt. Es werden alle bekannten BSSIDs der Umgebung aufgeführt, dabei sind in der Tabelle die einzelnen Spalten die Signalstärken der jeweiligen BSSID, aufgelistet in Zeile 1 (diese wird später nicht an Google Prediction übermittelt). Die erste Spalte beinhaltet das Label, also die Orts-ID wo die Messung durchgeführt wurde. Ein Wert von -100 bedeutet hierbei, dass die entsprechende BSSID nicht in Reichweite war, also nicht im Scan erfasst wurde. Listing 6.1 zeigt einen Ausschnitt der CSV Datei, welche in der Trainingsphase an Google Prediction geschickt wurde. Die erste Tabellenzeile entfällt, sodass lediglich die Werte übrig bleiben. Als Konsequenz daraus geht jedoch auch die Zuordnung der Werte verloren, weswegen immer die gleiche Reihenfolge gewahrt bleiben muss. Besonders aufwändig ist dies, wenn eine neue BSSID gefunden wurde, denn aufgrund der alphabetischen Reihenfolge müssen die gesamten Datensätze umstrukturiert werden. Hat man bereits eine ausreichend genaue Positionsbestimmung, so kann das Hinzufügen zusätzlicher BSSIDs auch entfallen, schließlich wurde festgestellt, dass die hinzukommenden BSSIDs in den allermeisten Scans lediglich einmal erfasst wurden und ansonsten außerhalb der Reichweite lagen, somit also auch keine deutliche Verbesserung der Genauigkeit lieferten.

Die Kommunikation mit dem Google Prediction API erfolgt über eine REST-Schnittstelle [Fie00] mittels HTTP Get Requests [Gro99]. Listing 6.2 zeigt, wie eine Anfrage an Google Prediction gestellt werden kann. Dieser Ausschnitt der Anfrage beginnt mit der Deklaration eines folgenden inputs, welcher aus einer CSV-Komponente besteht. Innerhalb die-

	bd:05:43:ef:5c:b9	88:2a:2c:5d:8c:69	14:da:3b:de:43:ed	5d:92:d2:6a:96:55	1a:7c:d5:20:65:f3
A1	-100	-100	-95	-61	-37
B2	-100	-100	-75	-100	-76
C3	-85	-64	-100	-100	-100
D4	-93	-100	-83	-42	-100

Tabelle 6.5: Tabellenstruktur der Trainings CSV Datei

ser CSV-Komponente wird jeder einzelne Wert in Anführungsstriche gesetzt und durch ein Komma vom Folgewert separiert.

In Listing 6.3 wird das Beispiel einer Antwort gezeigt, wobei die ID den Namen des genutzten Modells beschreibt und das outputLabel dasjenige ist, für welches die größte Übereinstimmung festgestellt wurde. Es folgt eine Auflistung aller anderen Label, mit den zugehörigen Übereinstimmungswerten. Für die getesteten Daten und die Daten des Labels „D4“ wurde in diesem Beispiel eine Übereinstimmung von 94% festgestellt.

Listing 6.2: Beispiel eines Requests

```

1 {
2   "input":
3     {"csvInstance":
4       [
5         "-93.0",
6         "-100.0",
7         "-83.0",
8         "-42.0",
9         "-100.0"
10      ]
11    }
12 }

```

Listing 6.3: Beispiel der Response

```

1 {
2   "kind": "prediction#output",
3   "id": "Test",
4   "outputLabel": "D4",
5   "outputMulti": [
6     {
7       "label": "A1",
8       "score": 0.002352
9     },
10  ]

```

```
11  "label": "D4",
12  "score": 0.941696
13  },
14  {
15  "label": "C3",
16  "score": 0.024858
17  },
18  {
19  "label": "B2",
20  "score": 0.031094
21  }
22  ]
23 }
```

6.5 Zusammenfassung

In diesem Kapitel wurde die Implementierung des Prototyps beschrieben, der im Rahmen dieser Arbeit entwickelt wurde. Zunächst wurde die Entscheidung für das Betriebssystem Android erläutert, ebenso wie die Testumgebung und das Testgerät vorgestellt wurden. Anschließend wurde gezeigt, wie der Aufenthaltsort berechnet werden kann. Es folgte eine Auflistung der zentralen Systemkomponenten und eine Beschreibung der Komplikationen, welche im Laufe der Zeit eintraten. Die Umsetzung der ursprünglich geplanten Intertialnavigation musste aufgrund von Zeit- und Umfangsvorgaben gestrichen werden.

Im nächsten Kapitel werden die erzielten Ergebnisse dieser Arbeit evaluiert, sowie weitere Dinge beschrieben, welche aus verschiedenen Gründen nicht erreicht oder umgesetzt werden konnten.

7 Evaluation

Im vorherigen Kapitel wurde der entwickelte Prototyp vorgestellt. Es wurden Schwachstellen aufgezeigt und die Herausforderungen einer Kommunikation mit einem externen Dienst in Form von Google Prediction verdeutlicht.

In diesem Kapitel wird auf die Ergebnisse der Eignungstests eingegangen, die mit dem entwickelten Prototyp durchgeführt wurden. Zunächst wird die Genauigkeit analysiert, woraufhin eine Beschreibung der möglichen auftretenden Probleme folgt. Diese Probleme werden teilweise wieder im nachfolgenden Kapitel aufgegriffen, zusätzlich werden dort Lösungsvorschläge für sie skizziert.

7.1 Überprüfung der Anforderungen

In Kapitel 5 wurden auf den Beispielszenarien aufbauend die Anforderungen an eine Indoor Navigation und somit an eine Indoor Positionsbestimmung ermittelt. In diesem Abschnitt sollen die einzelnen Anforderungen überprüft werden. Das Hauptkriterium für die Bestimmung des Aufenthaltsortes ist die Genauigkeit der ermittelten Position. Das Ziel dieser Arbeit sollte Raumgenauigkeit sein. Weitere Anforderungen waren die ressourcenschonende Implementierung und eine angemessene Frequenz der Messungen, um beide zuvor genannten Punkte zu erfüllen. Die Navigation zu einem gewählten Ziel und die Verbesserung der Genauigkeit durch eine Inertialnavigation konnte im Rahmen dieser Arbeit nicht umgesetzt werden und wird daher nicht näher evaluiert. Der Schutz von sensiblen Kontextinformationen konnte erreicht werden, indem die Nutzerdaten anonymisiert analysiert werden und keinerlei Zugriff auf andere Kontextinformationen benötigt wird.

7.1.1 Erreichte Genauigkeit

Im Rahmen dieser Arbeit wurde das Testgelände unterteilt und 50 Messpunkte festgelegt. An jedem einzelnen wurden 50 Messungen durchgeführt und diese Werte als Referenzdaten gespeichert. Die Genauigkeit von Google Prediction wurde in drei verschiedenen Vorgehen geprüft. Zunächst wurden zufällig ausgewählte Datensätze der Referenzdaten für einen Punkt ausgewählt, also die Signalstärken aus einer der jeweils 50 Messungen. Mit diesen Daten wurde eine Anfrage an das Google Prediction API gestellt. Dieses Vorgehen wurde für jeden der 50 Punkte durchgeführt, wobei in allen 50 Tests der korrekte Ort ermittelt wurde. Auch bei den Referenzpunkten auf dem Flur des Gebäu-

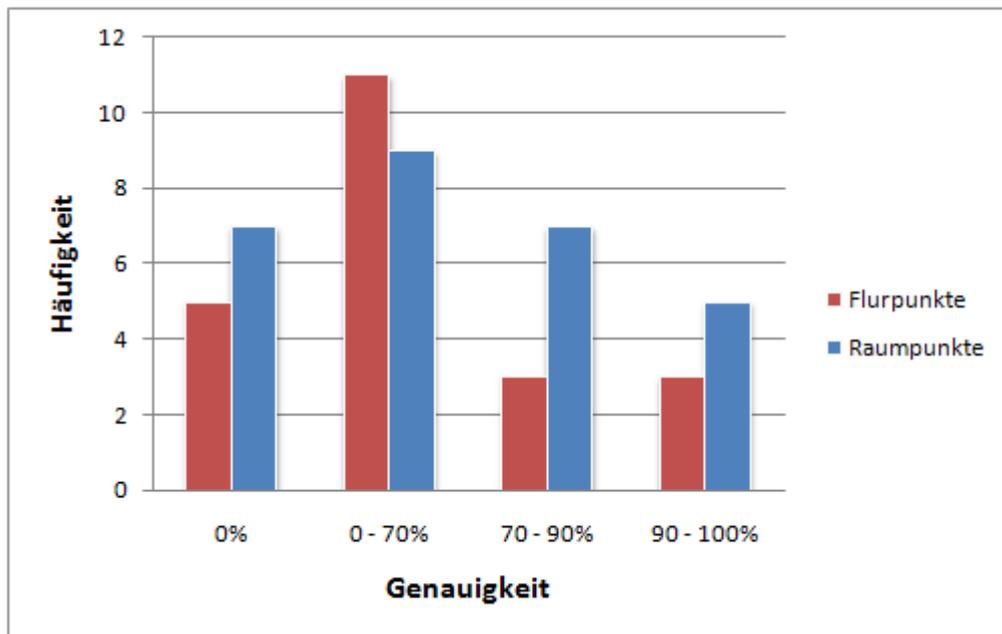


Abbildung 7.1: Anzahl der ermittelten Genauigkeiten, eigene Messung

des, welche direkt benachbart lagen und zwischen denen keine Gebäudewand existierte, die für deutliche Unterschiede der Signalstärken hätte sorgen können, konnte diese Genauigkeit erreicht werden.

Anschließend wurde aus den 50 Referenzdaten eines jeden Punktes der Durchschnitt der jeweiligen Signalstärken gebildet. Auch mit diesen Werten wurde eine Abfrage für jeden der 50 Punkte an Google Prediction gestellt, wobei es ebenfalls in allen Fällen zu einer richtigen Ortsbestimmung kam. Dies entspricht einer Genauigkeit von 100% für beide genannte Vorgehen.

Der dritte und letzte Test war der Betrieb des Prototyps und fand auf dem Testgelände statt. Es wurde das Gelände abgegangen und in Echtzeit die ermittelten Orte verglichen und validiert.

Hierbei wurden teilweise deutliche Abweichungen von angezeigtem und tatsächlichen Ort festgestellt, sodass insgesamt lediglich eine Genauigkeit von 14% erreicht werden konnte, anstelle der angestrebten 90%. Abbildung 7.1 zeigt die Verteilung der einzelnen Messergebnisse, gruppiert in Klassen von 0%, 0-70%, 70-90% und 90-100% Genauigkeit. Zusätzlich sind die Werte unterteilt in Messpunkte auf dem offenen Flur und in Raumpunkte, da zwischen diesen teilweise große Unterschiede festgestellt wurden. Es wird veranschaulicht, dass bei elf Flurpunkten die Genauigkeit zwischen 0% und 70% lag, was für eine Navigation unzureichend ist. Ebenso ist zu sehen, dass bei sieben Raumpunkten die Genauigkeit zumindest zwischen 70% und 90% lag, was zumindest durch zusätzliche Verbesserungen, wie die beschriebene Inertialnavigation, für die Nutzung ausreichen könnte.

In Abbildung 7.2 werden die unterschiedlichen Punkte gegenübergestellt. Es ist deut-

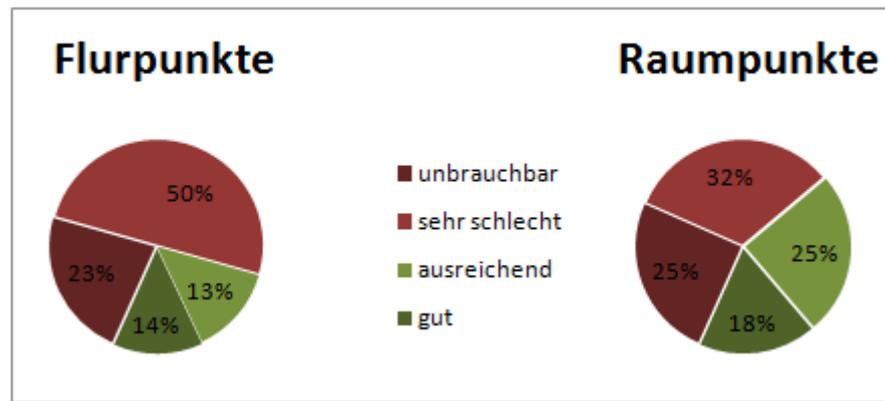


Abbildung 7.2: Vergleich der Genauigkeiten von Flur- und Raumpunkten, eigene Messung

lich, dass bei den Raumpunkten insgesamt eine höhere Genauigkeit erreicht werden konnte. So kam es bei insgesamt 43% der Messungen in den Räumen zu einer Genauigkeit von mindestens 70%. Unter den Flurpunkten waren lediglich 27% der Messungen in diesem Bereich.

Die Testergebnisse zeigen, dass allein durch das Fingerprinting in Kombination mit Google Prediction keine befriedigende Genauigkeit erreicht werden kann. Somit kann festgehalten werden, dass eine Indoor Navigation mithilfe von WLAN Fingerprints in Kombination mit maschinellem Lernen zwar grundsätzlich denkbar ist, aber durch weitere Verfahren gestützt werden muss.

7.1.2 Ressourcenschonende Implementierung und Messungsfrequenz

Wie in Abschnitt 5.1.2 dargelegt wurde, betrifft die Anforderung einer ressourcenschonenden Implementierung alle Systeme mit mobilen Bestandteilen. Im Rahmen dieser Arbeit sollte daher die angestrebte Genauigkeit erreicht werden, ohne das Mobilgerät stärker als nötig zu belasten. Dieses Ziel konnte durch die Ressourcensubstitution, welche im Abschnitt 5.1.2 angestrebt wurde, erreicht werden. Der entwickelte Prototyp führt im Betrieb nur die notwendige Erfassung der Sensordaten und Signalstärken aus, wohingegen der Vergleich mittels maschineller Lernalgorithmen ausgelagert wurde an den von Google bereitgestellten Dienst Google Prediction. In [HKD07] wurde gezeigt, dass der Aufbau einer Internetverbindung für kleine Datenmengen ressourcenschonender ist, als das Ausführen von maschinellen Lernverfahren auf dem Mobilgerät selbst.

Es konnte bei den Tests festgestellt werden, dass in einer Stunde aktiver Nutzung der App, lediglich 23% der Akkuleistung des Testgerätes verbraucht wurden. Dies bedeutet, dass mit der derzeitigen Implementierung eine durchgehende Navigation von ca. vier Stunden ermöglicht wird, trotz aktiver Internetverbindung und Sensordatenabfrage. Dieses Ergebnis kann durch eine Optimierung der Nutzung von Ressourcen weiter verbessert werden.

7.1.3 Datenschutz von sensiblen Kontextinformationen

Um die Privatsphäre der Nutzer zu wahren, ist es bei einer Indoor Navigation nötig, die sensiblen Daten nicht an einem zentralen Ort zu bündeln oder sie verknüpfbar zu machen. Dieses Ziel konnte ebenfalls verwirklicht werden, indem die Verknüpfung des Aufenthaltsortes mit den Daten von Google Prediction auf Seiten des Mobilgerätes stattfindet. Der Vergleich, welcher extern von Google Prediction durchgeführt wird, ist weder verknüpfbar mit den erfassten BSSIDs der umliegenden Netze, noch mit dem Gebäude, in welchem sich ein Nutzer befindet oder mit dem Nutzer selbst (vgl. hierzu Listing 7.1). Hierzu muss ein entsprechendes Level der Abstraktion bei der Benennung der Dateien und Label gewählt werden. Im Rahmen dieser Arbeit wurden aus Gründen der Transparenz Namen gewählt, die Rückschlüsse erlauben, jedoch bietet die Struktur des Systems die Möglichkeit zu deutlich besserem Schutz.

7.1.4 Streichungen in der Implementierung

Aufgrund der begrenzten Bearbeitungszeit dieser Arbeit musste auf einige Ziele aus dem Entwurfskonzept verzichtet werden. Hierzu gehört die Validierung der von Google Prediction zurückgelieferten Orte durch eine Inertialnavigation, sowie eine Richtungsanzeige und Berechnung des kürzesten Pfades zur Navigation.

Wie in Abschnitt 6.4.2 gezeigt, gelang es im Rahmen dieser Arbeit die verschiedenen Sensoren zu vereinen, um eine genaue Schrittmessung zu erhalten und zusätzlich die Blickrichtung zu ermitteln, jedoch musste auf die Verbindung aus Sensordaten und den Werten von Google Prediction verzichtet werden. Diese einzubauen hätte weitere Zeit beansprucht, welche von der Zeit des Hauptthemas, der Positionsbestimmung mittels WLAN Fingerprints und maschinellem Lernen, hätte abgezogen werden müssen.

Eine Navigation, also die Berechnung eines kürzesten Pfades zwischen zwei Punkten und die anschließende Anzeige dieses Pfades musste ebenfalls gestrichen werden. Diese wurde jedoch in vielen anderen Arbeiten bereits zentral untersucht, sodass es sich anbietet, eine bereits entwickelte, gute Navigationslösung mit der im Rahmen dieser Arbeit entwickelten Positionsbestimmung zu verbinden.

7.2 Der entwickelte Prototyp

Der entwickelte Prototyp kann, innerhalb einer zuvor vermessenen Umgebung, die eigene Aufenthaltsposition, anhand von WLAN Fingerprints und maschinellem Lernen, ermitteln. Hierzu wurden zwei wesentliche Bedienungsoberflächen geschaffen, welche in Abbildung 7.3 gezeigt werden. Hierbei wurde sich weitestgehend an das Mock-up aus Abschnitt 5.2.5 gehalten und dieses ergänzt.

Der erste Bildschirmausschnitt zeigt die Oberfläche zum Erfassen neuer Referenzdaten. Es muss der aktuelle Aufenthaltsort angegeben werden, damit die Daten diesem zu-



Abbildung 7.3: Screenshots des entwickelten Prototyps, eigene Abb.

geordnet werden können. Zusätzlich muss angegeben werden wie viele Referenzdaten erfasst werden sollen. Ergänzend zum Mock-up kam hinzu, dass ebenso Durchschnittswerte der erfassten Signalstärken gespeichert werden, sowie eine Liste aller erfassten BSSIDs. Um das Training von Google Prediction zu vereinfachen, ist es dazu möglich, die Daten direkt als transponierte Tabelle zu speichern, die ohne weitere Bearbeitung zu Google Prediction geschickt werden kann. Der zweite Bildschirm zeigt den Gebäudeplan, auf welchem, anhand eines hellblauen Kreises, der aktuelle Aufenthaltsort markiert ist. Im Hintergrund werden regelmäßig die verfügbaren Signalstärken erfasst und gesammelt und an Google Prediction übermittelt, sodass die Anzeige aktualisiert wird, sobald sich der Aufenthaltsort verändert.

Zu Kontrollzwecken wurde eine Ausgabe im Android Log eingebaut, welche die aktuellen Messwerte und die von Google Prediction zugehörigen Orte ausgibt. Listing 7.1 zeigt eine solche Ausgabe für einen Scan.

Listing 7.1: Auszug aus dem Log

```

1 I/System.out(11076): Model: HausF
2 I/System.out(11076): Predict: {input={csvInstance=[
3 -100,-100,-100,-100,-83,-82,-77,-77,-100,-100,
4 -100,-100,-100,-100,-100,-100,-100,-100,-100,-100,
5 -100,-100,-100,-100,-100,-100,-100,-100,-100,-100,
6 -100,-100,-100,-100,-100,-100,-100,-100,-100,-100,
7 -100,-100,-100,-100,-88,-89,-89,-90,-100,-100,
8 -100,-100,-100,-100,-100,-100,-100,-100,-100,-100,
9 -100,-100,-100,-100,-100,-100,-100,-100,-100,-100,
10 -100,-100,-100,-100,-100,-100,-100,-100,-100 ]}}
```

11 ==> F524

Der Aufbau gleicht hierbei der Anfrage, welche direkt an das Google Prediction API gerichtet wird und in Listing 6.2 auf Seite 63 gezeigt wurde. Die letzte Zeile gibt Aufschluss über den von Google Prediction angenommenen Ort. Hierbei wird das Label ausgegeben, nach welchem sich die App richtet, um die Ortsanzeige zu aktualisieren, für die gezeigten Werte, ist also die Übereinstimmung mit den Daten des Labels F524 am größten.

7.3 Nutzenanalyse

In diesem Abschnitt wird der erreichte Nutzen diskutiert, welcher von den entwickelten Prototyp generiert wird. Hierbei wird auf die Tauglichkeit im Alltag, also die Positionsbestimmung in einem Gebäude und auf die fachlichen und technischen Probleme eingegangen.

7.3.1 Alltagstauglichkeit des Prototyps

Das grundlegende Ziel, eine raumgenaue Positionsbestimmung zu erreichen, konnte im Rahmen dieser Arbeit nicht erreicht werden. Ein Mehrwert durch das Nutzen der App, speziell für eine breite Masse anderer Nutzer, würde erst entstehen, wenn die Genauigkeit weiter verbessert würde und auch eine Navigation von einem Punkt zum nächsten ermöglicht wird. Daher dient der entwickelte Prototyp eher als Proof of Concept, denn als alltagstaugliche App, welche anderen Nutzern in ihrem jetzigen Zustand zur Verfügung gestellt werden könnte.

Hervorzuheben bleibt der positive Effekt in der Entwicklung, durch den Ansatz, den externen Dienst Google Prediction zur Bestimmung des Aufenthaltsortes einzubinden. Hierdurch konnten erhebliche Aufwände in der Entwicklung verschiedener Maschinel-ler Lernverfahren eingespart werden. Das Einbetten dieses Dienstes scheint auch in Zukunft sinnvoll und wünschenswert, wenn große Datenmengen analysiert werden müssen und gleichzeitig das Mobilgerät geschont werden soll.

7.3.2 Fachliche Probleme des Konzepts

Zu der mangelnden Genauigkeit gibt es andere Schwächen, welche den Nutzen der App unter Umständen mindern.

Ein Problem ist die benötigte Internetverbindung, um mit Google Prediction zu kommunizieren. Da eine Indoor Navigation oft fürs Ausland sinnvoll wäre, fielen hier in den meisten Fällen Roaming Gebühren an. Abgeschwächt wird dieses Problem, da an immer mehr öffentlichen Räumen freie WLAN Netze verfügbar gemacht werden. Bekanntestes Beispiel hierfür sind die HotSpots, welche in Abschnitt 3.1.6 bereits erwähnt wurden.

Die Navigation mithilfe der entwickelten App ist zudem erst möglich, sobald das Gelände mindestens einmal komplett erfasst wurde. Diese Schwachstelle ist grundlegend für das genutzte Fingerprinting Verfahren und wird von allen Anwendungen geteilt, die auf diese Weise navigieren.

7.3.3 Technische Probleme

Ebenso leidet diese Arbeit, wie jede andere Arbeit, welche Fingerprinting als Verfahren nutzt, an der aufwändigen Kalibrierung innerhalb der Trainingsphase. Eine große Herausforderung für einen tatsächlichen Alltagsbetrieb ist, die Referenzdaten innerhalb der Gebäude zu erfassen, in welchen navigiert werden soll, im Speziellen auch den Gebäudeplan zu beschaffen und in die einzelnen Punkte zu unterteilen. Das Gebäude muss idealerweise in ein Raster gleichgroßer Quadrate unterteilt werden, in deren Mittelpunkt mehrere Messungen für Referenzdaten durchgeführt werden. Im Rahmen dieser Arbeit musste auf ein hochgenaues Vermessungs-Verfahren verzichtet werden, um innerhalb der Zeitvorgaben zu bleiben. Ebenso wäre eine Messung an verschiedenen Tagen und zu verschiedenen Zeiten sinnvoll. Es ist anzunehmen, dass die beobachteten Positionierungsfehler durch eine verbesserte Referenzdatensammlung hätten reduziert werden können.

Auch muss mit Sensorungenauigkeiten gerechnet werden, welche durch die App kompensiert werden müssen. Dies gilt sowohl für die WLAN-Signalstärke-Messung, als auch für die ursprünglich geplante Inertialnavigation. Sei es ein Störsignal in der Umgebung, weswegen zeitweise die Messwerte verfälscht werden oder spezielle Veränderungen, die den WLAN-Empfang beeinträchtigen, wie ein neues Möbelstück, Umstrukturierung der Einrichtung oder andere Menschen um den Nutzer herum. All diese Dinge und vieles mehr beeinflussen den Empfang der WLAN Signale. Magnetische Gegenstände beeinträchtigen zusätzlich die Daten des eingebauten Kompasses.

Eine weitere Schwachstelle der Inertialnavigation ist die Sensibilität der Sensoren, wenn das Mobilgerät nicht genau gerade vor dem Körper gehalten wird. Ein Kippen oder Drehen, verändert die ermittelte Blickrichtung und beeinflusst zudem die Genauigkeit der Schrittmessung. Es konnte jedoch in den Tests festgestellt werden, dass diese Ungenauigkeiten zu vernachlässigen sind, da zum einen von der korrekten Haltung des Mobilgerätes ausgegangen werden kann und zum anderen die Inertialnavigation lediglich zur Unterstützung geplant war.

Neben der Internetverbindung spielt die Internetgeschwindigkeit eine weitere Rolle. Für die Tests wurde eine Ortsabfrage alle 0.5 Sekunden gestartet und lieferte schnelle Antworten, sodass eine quasi Echtzeit Positionsangabe möglich war. Diese Zeit ließe sich bei Bedarf anpassen, um detailliertere Werte zu erhalten. Hierzu ist jedoch eine zuverlässige, schnelle Internetverbindung nötig und auch die Belastung für den Akku stiege drastisch an.

Während der Implementierung mussten an verschiedenen Stellen Abstriche in der

Robustheit des Systems gemacht werden, sodass der Prototyp nur auf die unbewusste Fehlbedienung reagieren kann und beispielsweise automatisch die WLAN Verbindung einschaltet, falls diese nicht zuvor aktiviert wurde oder zwischenzeitig ausgefallen sein sollte. Auch auf die häufigsten Fehlermeldungen von Google Prediction konnte reagiert werden, wenn auch nur über die Kontrolle an einem angeschlossenen Computer. Eine entsprechende Anzeige für den Nutzer wurde nicht implementiert. Ebenso wurden keine Maßnahmen zur Vorbeugung gegen absichtliches Fehlbedienen umgesetzt.

7.4 Zusammenfassung

Ziel der Arbeit war es, einen Prototypen zu entwickeln, welcher auf Basis von WLAN-Fingerprints und maschinellem Lernen eine Ortsbestimmung durchführen kann, die anschließend durch Sensordaten validiert werden sollte. Die letzte Anforderung hiervon konnte nicht vollständig umgesetzt werden, sodass für den Anwendungsfall lediglich die Positionsbestimmung mittels Fingerprintings und maschinellem Lernen bleibt. Auch musste auf die Implementierung der tatsächlichen Navigation von einem Ort zum anderen verzichtet werden, welche idealerweise hätte umgesetzt werden können.

Dennoch ist die Entwicklung als Erfolg anzusehen, da die Positionsbestimmung auf eine neue Art realisiert wurde, die in bisherigen Arbeiten nicht zu finden war und vielversprechende Ergebnisse erzielt werden konnten. Aufgrund des genutzten Google Prediction Dienstes, welcher erst seit kurzem zur Verfügung steht, konnte eine leichtgewichtige Lösung entwickelt werden, der die meisten bekannten maschinellen Lernverfahren zur Verfügung stehen, ohne dass hierdurch zusätzlicher Entwicklungsaufwand entstand. Da die Lernverfahren in dem Google Prediction Dienst vorhanden sind, fand eine Ressourcensubstitution statt, bei der verhältnismäßig wenig Akkuverbrauch für den Aufbau einer Internetverbindung gegen eine große Rechenleistung auf Seiten von Google Prediction getauscht wurde. Diese Arbeit hat gezeigt, dass eine komplexe Anwendung, wie die Indoor Navigation, auch mit Mobilgeräten möglich ist und diese gleichzeitig die Vertraulichkeit von sensiblen Nutzerdaten gewährleisten kann. Durch eine präzisere Vermessung von Testgelände und Fingerprints, sowie die Vollendung der Validierung durch Sensordaten, scheint eine Genauigkeit von 3 Metern in 90% der Fälle durchaus machbar. Diesen Eindruck festigen die verwandten Arbeiten aus Kapitel 4. In diesem Kapitel wurden die Ergebnisse der durchgeführten Tests und Analysen vorgestellt und diskutiert, im folgenden Kapitel werden Vorschläge gemacht, wie diese Arbeit fortgeführt werden könnte und durch welche Technologien sie ergänzt und verbessert werden kann.

8 Ausblick und Zusammenfassung

Nachdem im vorherigen Kapitel die erzielten Ergebnisse analysiert und Probleme aufgezeigt wurden, folgt in diesem Kapitel ein kurzer Ausblick auf interessante und vielversprechende Techniken, durch die ein zusätzlicher Nutzen aus der App entsteht oder die erreichte Genauigkeit verbessert werden kann. Zunächst wird auf zusätzliche Techniken eingegangen, woraufhin eine Alternative in der Bedienung von Google Prediction vorgestellt wird. Abschließend folgt in diesem Kapitel eine Zusammenfassung der Arbeit.

8.1 Ausblick

Bei den Tests zu dieser Arbeit konnte eine Genauigkeit von zwei bis drei Metern in 14% der Fälle erreicht werden. Dies ist nicht ausreichend für eine raumgenaue Indoor Navigation. Aus diesem Grund werden in den folgenden Abschnitten Technologien und Anpassungen vorgeschlagen, um die Genauigkeit zu erhöhen und ebenso den Alltagsnutzen einer Indoor Navigation zu steigern. Es soll vor allem das Potential einer Indoor Navigation betont werden, welches von immer mehr Unternehmen erkannt wird. Im März 2012 soll auf der *Droidcon*¹ ein Vortrag gehalten werden über eine Navigation innerhalb von U-Bahnhöfen, möglich soll dies aufgrund des Barometers sein, welches in dem aktuell verkauften Smartphones der Firma Google eingebaut ist [Wag12]. Diese und andere Neuigkeiten regen dazu an, das Thema Indoor Navigation auch in Zukunft aktiv zu erforschen.

In den folgenden Abschnitten soll sich auf Technologien und Ansätze beschränkt werden, welche direkt an den Prototypen dieser Arbeit anschließen. Begonnen wird mit einer Datenbank für zusätzliche Informationen, woraufhin ergänzende Techniken vorgestellt werden und Verbesserungsvorschläge in der aktuellen Benutzung von Google Prediction.

8.1.1 Datenbank mit POIs

Der kommerzielle Nutzen einer App zur Indoor Navigation liegt in den zusätzlichen Informationen, die durch die App bereitgestellt werden. So können Personen eines Unternehmens in der Datenbank mit ihren jeweiligen Büros verknüpft werden. Ebenso können für Einkaufszentren oder Flughäfen die einzelnen Läden hinterlegt werden, eventuell ergänzend mit den angebotenen Produkten, sodass diese auf Stichworte durchsucht wer-

¹<http://de.droidcon.com/>

den können. Eine solche Datenbank ließe sich beliebig erweitern und würde mit wachsendem Bestand den Nutzen der App maximieren.

8.1.2 Augmented Reality

Das Handy permanent still vor dem Körper zu halten ist zwar die intuitivste Nutzung bei einer Navigation, häufig sieht man jedoch auch Apps, die eine *Augmented Reality*-Ansicht bieten [Sto11]. In dieser Ansicht wird das Bild, welches durch die Handykamera erfasst wird, auf dem Display ausgegeben und durch technische Nachbearbeitungen ergänzt, beispielsweise um Läden oder Personen in der Nähe. Dieser Ansatz scheint für eine Navigation praktisch und hilfreich, denn auch die einzuschlagende Richtung oder das Ziel selbst könnten auf dem Display visualisiert werden.

8.1.3 Barometer

Bei vielen Navigationslösungen wird als ein Problem die Ermittlung des Stockwerkes in Gebäuden genannt. Daher lohnt sich ein Blick auf neuere Mobilgeräte, welche mit einem eingebauten Barometer verkauft werden. An der Beuth Hochschule für Technik in Berlin wird bereits an einem entsprechenden Projekt geforscht [Wag12].

8.1.4 Soziale Netzwerke

Die Entwickler von *Magic Map* (Abschnitt 4.3) ergänzten ihre Software durch ein soziales Netzwerk, in welchem Nutzer ihren eigenen Standort anzeigen lassen konnten, sodass andere Nutzer zu ihnen navigiert wurden. Dieser Ansatz scheint von großem Nutzen zu sein und würde sich vermutlich in jeder Indoor Navigation als hilfreich erweisen.

8.1.5 Google Prediction Update-Funktion

Die Navigation mittels Fingerprinting basiert hauptsächlich auf der empfangenen Signalstärke von WLAN-Netzen in der Umgebung. Es gibt jedoch eine Vielzahl von Gründen, weswegen für bekannte Netzwerke im Laufe der Zeit andere Signalstärken gemessen werden können. Denkbar wäre dies zum Beispiel wegen neuen Leistungskonfigurationen, neuer Hardware, Gebäudeumbauten und einhergehender Störquellen oder einfach unterschiedlicher Netzauslastungen. Bei einmaligen Schwankungen wird versucht diese durch die Mittelwert-Berechnung auszugleichen, dauerhafte Veränderungen würden jedoch nicht vom System ausgeglichen werden können.

Google Prediction bietet zusätzlich zur reinen Anfrage im Betrieb auch eine Update-Funktion. Mittels dieser Update-Funktion kann erreicht werden, dass Google Prediction die neuen Werte den vorhandenen hinzufügt, zugeordnet dem Ort, welcher zusätzlich angegeben werden muss. Dies ermöglicht einen kontinuierlichen Lernprozess. Nutzer müssten dies nicht manuell erledigen, sondern die App könnte im Hintergrund ein solches Update anstoßen und den zuvor ermittelten Aufenthaltsort als zu lernenden Ort

übergeben. Diese Methode hat jedoch den Nachteil, dass die Vorhersagen von Google Prediction falsch sein könnten und dadurch nun falsche Werte den Orten zugewiesen würden. Eine Möglichkeit dieses Problem zu entschärfen, wäre, den Nutzer zu einem gewissen Zeitpunkt zu fragen, wo er sich befindet und nur die so validierten Werte an Google Prediction zu schicken. Eine solche Abfrage könnte einmal pro Nutzung des Services stattfinden, zum Beispiel am Anfang oder auch automatisch angestoßen, wenn festgestellt wird, dass die Vorhersagen nur einen geringen Wahrscheinlichkeitswert haben. Dieser Ansatz würde jedoch ebenfalls nicht die lange Kalibrierungszeit im Vorfeld ersetzen können, sondern sorgt lediglich für Aktualität der Daten.

8.2 Zusammenfassung

Ziel dieser Arbeit war es, einen Prototypen zu entwerfen, welcher auf einem Smartphone mittels sensorgestützter WLAN-Fingerprint Positionsbestimmung und maschinellen Lernverfahren eine Indoor Navigation ermöglicht. Dieses Ziel konnte nicht vollständig erreicht werden, was hauptsächlich der strikten Zeitvorgabe dieser Arbeit geschuldet ist.

Zunächst verdeutlichten Anwendungsbeispiele den Nutzen einer Indoor Navigation, woraufhin die benötigten technischen und methodischen Grundlagen zu einer solchen vorgestellt wurden. Hierbei zeichnete sich bereits ab, dass für die angestrebte Positionsbestimmung innerhalb von Gebäuden am ehesten die WLAN-Technik geeignet ist, in Kombination mit dem Fingerprinting-Verfahren. Anschließend fand eine Vorstellung verwandter Arbeiten statt, welche sich mit einigen der relevanten Techniken befasst haben. Nach einer Analyse der Ergebnisse der unterschiedlichen Arbeiten, wurde anhand der Beispielszenarien eine Anforderungsanalyse und der Entwurf einer App zur Indoor Positionsbestimmung erstellt. Es folgte die Implementierung des Prototyps MyLIN und die Darstellung einiger Probleme in der Entwicklung. Abschließend wurden im vorherigen Kapitel die unterschiedlichen Ergebnisse der einzelnen Tests mit dem Prototypen vorgestellt und bewertet.

Es wurde versucht, mithilfe von den im Testgerät eingebauten Sensoren eine Inertialnavigation zu ermöglichen, welche die berechneten Aufenthaltsorte validieren sollte. Diese Arbeit konnte lediglich zeigen, dass dies im Rahmen der Möglichkeiten ist, jedoch musste auf die eigentliche Implementierung verzichtet werden. Somit konnte die Genauigkeit der Ortsbestimmung lediglich anhand der WLAN-Fingerprints bestimmt werden, welche über das Internet von Google Prediction ausgewertet werden. Es konnte gezeigt werden, dass die Positionsbestimmung eines Mobilgerätes anhand von WLAN-Fingerprints möglich ist, ohne diese Daten auf dem Mobilgerät selbst auszuwerten und gleichzeitig den Aufenthaltsort geheim zu halten, indem anonymisierte Orts-IDs genutzt wurden. Dieser Art der Indoor Navigation ist, den Recherchen nach zu urteilen, derzeit einzigartig und ermöglicht eine Indoor Navigation bereits auf Leistungsschwachen Geräten, sofern diese über eine WLAN-Verbindung verfügen.

Literaturverzeichnis

- [AAH⁺09] ANAGNOSTOPOULOS, Theodoros ; ANAGNOSTOPOULOS, Christos ; HADJIEFTHYMIADES, Stathes ; KYRIAKAKOS, Miltos ; KALOUSIS, Alexandros: Predicting the location of mobile users: a machine learning approach. In: *Proceedings of the 2009 international conference on Pervasive services*, 2009, S. 65–72
- [ABC⁺03] ANASTASI, G. ; BANDELLONI, R. ; CONTI, M. ; DELMASTRO, F. ; GREGORI, E. ; MAINETTO, G.: Experimenting an Indoor Bluetooth-based Positioning Service. In: *Proceedings of the 23 rd International Conference on Distributed Computing Systems Workshops*, 2003
- [App12] APPLE: *iOS Dev Center*. <https://developer.apple.com/devcenter/ios/index.action>. Version: 2012. – Zuletzt abgerufen am 10.03.2012
- [BB05] BRUNATO, Mauro ; BATTITI, Roberto: Statistical learning theory for location fingerprinting in wireless LANs. In: *Comput. Netw. ISDN Syst.* 47 (2005), S. 825–845
- [BE05] BADEA, Vlad ; ERIKSSON, Rikard: *Indoor navigation with pseudolites (fake GPS sat.)*, Linköping University, Diplomarbeit, 2005
- [BF95] BARSHAN, B. ; F., Durrant-Whyte H.: Inertial navigation systems for mobile robots. In: *Robotics and Automation, IEEE Transactions*, 1995
- [BFS09] BFS: *Anwendungen von Infrarot (IR-)Strahlung*. <http://www.bfs.de/de/uv/ir/anwendungen.html>. Version: 2009. – Zuletzt aufgerufen am 10.03.2012
- [BH12] BUNSE, Christian ; HÖPFNER, Hagen: Analyse des Zusammenhangs zwischen Energiebedarf, Dienstgüte und Performanz bei der Ressourcensubstitution in Softwaresystemen. In: HALANG, Wolfgang A. (Hrsg.): *Herausforderungen durch Echtzeitbetrieb*. Springer Berlin Heidelberg, 2012 (Informatik aktuell), S. 101–110
- [Boo07] BOOGAARD, H. C. d.: *Towards indoor navigation for a multimedia platform*, University of Twente, Enschede, The Netherlands, Diplomarbeit, 2007
-

- [BP00] BAHL, P. ; PADMANABHAN, V.N.: 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, 2000*
- [BS11] BLUETOOTH SIG, Inc.: *A Look at the Basics of Bluetooth Wireless Technology*. <http://www.bluetooth.com/Pages/Basics.aspx>. Version: 2011. – Zuletzt aufgerufen am 10.03.2012
- [Cav05] CAVALIERI, Salvatore: Outdoor Location Detection Based On Signal Propagation Models. In: *Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference, 2005*
- [Dic10] DICTIONARYOF COMPUTING, TheFreeOnline: *Machine Learning*. <http://dictionary.reference.com/browse/machinelearning>. Version: 2010. – Zuletzt aufgerufen am 10.03.2012
- [EGC⁺10] ENCK, William ; GILBERT, Peter ; CHUN, Byung-Gon ; COX, Landon P. ; JUNG, Jaeyeon ; MCDANIEL, Patrick ; SHETH, Anmol N.: TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones. In: *9th USENIX Symposium on Operating Systems Design and Implementation, 2010*
- [Elm02] ELMENREICH, Wilfried: *Sensor Fusion in Time-Triggered Systems*, Technische Universität Wien, Diplomarbeit, 2002
- [Emn02] EMNID, TNS: *Hier geht's lang mit den Location-based Services*. Studie, 2002
- [Fie00] FIELDING, Roy T.: *Architectural Styles and the Design of Network-based Software Architectures*, University of California, Irvine, Diplomarbeit, 2000
- [Fuc09] FUCHSS, Thomas: *Mobile Computing: Grundlagen und Konzepte für mobile Anwendungen*. Carl Hanser Verlag München, 2009. – 134 S.
- [Gar11a] GARMIN: *What is GPS?* <http://www8.garmin.com/aboutGPS/>. Version: 2011. – Zuletzt aufgerufen am 10.03.2012
- [Gar11b] GARTNER: *Gartner Says Sales of Mobile Devices Grew 5.6 Percent in Third Quarter of 2011; Smartphone Sales Increased 42 Percent*. <http://www.gartner.com/it/page.jsp?id=1848514>. Version: 2011. – Zuletzt aufgerufen am 10.03.2012
- [Gar11c] GARTNER: *Hype Cycles 2011*. <http://www.gartner.com/technology/research/hype-cycles/>. Version: 2011. – Zuletzt aufgerufen am 10.03.2012
-

-
- [Gle11] GLEIM, Daniel: *Im Androidmarket durchgeführte Suche*. <https://www.androidpit.de/de/android/search?st=Navigation&filter=a>. Version: November 2011. – Zuletzt aufgerufen am 10.03.2012
- [GNU91] GNU: *GNU General Public License, Version 2*. <http://www.gnu.org/licenses/gpl-2.0.html>. Version: 1991. – Zuletzt aufgerufen am 10.03.2012
- [Goo10] GOOGLE: *What is Android?* <http://developer.android.com/guide/basics/what-is-android.html>. Version: 2010. – Zuletzt aufgerufen am 10.03.2012
- [Goo11a] GOOGLE: *Creating a Sentiment Analysis Model*. https://developers.google.com/prediction/docs/sentiment_analysis?hl=de. Version: 2011. – Zuletzt aufgerufen am 10.03.2012
- [Goo11b] GOOGLE: *Go indoors with Google Maps 6.0 for Android*. <http://googlemobile.blogspot.com/2011/11/go-indoors-with-google-maps-60-for.html>. Version: 2011. – Zuletzt aufgerufen am 10.03.2012
- [Goo11c] GOOGLE: *Google Prediction API*. <http://code.google.com/intl/de-DE/apis/predict/>. Version: 2011. – Zuletzt aufgerufen am 10.03.2012
- [Goo11d] GOOGLE: *SensorManager*. <http://developer.android.com/reference/android/hardware/SensorManager.html>. Version: 2011. – Zuletzt aufgerufen am 10.03.2012
- [Goo11e] GOOGLE: *Structuring the Training Data*. <http://code.google.com/intl/de-DE/apis/predict/docs/developer-guide.html>. Version: 2011. – Zuletzt aufgerufen am 10.03.2012
- [Goo12a] GOOGLE: *Activity*. <http://developer.android.com/reference/android/app/Activity.html>. Version: 2012. – Zuletzt aufgerufen am 10.03.2012
- [Goo12b] GOOGLE: *Application Fundamentals*. <http://developer.android.com/guide/topics/fundamentals.html>. Version: 2012. – Zuletzt aufgerufen am 10.03.2012
- [Goo12c] GOOGLE: *Security and Permissions - Security Architecture*. <http://developer.android.com/guide/topics/security/security.html#arch>. Version: 2012. – Zuletzt abgerufen am 05.03.2012
-

- [Goo12d] GOOGLE: *SensorEvent*. <http://developer.android.com/reference/android/hardware/SensorEvent.html>. Version: 2012. – Zuletzt aufgerufen am 10.03.2012
- [Gro99] GROUP, Network W.: *Hypertext Transfer Protocol – HTTP/1.1*. <http://www.w3.org/Protocols/rfc2616/rfc2616.html>. Version: 1999. – Zuletzt aufgerufen am 10.03.2012
- [GSH07] GROSSMANN, U. ; SCHAUCH, M. ; HAKOBYAN, S.: The Accuracy of Algorithms for WLAN Indoor Positioning and the Standardization of Signal Reception for different Mobile Devices. In: *International Journal of Computing, Vol. 6, Is-sue 1 (2007), 103-109.*, 2007
- [GSHD07] GROSSMANN, U. ; SCHAUCH, M. ; HAKOBYAN, S. ; DALHAUS, M.: RSSI based WLAN Indoor Positioning with Personal Digital Assistants. In: *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2007. IDAACS 2007. 4th IEEE Workshop, 2007, S. 653 – 656*
- [gyr12] GYROSCOPE.COM: *Gyroscope*. <http://gyroscope.com/>. Version: 2012. – Zuletzt aufgerufen am 10.03.2012
- [Har07] HARMS, Prof. Jörg M.: Mobile Navigationsgeräte - Marktzahlen und Trends. In: *CeBIT Pressekonferenz des BITKOM BITKOM, 2007*
- [HB00] HIGHTOWER, Jeffrey ; BORRIELLO, Gaetano: SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength. In: *UW CSE Technical Report, 2000*
- [HB01] HIGHTOWER, J. ; BORRIELLO, G.: Location systems for ubiquitous computing. In: *Computer 34 , Issue:8 (2001), S. 57–66*
- [HFH⁺09] HALL, Mark ; FRANK, Eibe ; HOLMES, Geoffrey ; PFAHRINGER, Bernhard ; REUTEMANN, Peter ; WITTEN, Ian H.: The WEKA Data Mining Software: An Update. In: *SIGKDD Explorations 11 (2009), S. 10–18*
- [HH06] HAZAS, M. ; HOPPER, A.: Broadband Ultrasonic Location Systems for Improved Indoor Positioning. In: *Mobile Computing, IEEE Transactions 5 , Issue:5 (2006), S. 536–547*
- [HKD07] HUBER, Thomas ; KREUZER, Johannes ; DIEMER, Robert: Mobiles Monitoring: Energiebedarf von Sensoren und Smartphone für die Verarbeitung und Übertragung relevanter Daten auf einen Server. In: *Mobiles Computing in der Medizin, 2007*
- [HLCY10] HAN, Dongsoo ; LEE, Minkyu ; CHANG, Laeyoung ; YANG, Hyunil: Open radio map based indoor navigation system. In: *Pervasive Computing and*
-

-
- Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference, 2010*
- [IEE05] IEEE: *IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements*. 2005
- [IHS04] IBACH, Peter ; HÜBNER, Tobias ; SCHWEIGERT, Martin: *MagicMap - Kooperative Positionsbestimmung über WLAN*. In: *Chaos Communication Congress Proceedings, 2004*
- [III2] INSIDE-INTERMEDIA: *HTC Desire*. http://www.inside-handy.de/handys/htc_desire/2043_komplett.html. Version: 2012. – Zuletzt aufgerufen am 10.03.2012
- [Int07] INTEL: *Intel and 802.11 Helping Define 802.11n and other Wireless LAN Standards*. http://www.intel.com/standards/case/case_802_11.htm. Version: 2007. – Zuletzt aufgerufen am 10.03.2012
- [Job08] JOBMANN, Christoph: *Ganzzahlige Programmierung und kernbasierte Lernverfahren für das Anschlussicherungsproblem*, Institut für Numerische und Angewandte Mathematik der Georg-August-Universität zu Göttingen, Diplomarbeit, 2008
- [Kal96] KALBACHER, Martin: *Ultraschallgestützte Navigation und Wegplanung in bekannter Umgebung*, Universität Stuttgart, Diplomarbeit, 1996
- [Küp05] KÜPPER, A.: *Location-based Services: Fundamentals and Operation*. 1st. John Wiley & Sons, 2005. – 392 S.
- [KYJ⁺01] KEE, Changdon ; YUN, Doohee ; JUN, Haeyoung ; PARKINSON, Bradford ; PULLEN, Sam ; LAGENSTEIN, Tom: *Centimeter-Accuracy Indoor Navigation Using GPS-Like Pseudolites*. In: *GPS World 12 (2001)*, S. 1–2
- [Loc12] LOCATIONS, Hotspot: *Hotspot-Locations*. <http://www.hotspot-locations.de/>. Version: 2012. – Zuletzt aufgerufen am 10.03.2012
- [LSW09] LE, Manh Hung V. ; SARAGAS, Dimitris ; WEBB, Nathan: *Indoor Navigation System for Handheld Devices*, Worcester Polytechnic Institute Worcester, Massachusetts, USA, Diplomarbeit, Oktober 2009
- [Mau09a] MAUTZ, Rainer: *The Challenges of Indoor Environments and Specification on some Alternative Positioning Systems*. In: *Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop, 2009*, S. 29 – 36
-

- [Mau09b] MAUTZ, Rainer: Innovative Techniken in der Innenraumpositionierung. In: *Intergeo 22.-24. September 2009, Emerging Technologies - Positionierung*, 2009
- [MD07] MANLEY, E.D. ; DEOGUN, J.S.: Location Learning for Smart Homes. In: *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference*, 2007
- [Mic00] MICROSOFT: RADAR. <http://research.microsoft.com/en-us/projects/radar/>. Version: 2000. – Zuletzt aufgerufen am 10.03.2012
- [MIT00] MIT: The Cricket Location-Support System. In: *6th ACM International Conference on Mobile Computing and Networking (ACM MOBICOM)*, 2000
- [MT05] MADHAVAPEDDY, Anil ; TSE, Alastair: Study of Bluetooth Propagation Using Accurate Indoor Location Mapping. In: *The Seventh International Conference on Ubiquitous Computing (UbiComp, 2005, S. 105–122*
- [Ndi94] NDILI, Awele: GPS pseudolite signal design. In: *ION-GPS-94*, 1994
- [Nil98] NILSSON, Nils J.: *Introduction to Machine Learning*. Department of Computer Science Stanford University, 1998
- [NSM09] NASIRI, Steve ; SACHS, David ; MAIA, Michael ; INVENSENSE INC (Hrsg.): *Selection and integration of MEMS-based motion processing in consumer apps*. InvenSense Inc, 2009
- [Ora12] ORACLE: *Lernen über Java-Technologie*. <http://java.com/de/about/>. Version: 2012. – Zuletzt aufgerufen am 10.03.2012
- [PBDT05] PRIYANTHA, Nissanka B. ; BALAKRISHNAN, Hari ; DEMAINE, Erik D. ; TELLER, Seth: Mobile-Assisted Localization in wireless sensor networks. In: *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, 2005
- [Pel12] PELKMANN, Thomas: *Forrester-Studie Die Mobile-Strategie für 2016*. http://www.cio.de/knowledgecenter/mobile_it/2304803/index.html. Version: 2012. – Zuletzt aufgerufen am 10.03.2012
- [Qub11] QUBULUS: *The Indoor Positioning Market Shaped: List of Indoor Positioning Companies*. <http://www.qubulus.com/2011/06/07/the-indoor-positioning-market/shaped-list-of-indoor-positioning-companies/>. Version: 2011. – Zuletzt aufgerufen am 10.03.2012
- [Rei04] REINERT, Denise: *Prototypische Implementation eines datenschutzfreundlichen Location Based Service*, Universität Regensburg, Diplomarbeit, 2004
-

-
- [RIM12] RIM: *BlackBerry Developer*. <https://bdsc.webapps.blackberry.com/devzone/>. Version: 2012. – Zuletzt abgerufen am: 05.03.2012
- [RN04] RUSSELL, Stuart ; NORVIG, Peter: *Künstliche Intelligenz: Ein moderner Ansatz*. Pearson Studium, 2004
- [Sac10] SACHS, David: *Sensor Fusion on Android Devices: A Revolution in Motion Processing*. <http://www.youtube.com/watch?v=C7JQ7Rpwn2k>. Version: 2010. – Zuletzt aufgerufen am 10.03.2012
- [Sky12] SKYHOOK: *Coverage*. <http://www.skyhookwireless.com/location-technology/coverage.php>. Version: 2012. – Zuletzt aufgerufen am 10.03.2012
- [SLTL10] SHEU, Shiann-Tsong ; LI, Meng-Shin ; TSAI, Yi-Hsueh ; LIN, Po-Min: *Indoor Wireless Positioning Technology*. In: *Communications and Networking in China (CHINACOM), 2010 5th International ICST Conference, 2010*
- [Sto11] STOCKBURGER, Lukas: *The Augmented Reality*. <http://www.theaugmentedreality.de/>. Version: 2011. – Zuletzt aufgerufen am 10.03.2012
- [TO12] T-ONLINE: *HotSpot Location Finder*. http://www.hotspot.de/content/hs_finden.html. Version: 2012. – Zuletzt aufgerufen am 10.03.2012
- [Unk09] UNKNOWN: *Versailles heute / heutiger Grundriss*. <http://www.versailles-forum.de/showthread.php?tid=311&page=2>. Version: 2009. – Zuletzt aufgerufen am 10.03.2012
- [Wag12] WAGNER, Roland: *Indoor and underground orientation and navigation by using the barometer sensor*. <http://de.droidcon.com/session/indoor-and-underground-orientation-and-navigation-using-barometer-sensor>. Version: 2012. – Zuletzt aufgerufen am 10.03.2012
- [WFH11] WITTEN, I.H. ; FRANK, E. ; HALL, M.A.: *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier Science, 2011 (The Morgan Kaufmann Series in Data Management Systems). – ISBN 9780080890364
- [WHFaG92] WANT, Roy ; HOPPER, Andy ; FALCÃO, Veronica ; GIBBONS, Jonathan: *The active badge location system*. In: *ACM Trans. Inf. Syst.* 10 (1992), S. 91–102
- [Wil02] WILDPACKETS: *Converting Signal Strength Percentage to dBm Values*. http://www.wildpackets.com/elements/whitepapers/Converting_Signal_Strength.pdf. Version: 2002. – Zuletzt aufgerufen am 10.03.2012
-

- [WJH97] WARD, A. ; JONES, A. ; HOPPER, A.: A New Location Technique for the Active Office. In: *Personal Communications, IEEE 4* , Issue:5 (1997), S. 42–47
- [WM11] WEKA-MEDIA: *Weka IT-Management Infrarot-Kommunikation*. <http://www.weka.de/it-management/1100--~aktuelles~news~news.html>. Version: 2011. – Zuletzt aufgerufen am 10.03.2012
- [WMW05] WILLIG, A. ; MATHEUS, K. ; WOLISZ, A.: Wireless Technology in Industrial Networks. In: *Proceedings of the IEEE 93* , Issue:6 (2005), S. 1130–1151
- [WW00] WINTER, Jürgen ; WENGERTER, Christian: High Resolution Estimation of the Time of Arrival for GSM Location. In: *Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st*, 2000
- [XLZL10] XU, Liqiang ; LIU, Xingchuan ; ZHANG, Sheng ; LIN, Xiaokang: Improved fingerprint algorithm for WLAN-based vehicle positioning. In: *2010 International Conference On Computer Design And Applications (ICDDA 2010)*, 2010
- [XZ09] XUE, Ming ; ZHU, Changjun: A Study and Application on Machine Learning of Artificial Intelligence. In: *Artificial Intelligence, 2009. JCAI '09. International Joint Conference, 2009*
-

Eidesstattliche Erklärung

Ich versichere, dass ich die vorstehende Arbeit selbstständig und ohne fremde Hilfe angefertigt und mich anderer als der im beigefügten Verzeichnis angegebenen Hilfsmittel nicht bedient habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht.

Ich bin mit einer Einstellung in den Bestand der Bibliothek des Fachbereiches einverstanden.

Hamburg, den _____ Unterschrift: _____