

Bachelorarbeit

Max Neuwirt

Evaluation eines Indoor-Positioning-Systems im Kontext
autonomer Systeme

Max Neuwirt

Evaluation eines Indoor-Positioning-Systems im Kontext autonomer Systeme

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Angewandte Informatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Martin Becke
Zweitgutachter: Prof. Dr. Marina Tropmann-Frick

Eingereicht am: 17. Dezember 2021

Max Neuwirt

Thema der Arbeit

Evaluation eines Indoor-Positioning-Systems im Kontext autonomer Systeme

Stichworte

Indoor-Positionierung, autonome Systeme, WLAN-Positionierung, Fingerprinting

Kurzzusammenfassung

Durch voranschreitende Entwicklungen im Bereich der künstlichen Intelligenz und autonomer Systeme wird die Automatisierung verschiedener industrieller Prozesse immer häufiger zum Diskussionspunkt [38][54]. Teil dieser Automatisierung ist der Einsatz autonomer mobiler Systeme. Um sowohl in kritischen Anwendungsfällen oder zwecks Optimierung eine kontinuierliche Überwachung der autonomen mobilen Systeme zu gewährleisten als auch im Falle von Defekten sofort eingreifen zu können, muss der Aufenthaltsort dieser Systeme stets bekannt sein. Dies resultiert in einer zunehmenden Bedeutung von standortbasierten Diensten und damit ebenfalls in einem Bedarf an Positionierungssystemen, die für diese Anwendungsfälle geeignet sind. Die Positionierung innerhalb von Gebäuden kann aufgrund von Umgebungsbedingungen nicht mit bereits bekannten satellitengestützten Positionierungssystemen [48] umgesetzt werden. Dies hat zur Entwicklung von Indoor-Positionierungssystemen [42] geführt. In dieser Arbeit wird das Indoor-Positionierungssystem Framework for Internal Navigation and Discovery (FIND) vorgestellt. Es wurde eine Evaluation bezüglich der Nutzbarkeit von FIND für autonome mobile Systeme durchgeführt. Wesentliches Argument für die Nutzbarkeit ist die Genauigkeit der Positionsbestimmung. Um eine These bezüglich der Nutzbarkeit aufstellen zu können, wird die Position eines autonomen mobilen Systems in einem virtuellen Raster bestimmt. Dabei wird die Granularität des Rasters variiert, bis die Positionsbestimmung nicht mehr korrekt ist. Die Experimente zeigen, dass FIND mobile Geräte in einem Raster mit einer Größe von 18 Quadratmeter lokalisieren kann.

Max Neuwirt

Title of Thesis

Evaluation of an Indoor-Positioning-System in the context of autonomous systems

Keywords

Indoor-Positioning, autonomous systems, WLAN-Positioning, Fingerprinting

Abstract

Due to advancing development of artificial intelligence and autonomous systems, the automation of industry processes becomes more frequently the center of discussion [38]. Part of this automation is the use of mobile autonomous systems. In order to monitor these mobile autonomous systems in critical environments or to optimize workflows as well as act immediately in case of malfunction, the location of a autonomous system must always be known. This results in a growing importance of location based services and with this in positioning systems that are suited for described applications. While satellite-based positioning [48] systems are already working for outdoor use cases, the indoor environment raises new challenges. To tackle these challenges, Indoor-Positioning-Systems [42] were developed. In this work, the Indoor-Positioning-System Framework for Internal Navigation and Discovery (FIND) is presented. An evaluation of the usability of FIND in the context of autonomous systems was made. One of the essential arguments for the usability is the precision of the positioning. To set up a thesis regarding the usability, there have to be an evaluation of the positioning precision that FIND has to offer. The precision was measured by a virtual grid with changing granularity. Granularity was reduced until FIND wasn't able to distinguish different areas of the grid. Experiments show that FIND was able to distinguish grid areas with 18 square meters in size.

Inhaltsverzeichnis

Abbildungsverzeichnis	vii
Tabellenverzeichnis	viii
Abkürzungen	ix
1 Einleitung	1
1.1 Motivation	1
1.2 Fragestellung und Abgrenzung	2
1.3 Struktur der Arbeit	2
2 Grundlagen	3
2.1 Positionierungssysteme	3
2.1.1 Messprinzipien zur Entfernungsbestimmung	4
2.1.2 Methoden der Positionierung	6
2.1.3 Technologische Einteilung für Positionsbestimmung	10
2.2 WLAN-Signalausbreitung	15
2.3 Framework for Internal Navigation and Discovery	17
2.3.1 Funktionsweise	18
2.4 Robot Operating System	21
2.5 Das autonome mobile System	21
2.6 Verwandte Arbeiten	23
3 Anwendungsszenario	25
3.1 System Design	25
3.1.1 System Anwendung	28
3.2 Hindernisse	29
3.3 Implementierung	32
3.3.1 Benutzeroberfläche: Loomo	32
3.3.2 Benutzeroberfläche: FIND-Applikation	33

3.3.3	Technologien	35
4	Experiment	36
4.1	Aufbau	37
4.2	Durchführung	38
4.3	Erwartung	39
5	Evaluation	41
5.1	Ergebnisse	41
5.2	Diskussion	44
6	Fazit	51
7	Ausblick	53
	Literaturverzeichnis	54
	Selbstständigkeitserklärung	60

Abbildungsverzeichnis

2.1	IBSS-Netzwerktopologie [7]	12
2.2	BSS-Netzwerktopologie [7]	12
2.3	ESS-Netzwerktopologie [7]	12
2.4	Freiraumdämpfung [31]	15
2.5	Effekte der Wellenausbreitung [50]	16
2.6	Segway Loomo Robot (Loomo)	22
3.1	System Architektur	26
3.2	Loomo Benutzeroberfläche	32
3.3	FIND App Benutzeroberfläche	34
4.1	Skizze des virtuellen Rasters	36
5.1	Erster Experimentdurchlauf	45
5.2	Zweiter Experimentdurchlauf	47
5.3	Erster Experimentdurchlauf mit Einsatz des Filters	49

Tabellenverzeichnis

2.1	Positionierungsmethoden Überblick [31]	9
2.2	Positionierungstechnologien Überblick [31]	14
2.3	Typische Dämpfungswerte bei einer Signalfrequenz von 2,4 GHz, [43]	16
5.1	Ergebnisse ohne Filter	42
5.2	Ergebnisse mit Filter	42

Abkürzungen

AoA Angle of Arrival.

AP Access Point.

API Application Programming Interface.

BSS Basic Service Set.

BSSID Basic Service Set Identifier.

CoO Cell of Origin.

ESS Extended Service Set.

FIND Framework for Internal Navigation and Discovery.

GNSS Global Navigation Satellite System.

IBSS Independent Basic Service Set.

IEEE Institute of Electrical and Electronics Engineers.

IPS Indoor-Positionierungssystem.

MAC Media Access Code.

RFID Radio-Frequency Identification.

ROS Robot Operating System.

RSS Received Signal Strength.

RSSI Received Signal Strength Indicator.

RTT Round-Trip-Time.

SSID Service Set Identifier.

Abkürzungen

TDoA Time Difference of Arrival.

ToA Time of Arrival.

UWB Ultra-Wideband.

WLAN Wireless Local Area Network.

1 Einleitung

1.1 Motivation

Mit dem Einsatz von autonomen mobilen Systemen, haben standortbasierte Dienste (engl. Location Based Services, LBS) [30] eine immer höhere Bedeutung in der Industrie. LBS sind Dienste, die den Standort eines Gerätes benötigen, um bestimmte Funktionalitäten anbieten zu können. LBS finden sich im täglichen Leben in Form von beispielsweise Navigations- oder Notfalldiensten wieder. Zudem gewinnen LBS immer mehr Bedeutung in der Industrie, wo diese in automatisierten Lagerhallen zur Standortermittlung beziehungsweise Steuerung von autonomen mobilen Systemen oder dem Verfolgen von Lagerbeständen genutzt werden. Eine der wichtigsten Komponenten der LBS ist das Positionierungssystem. Dieses erlaubt die Standortbestimmung der mobilen Geräte und macht damit LBS erst möglich. Die bekanntesten Positionierungssysteme lassen sich als Global Navigation Satellite System (GNSS) [28] zusammenfassen [31]. Das sind Positionierungssysteme, die speziell für den Outdoorbereich entwickelt wurden und erfordern deshalb eine kontinuierliche Sichtverbindung zu entsprechenden GNSS-Satelliten. Für den Einsatz innerhalb von Gebäuden oder anderen überdachten Bereichen sind solche Outdoor-Positionierungssysteme nicht geeignet, da eine Sichtverbindung zu Satelliten nicht gegeben ist. Deshalb muss hier auf alternative Technologien zurückgegriffen werden. Indoor-Positionierungssysteme (Indoor Positioning Systems, IPS) sind Positionierungssysteme, die für den Einsatz innerhalb von Gebäuden entwickelt wurden und stellen somit das Pendant zu den Outdoor-Positionierungssystemen dar.

1.2 Fragestellung und Abgrenzung

Im Hauptteil der hier vorliegenden Thesis wird die Frage gestellt, ob und mit welcher Genauigkeit die Position des autonomen mobilen Systems mittels FIND bestimmt werden kann.

Dafür wird ein FIND-Client für ein autonomes mobiles System implementiert, mit dem gezeigt werden soll, dass FIND zusammen mit diesem System eingesetzt werden kann. Dieser Client ermöglicht die Positionsbestimmung mittels FIND. Zusätzlich wird ein Anwendungsfall, welcher FIND als Teilsystem einsetzt, umgesetzt. Dafür wird unter anderem ein FIND-Client für ein Smartphone implementiert. Ziel des Anwendungsfalles ist es zu zeigen, ob FIND in Anwendungen oder Applikationen eingesetzt werden kann. Des Weiteren werden Experimente zur Evaluation der Genauigkeit von FIND durchgeführt, um die anfangs Fragestellung beantworten zu können.

In der vorliegenden Arbeit werden keine Optimierungen an FIND vorgenommen. Im Kapitel 7 werden lediglich mögliche Verbesserungen für FIND vorgeschlagen. Diese sind bei Bedarf in einer weiterführenden Arbeit umzusetzen.

1.3 Struktur der Arbeit

Diese Arbeit wird folgendermaßen strukturiert: Zunächst werden in Kapitel 2 die Grundlagen zum Verständnis dieser Arbeit geschaffen. Dabei wird ein Überblick über verschiedene Messprinzipien, Positionierungsmethoden und Technologien der Positionierung gegeben. Außerdem werden FIND, weitere verwendete Software und das mobile autonome System vorgestellt. Im Unterkapitel 2.6 werden aktuelle Arbeiten zum Thema fingerprint-basierte Indoor-Positionierungssysteme vorgestellt und kurz erläutert. In Kapitel 3 wird das Anwendungsszenario erklärt sowie ein vertiefender Einblick in den Aufbau des Systems gegeben. Kapitel 4 erklärt die Experimente zur Evaluierung von FIND. Dabei wird auf den Aufbau, die Durchführung und auf die Erwartungen bezüglich der Ergebnisse eingegangen. Im anschließenden Kapitel 5 werden die Ergebnisse aus den Experimenten vorgestellt und diskutiert. Die Kapitel 6 und 7 schließen die Arbeit mit einem Fazit und einem Ausblick ab.

2 Grundlagen

Im folgenden Kapitel werden die notwendigen Grundlagen zum Verständnis der vorliegenden Bachelorarbeit vermittelt. Zunächst wird ein allgemeiner Einblick in das Thema Positionierungssysteme gegeben. Dabei wird bereits ein Fokus auf die für diese Arbeit relevanten Technologien gelegt. Weiter wird das Framework for Internal Navigation and Discovery (FIND) vorgestellt und die Funktionsweise erläutert. Anschließend wird eine kurze Einleitung zu dem in dieser Arbeit verwendeten Robot Operating System (ROS) gegeben. Daraufhin wird das autonome mobile System vorgestellt, mit dem sich in dieser Arbeit befasst wurde. Schließlich werden noch aktuelle Forschungsarbeiten aus dem Bereich der Indoor-Positionsermittlung vorgestellt.

2.1 Positionierungssysteme

Für die Bestimmung der Position eines mobilen Gerätes wird ein Positionierungssystem benötigt. Abhängig davon, ob sich das mobile Gerät in einem Gebäude oder im Freien befindet, wird jeweils eine andere Kategorie von Positionierungssystem eingesetzt. Das Outdoor-Positionierungssystem dient der Positionsbestimmung im Freien, während das Indoor-Positionierungssystem (IPS) die Positionsbestimmung innerhalb von Gebäuden übernimmt. Die häufiger verwendeten Positionierungssysteme sind für den Outdoor-Bereich zuständig und lassen sich als satellitengestützten Positionierungssysteme beziehungsweise GNSS zusammenfassen [31]. Da für diese Systeme eine direkte Sichtverbindung (engl. Line-of-Sight, LoS) zu GNSS-Satelliten benötigt wird, ist eine Positionierung in überdachten Bereichen ungenau bis unmöglich. Hier kommen die speziell für dieses Problem entwickelten IPS zum Einsatz. IPS setzen sich in der Regel aus mindestens zwei Hardwarekomponenten zusammen: einem Sender und einem Empfänger. Eine dieser Komponenten ist dabei das mobile Gerät. Je nach Verteilung der Aufgaben zwischen den Komponenten lassen sich IPS in vier verschiedene System-Topologien aufteilen [32]:

Die **Self-Positioning System-Topologie** nutzt das mobile Gerät als Empfänger. Der Empfänger misst Signale von Sendern mit bekannten Positionen und berechnet daraus die Position direkt auf dem mobilen Gerät. Sendet der Empfänger die Messergebnisse an einen Server, der daraufhin die Position daraus berechnet, wird von einem **Indirect-Remote-Positioning System** gesprochen.

Bei einem **Remote-Positioning System** ist der Sender das mobile Gerät und die Empfänger haben fixe bekannte Positionen. Die Messergebnisse aller Empfänger-Komponenten werden gesammelt und an einen Server geschickt, der dann die Position des mobilen Gerätes berechnet. Werden die Messergebnisse nach dem Sammeln zurück zum mobilen Gerät geschickt, wird von einem **Indirect-Self-Positioning System** gesprochen.

Bei der Auswahl eines Indoor-Positionierungssystems müssen verschiedene Aspekte in Betracht gezogen werden. Dazu gehören die Komplexität der Gebäudestruktur beziehungsweise die Verteilung und Anordnung der Räume und die daraus resultierenden direkten Sichtverbindungen zwischen Empfänger- und Sender-Komponenten, sowie Schwankungen in der Signalstärke, aber auch Kosten für die Anschaffung neuer Hardware zur Positionierung. Nicht zuletzt sollte, für den konkreten Anwendungsfall, immer feststehen, welche Genauigkeit bei der Positionierung benötigt wird, um die richtige Wahl bei der Positionierungsmethode sowie dem Messprinzip zu treffen.

Im weiteren Verlauf des Kapitels wird ein kurzer Überblick über Messprinzipien, Positionierungsmethoden sowie die gängigsten Technologien zur Positionsbestimmung gegeben. Dabei werden nur die für diese Arbeit relevanten Messprinzipien, Positionierungsmethoden sowie Technologien kurz erläutert.

2.1.1 Messprinzipien zur Entfernungsbestimmung

Mithilfe der hier vorgestellten Messprinzipien kann die Entfernung zwischen einem Sender und einem Empfänger berechnet werden.

Zu den gängigsten Messprinzipien gehören Angle of Arrival (AoA), Time of Arrival (ToA), Round-Trip-Time (RTT), Time Difference of Arrival (TDoA) und Received Signal Strength (RSS).

Das Messprinzip **AoA** basiert auf Winkelmessungen von einfallenden Signalen. Dabei werden die Einfallwinkel von zwei Sendern oder Empfängern mit bekannter Position

gemessen und es werden zwei Geraden gebildet, dessen Schnittstelle die Position des mobilen Gerätes markiert.

Beim **ToA** oder auch Time of Flight (ToF) handelt es sich um eine zeitbasierte Messtechnik. Hier wird die Laufzeit eines Signals vom Sender bis zum Empfänger mit der Ausbreitungsgeschwindigkeit des Signals multipliziert, daraus resultiert die Distanz zwischen Sender und Empfänger.

Bei der **RTT** oder auch Round-Trip Time of Flight (RToF) wird die Laufzeit eines Signals vom Sender zum Empfänger und wieder zurück gemessen. Da hier die Laufzeit auf einem Gerät berechnet wird, ist keine Zeitsynchronisation zwischen Geräten nötig. Bei ToA sowie bei RTT werden mindestens drei Sender oder Empfänger für die Positionsbestimmung benötigt.

Im Gegensatz zu den eben beschriebenen Messmethoden verwendet **TDoA** keine absoluten Signallaufzeiten, sondern arbeitet mit Differenzen in den Ankunftszeiten von Signalen von verschiedenen Empfängern. Dabei werden wie zuvor mindestens drei Empfänger benötigt, die das Signal des mobilen Senders empfangen. Aus den Differenzen wird dann die Position des mobilen Systems berechnet.

Received Signal Strength

Einen bisher ungeachteten Ansatz zur Positionsbestimmung verfolgt die Methode Received Signal Strength Indicator (RSSI). Bei dieser Methode wird mittels der Signalstärke beziehungsweise dem RSSI, versucht, die Position eines mobilen Gerätes zu bestimmen. Dabei wird sich die Tatsache zunutze gemacht, dass die Signalstärke unter anderem mit steigender Entfernung immer weiter abnimmt. Grund hierfür ist die sogenannte Freiraumdämpfung [34]. Die Freiraumdämpfung beschreibt die Reduktion der Stärke von Signalen im freien Raum, also ohne explizite Störung durch Luft oder Reflexion. Stehen nun drei Sender mit bekannter Position zur Verfügung und wird eine gleichmäßige, kugelförmige Ausbreitung des Signals angenommen, kann mittels Lateration die Position vom mobilen Gerät ermittelt werden [31]. Für das Errechnen der Entfernung aus dem gemessenen RSSI-Wert können zwei Methoden eingesetzt werden. Ein Signalausbreitungsmodell oder eine gerätespezifische empirische Kalibrierung. Beim Signalausbreitungsmodell ist keine vorherige Kalibrierung nötig, jedoch eignet sich diese Methode schlecht für heterogene Geräte, da verschiedene Geräte Signale verschieden stark messen und daraus entsprechend verschiedene Entfernungen berechnet werden. Bei einer gerätespezifischen Kalibrierung

werden beispielsweise die Signalstärken an mehreren Positionen mit bekannter Entfernung zum Sender gemessen, um daraus eine Funktion abzuleiten, die die Entfernung an beliebigen Positionen bestimmen kann [31].

Bei einer Positionsbestimmung mit den eben genannten Methoden zur Entfernungsberechnung müssen Hindernisse wie Signalausbreitung, Signaldämpfung, Signalschwankungen, Umwelteinflüsse sowie Multipath-Effekte [44][25] berücksichtigt werden. Die Signalausbreitung kann durch die Gebäudestruktur sehr komplex werden, sodass ein Ausbreitungsmodell nur sehr schwer zu modellieren ist [41]. Des Weiteren sind räumlich und zeitlich bedingte Signalschwankungen zu beobachten, welche unter Umständen in unterschiedlichen Entfernungsberechnungen für dieselben Positionen resultieren, wenn diese zu verschiedenen Zeiten stattfinden. Ebenso können beispielsweise eine hohe Luftfeuchtigkeit und Multipath-Effekte ein Problem darstellen, da sie die Ausbreitung der Signale beeinflussen. Weiterhin sind Messungen von Änderungen in einer dynamischen Umgebung ebenso wie der Ausrichtung der Antenne abhängig [27].

Eine andere Möglichkeit RSSI-Werte zur Positionierung zu verwenden wird Fingerprinting genannt. Beim Fingerprinting wird nicht die Entfernung zu einem Sender berechnet, sondern es werden Messungen von RSSI-Werten an bestimmten Referenzpunkten durchgeführt, woraufhin diese in einer Datenbank gespeichert und später zum Vergleich mit aktuellen Messungen genutzt werden. Aus dem Vergleich kann gefolgert werden, an welchem Referenzpunkt sich das mobile Gerät befindet. Der Referenzpunkt mit der höchsten Übereinstimmung wird dann als Resultat zurückgegeben und dient als Ersatz für Koordinaten. Die Datenbank kann, alternativ zu den eben beschriebenen empirischen Messungen, auch mit einem Signalausbreitungsmodell erzeugt werden.

2.1.2 Methoden der Positionierung

Bei der Betrachtung von Positionierungsmethoden für Indoor-Positionsermittlung lassen sich diese grob in fünf Kategorien einordnen. Dazu gehören die Lateration, Angulation, Nachbarschaft (engl. Proximity), Koppelnavigation (engl. Dead Reckoning) sowie die Szenenanalyse [18][31]. Die Methoden werden im folgenden Abschnitt kurz vorgestellt.

Bei der **Lateration** wird die Position eines mobilen Gerätes mittels Kreisen, deren Radien sich aus der Entfernung von koordinativ bekannten Sender oder Empfängern ergeben, ermittelt. Der Schnittpunkt dieser Kreise markiert dabei die Position des mobilen Gerätes. Für die Positionierung werden mindestens drei Sender beziehungsweise Empfänger

benötigt. Für die Entfernungsmessung können die in Kapitel 2.1.1 vorgestellten Methoden ToA, TDoA, RTT und RSSI eingesetzt werden.

Die **Angulation** ist eine der Lateration verwandte Methode. Ähnlich wie bei der Lateration werden bei der Angulation fixe Stationen benötigt. Im Gegensatz zur Lateration nutzt die Angulation jedoch Winkel anstelle von Entfernungen für die Positionierung. Hierzu werden die Richtungswinkel von mindestens zwei Signalen gemessen, um zwei Geraden zu berechnen, deren Schnittpunkt die Position des mobilen Gerätes markiert. Die Winkelmessung kann mit der in Kapitel 2.1.1 beschriebenen AoA Messung vollzogen werden.

Die intuitivste Methode zur Positionsbestimmung beruht auf Nähe und wird als zellbasiertes Verfahren, **Proximity** oder Cell of Origin (CoO) bezeichnet. Dabei wird ein mobiles Gerät im Abdeckungsbereich eines koordinativ bekannten Radiofrequenzsenders erkannt und die Position des Senders als Position des mobilen Gerätes angenommen.

Die Koppelnavigation (engl. **Dead Reckoning**) verzichtet fast vollständig auf externe Komponenten für die Positionsbestimmung. Stattdessen werden Bewegungsrichtung und Bewegungsgeschwindigkeit, gemessen mit Gyroskop und Accelerometer (Beschleunigungssensor), genutzt, um auf Basis einer vorher gegebenen Position die aktuelle Position zu berechnen.

Szenenanalyse

Bei der Szenenanalyse (engl. Scene Analysis) wird die Position eines mobilen Gerätes anhand von aktuellen Umgebungsdaten ermittelt [18]. Dafür können verschiedene Daten genutzt werden. Die gängigsten Datenquellen sind Kameraaufnahmen, Vermessungen mit Ultraschall oder Signalstärken von verschiedenen Access Points (AP). Dabei spielt es keine Rolle, ob es sich um Wireless Local Area Network (WLAN), Bluetooth, Ultra-Wideband (UWB) et cetera -Signale handelt. Theoretisch sind alle Datenquellen möglich, solange sich diese durch verschiedene Positionen unterscheiden lassen.

Die Szenenanalyse setzt sich in der Regel aus zwei Schritten zusammen. Im ersten Schritt werden Daten über die Umgebung, in der später die Positionierung stattfinden soll, gesammelt und persistiert. Dies kann an vorher ausgewählten Referenzpunkten oder in der gesamten Umgebung stattfinden. Dieser Schritt wird auch als Trainings- oder Offline-Phase bezeichnet und ist essenziell für die Positionierung mittels Scene Analysis. Ergebnis

der Trainingsphase ist eine Datenbank mit Informationen über die Umgebung. Im zweiten Schritt, der Tracking- oder Online-Phase, werden dann Daten an einer unbekannt Position erhoben. Diese Daten werden mit den Daten aus der Datenbank verglichen und es wird eine Vorhersage über die Position des mobilen Gerätes gemacht.

Fingerprinting. Das sogenannte Fingerprinting basiert auf dem Prinzip der Szenenanalyse und gehört zu den beliebtesten Positionierungsmethoden [46]. Die erhobenen Daten sind hier RSSI-Werte sowie die Basic Service Set Identifier (BSSID) der sichtbaren AP. Beim Fingerprinting werden Multipath-Effekte und Signaldämpfung dazu genutzt, um verschiedene Referenzpunkte voneinander unterscheidbar zu machen, damit eine Positionierung bewerkstelligt werden kann. Somit ist das Fingerprinting weder anfällig für Multipath-Effekte noch für Signaldämpfung. Nachteilig ist jedoch der hohe Zeitaufwand für das Erstellen der Datenbank in der Trainingsphase, den alle szenenanalysebasierten Positionierungsmethoden haben. Zudem muss bei einer baulichen Veränderung oder besonders beim Fingerprinting, einer Veränderung in der Sender-Landschaft (hinzufügen oder entfernen von APs) die Datenbank neu aufgebaut werden [43]. Des Weiteren spielt die Heterogenität von Geräten eine große Rolle beim Aufbau der Datenbank. Verschiedene Geräte messen Signale verschieden stark, dies führt zu geräteabhängigen Positionierung. Sprich RSS-Messungen von Geräten, die verschiedene Antennen oder Chips verbaut haben, können unterschiedlich sein, selbst wenn sich diese an derselben Position befinden [20]. Auch bei Messungen mit gleichen Geräten wurden Schwankungen von bis zu 10 dBm beobachtet [52]. Trotz dieser Nachteile ist Fingerprinting eine der beliebtesten Methoden für IPS [31].

In Tabelle 2.1 wurde ein Überblick über die hier beschriebenen Positionierungsmethoden gegeben.

Positionierungsmethode	Messprinzip	Vorteile	Nachteile	Genauigkeit
Lateration	ToA, TDoA, RTT, RSS	keine Trainingsphase	LoS notwendig; Multipath-Effekte	cm - m
Angulation	AoA	keine Trainingsphase	LoS notwendig; Antennenarray notwendig; Multipath-Effekte	cm - m
Szenenanalyse/ Fingerprinting	-	keine Multipath-Effekte; kein LoS notwendig	Trainingsphase notwendig; Datenbank	cm - m/m
Proximity	Zellbasiert	einfache Algorithmen	Position abhängig vom Sender	abhängig von Zellgröße
Dead Reckoning	-	nur lokale Sensoren notwendig	relative Positionierung; großer Fehlerdrift	m

Tabelle 2.1: Positionierungsmethoden Überblick [31]

2.1.3 Technologische Einteilung für Positionsbestimmung

Indoor-Positionierung kann mit verschiedenen Technologien erfolgen. Eine Übersicht über die gängigsten Technologien wird im folgenden Kapitel gegeben. Eine grobe Einteilung der Technologien erfolgt in: optisch-, akustisch-, inertialsensor-, magnetisch- und radiofrequenzbasierte.

Optische Technologien nutzen Licht zur Bestimmung der Position. Der Informationsaustausch mithilfe von Licht wird als Visible Light Communication (VLC) bezeichnet. Licht emittierende Dioden (LED) werden genutzt, um Informationen über ein Codeschema zu übertragen, welches durch schnelles ein- und ausschalten einer LED generiert wird.

Akustische Technologien verwenden Tonsignale zur Entfernungsmessung beziehungsweise Positionsbestimmung. Beispielsweise kann die Differenz der Ankunftszeit eines Radiosignals und eines Tonsignals dazu verwendet werden, die Entfernung zwischen einem Sender und einem Empfänger zu berechnen.

Magnetische Technologien erlauben Positionierung mithilfe von Magnetfeldern. Mit einem Magnetometer können Anomalien in künstlich erzeugten oder natürlich vorkommenden Magnetfeldern gemessen werden [40]. Die Anomalien lassen das Magnetfeld an verschiedenen Punkten anders aussehen und ermöglichen damit eine Unterscheidung von Referenzpunkten (siehe 2.1.2).

Positionsbestimmung kann auch mittels **Inertialsensoren** erfolgen. Zur Positionsbestimmung wird dafür ein Gyroskop und ein Accelerometer verwendet. Das Gyroskop gibt die Richtung an, in die das mobile Gerät gerichtet ist und erlaubt damit die Bestimmung einer Richtungsänderung. Das Accelerometer erlaubt die Messung der Beschleunigung eines mobilen Gerätes.

Radiofrequenz Technologien

Ein großer Vorteil von Radiofrequenzsignalen gegenüber optischen und akustischen Signalen ist, dass Radiofrequenzsignale Hindernisse wie beispielsweise Wände durchdringen können. Damit sind Positionierungssysteme, die auf dieser Technologie basieren, nicht an einzelne Räume gebunden. Die radiofrequenzbasierten Technologien lassen sich weiter in Bluetooth, RFID, UWB und WLAN unterteilen.

Bluetooth ist bei vielen mobilen Geräten wie Smartphones bereits verfügbar, somit fallen Anschaffungskosten in den meisten Fällen weg. Selbst wenn dies nicht der Fall ist, sind die Kosten sowie der Stromverbrauch für Bluetooth-Geräte eher gering [21]. Ein Problem von Bluetooth-Signalen ist der Frequenzbereich, in dem diese versendet werden. Dieser liegt zwischen 2.402 und 2.480 GHz und ist somit im selben Frequenzbereich wie WLAN-Signale, was in einer Anfälligkeit für Interferenzen resultiert [31].

Radio-Frequency Identification (RFID) ist eine weitere Radiofrequenz Technologie, die zur Positionsbestimmung eingesetzt werden kann. Bei RFID werden drei Frequenzbereiche genutzt: die Niederfrequenz mit 125–134 kHz (Low Frequency, LF), die Hochfrequenz mit 13,56 MHz (High Frequency, HF) und die Ultrahochfrequenz mit 860–960 MHz (Ultra-High Frequency, UHF) [36]. Die Nachteile von RFID sind die geringe Reichweite von 1 bis 2 m der kostengünstigen passiven RFID-Tags, die eine Positionsbestimmung nur an vorher ausgewählten Kontrollpunkten erlaubt oder im Vergleich zu den passiven Tags höheren Kosten der aktiven Tags [19]. In beiden Fällen wird zusätzliche Hardware benötigt, mit der Personen oder Geräte ausgestattet werden müssen.

Bei **Ultrabreitband (engl. Ultra-Wideband, UWB)** werden elektromagnetische Wellen in Form von kurzen Impulsen versendet. Dabei nutzt UWB eine große Bandbreite von über 500 MHz und einen ebenso großen Frequenzbereich von 30 MHz bis 10,6 GHz (in Deutschland [1]) [19][51]. Die große Bandbreite verleiht UWB einige Vorteile, wie beispielsweise weniger Stromverbrauch, hohe Datenraten, Resistenz gegenüber Interferenz und leichtere Hindernisdurchdringung etc. [51].

Ein **Wireless Local Area Network (WLAN)** ist ein lokales Funknetz, welches in der Regel auf dem Institute of Electrical and Electronics Engineers (IEEE) 802.11 WLAN-Standard basiert. Genau genommen können unter dem Begriff WLAN alle Technologien verstanden werden, die ein lokales Funknetz aufbauen. Ein WLAN-Netzwerk besteht aus mindestens zwei Netzwerkteilnehmern: einem WLAN-Client und einer WLAN-Basisstation, die als Wireless Access Point (WAP) oder einfach nur Access Point (AP) bezeichnet wird. Diese Art von WLAN-Topologie wird als Basic Service Set (BSS) bezeichnet. Weitere WLAN-Topologien sind der Independent Basic Service Set (IBSS) und der Extended Service Set (ESS) [7].

Bei einer IBSS-Topologie oder auch Ad-hoc-Netzwerk besteht das WLAN-Netzwerk aus mindestens zwei WLAN-Clients [7]. Die Kommunikation findet dabei direkt zwischen den WLAN-Clients statt. Es wird kein AP benötigt. Da der Standard für IBSS-Netzwerke

nur sehr grob definiert ist, gibt es bis heute Probleme, Geräte verschiedener Hersteller miteinander zu verbinden.

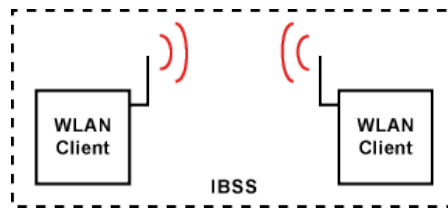


Abbildung 2.1: IBSS-Netzwerktopologie [7]

Die BSS-Topologie ist die am weitesten verbreitete Art ein WLAN-Netzwerk aufzubauen. Bei dieser Topologie gibt es mindestens einen WLAN-Client und mindestens einen AP. Der AP stellt die Verbindung beziehungsweise den Übergang von drahtgebundenem zu drahtlosem Netzwerk dar. Die WLAN-Clients, die sich innerhalb der vom AP zur Verfügung gestellten Funkzelle befinden, haben Zugriff auf das Netzwerk.

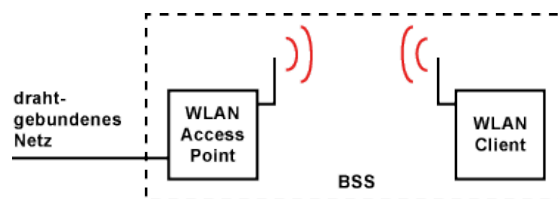


Abbildung 2.2: BSS-Netzwerktopologie [7]

Werden zwei oder mehr BSSs über eine Funkverbindung zusammengeschlossen, wird von einem ESS gesprochen. Damit kann die Reichweite eines lokalen Netzes erweitert werden, ohne auf Verkabelung zurückgreifen zu müssen.

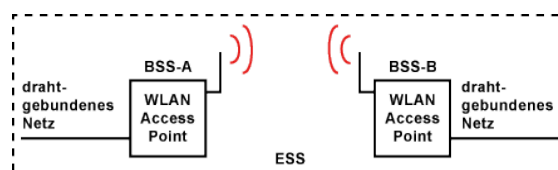


Abbildung 2.3: ESS-Netzwerktopologie [7]

Für die Positionsermittlung mittels Fingerprinting wird eine BSS- oder ESS-Topologie benötigt. Dies liegt daran, dass für die Positionsbestimmung APs mit einer fixen Position vorhanden sein müssen. Bei Netzwerken mit mobilen APs wie bei einem Ad-hoc-Netzwerk

ist dies nicht gegeben, was eine Wiedererkennung von Fingerprints an derselben Position unmöglich macht, da die Fingerprints an einem Referenzpunkt abhängig von den Positionen der WLAN-Clients sind.

Der IEEE 802.11 Standard wurde vom IEEE entwickelt und ist die Basis für WLAN. Seit der ersten Veröffentlichung des Standards wurden mehrere Erweiterungen herausgebracht. Jede Erweiterung besitzt eigene Eigenschaften, die beispielsweise das verwendete Frequenzband betreffen.

Heute werden WLAN-Signale über das Industrial, Scientific and Medical (ISM)-Band meistens im Frequenzband von 2,4 GHz (2400-2483,5 MHz) und 5 GHz (5150-5350 MHz) ausgesendet [31]. Dabei hat jedes Frequenzband seine Vor- und Nachteile. Allgemein kann gesagt werden, je höher die Frequenz ist, desto geringer ist die Reichweite [31]. Somit hat das 2,4 GHz Band die größere Reichweite, da es Hindernisse verlustarmer überwinden kann. Andererseits wird dieses Band von verschiedenen Geräten beziehungsweise Funktechnologien genutzt, was in einer höheren Anfälligkeit für Interferenzen resultiert [31]. Das 5 GHz Band hat im Vergleich zum 2,4 GHz Band eine deutlich höhere Datenübertragungsrate [31]. Diese spielt für IPS jedoch keine Rolle, da für die Positionierung keine Daten übertragen werden müssen. Der große Nachteil des 5 GHz-Bandes liegt darin, dass dieses stärker von Wänden gedämpft wird. Dies resultiert in einer geringeren Reichweite des Signals.

Ein WLAN-AP versendet circa alle 100 ms kleine Datenpakete (Beacons). Diese enthalten die BSSID beziehungsweise die Media Access Code (MAC)-Adresse [49] des AP und den Netzwerknamen, also die SSID. Dies ermöglicht es, unbekannte Netzwerke auf beispielsweise einem Smartphone zu sehen und eine Verbindung zu diesen herzustellen. Die einzelnen APs können über die BSSID identifiziert werden. Zusätzlich kann ein mobiles Gerät beim Empfangen des Signals, dessen Signalstärke messen. Diese wird in der Einheit Dezibel-Milliwatt (dBm) angegeben. Die Signalstärke wird als RSSI bezeichnet. Gemessene RSSI-Werte von WLAN-Signalen befinden sich in der Regel im Bereich von -20 dBm und -100 dBm. Zusammen mit der BSSID, die die Identifizierung von APs ermöglicht, wird der RSSI für das Fingerprinting eingesetzt. Aufgrund von Signaldämpfung werden die RSSI-Werte bei stärkerer Ausbreitung des Signals immer kleiner. Innerhalb von Gebäuden erreichen handelsübliche Access Points eine Reichweite von 20 bis 100 Metern [19]. Draußen kann eine Reichweite von mehreren Kilometern erreicht werden.

In Tabelle 2.2 wird ein Überblick über die hier beschriebenen Positionierungstechnologien gegeben.

Technologie		Vorteile	Nachteile	Kosten
Optisch	Infrarot	geringer Stromverbrauch	Störanfällig; raumgebunden; geringe Reichweite	gering
	sichtbares Licht	geringer Stromverbrauch; hohe Genauigkeit	Störanfällig; raumgebunden	gering
Akustisch	hörbar	geringe Kosten	Störanfällig; störend für Menschen	gering
	Ultraschall	hohe Genauigkeit	Störanfällig (Interferenz)	hoch
Radiofrequenz	WLAN	verwendet vorhandene Infrastruktur	hoher Stromverbrauch; anfällig für Signalschwankungen	mittel
	Bluetooth	geringer Stromverbrauch	anfällig für Signalschwankungen und Interferenz	gering
	RFID	Tags können überall positioniert werden	geringe Reichweite; Genauigkeit abhängig von Tags Positionen	mittel
	UWB	kein Multipath-Effekt; hohe Genauigkeit; geringer Stromverbrauch	Hardware teuer	hoch
Magnetfeld	natürlich	keine Kosten	Störanfällig	keine
	künstlich	geringe Schwankungen	Störanfällig	hoch
Sensoren	Inertialsensoren	kein Multipath; unabhängig von externen Komponenten	Fehlerdrift; Startposition nötig	keine (oft verbaut)

Tabelle 2.2: Positionierungstechnologien Überblick [31]

2.2 WLAN-Signalausbreitung

Bei WLAN-Signalen handelt es sich um elektromagnetische Signale. Diese breiten sich in der Regel kugelförmig aus und haben die Eigenschaft Hindernisse durchdringen zu können. Dennoch gibt es einige Herausforderungen, die bei der Positionierung mithilfe von WLAN-Signalen auftreten können. Diese werden im Folgenden vorgestellt und erklärt.

Während der Ausbreitung verliert ein Signal Energie und wird damit immer schwächer. Diese Abschwächung wird als **Freiraumdämpfung** bezeichnet und beschreibt die Reduktion der Leistungsdichte von elektromagnetischen Wellen im freien Raum [31]. Dies bedeutet, dass eine Reduktion der Leistung unabhängig von Störeinflüssen durch Medien wie Luft, ebenso wie Störungen durch Reflexionen stattfindet. Der Einfluss der Dämpfung ist dabei von der Signalfrequenz abhängig. Die Freiraumdämpfung wird in Dezibel angegeben. In Abbildung 2.4 ist die Freiraumdämpfung für bekannte WLAN-Frequenzen zu sehen.

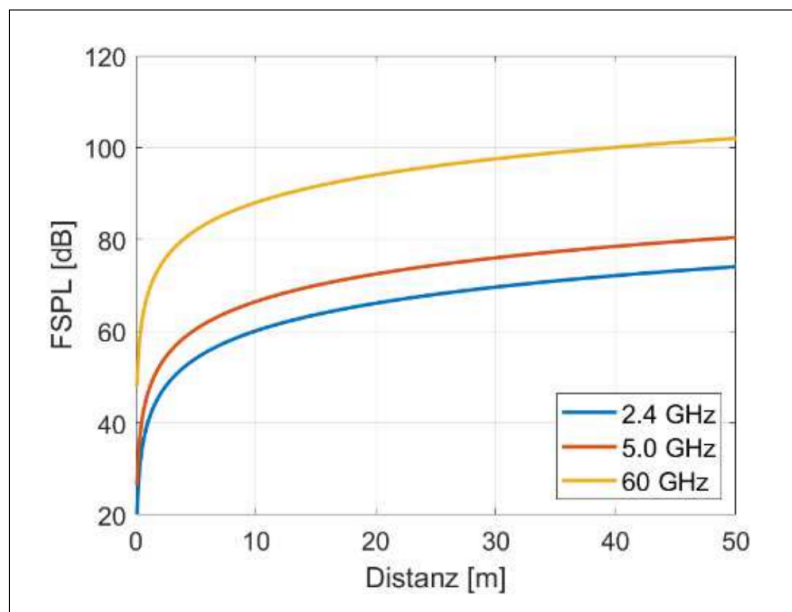


Abbildung 2.4: Freiraumdämpfung [31]

Dabei ist zu erkennen, dass ein Signal bereits kurz nach der Erzeugung deutlich an Energie verliert. Nach diesem starken Energieverlust am Anfang flacht die Kurve ab, dennoch ist zu sehen, dass der Energieverlust mit der Distanz weiter zunimmt. Außerdem ist zu erkennen, dass die Freiraumdämpfung mit steigender Frequenz ebenfalls größer wird.

Zusätzlich zur Freiraumdämpfung können weitere Faktoren zu einer Signalabschwächung führen. Bei der Ausbreitung durchläuft ein Signal verschiedene Medien. Je nach Medium können verschiedene Effekte beim Kontakt mit dem Medium auftreten. Zu diesen Effekten gehören: Absorption, Beugung, Refraktion, Reflexion und Streuung (siehe Abbildung 2.5). Absorption und Reflexion haben einen großen Einfluss auf die empfangene

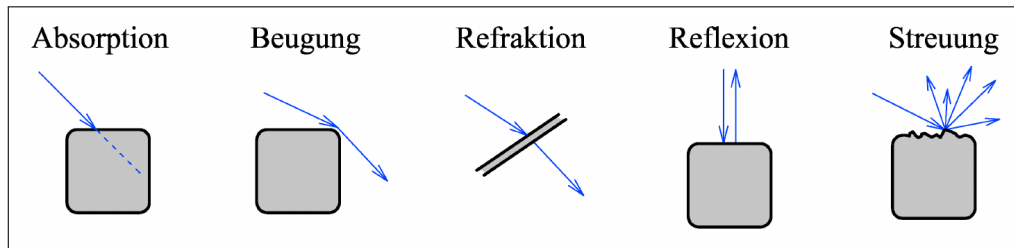


Abbildung 2.5: Effekte der Wellenausbreitung [50]

Signalstärke und Reichweite [31]. Das Ausmaß des Störeinflusses hängt dabei stark vom Medium und seinen Materialeigenschaften ab. In Tabelle 2.3 sind typische Dämpfungswerte von Medien, die oft in Gebäuden vorzufinden sind, dargestellt.

	Dämpfung [dB]
Dünne Mauer	2-5
Ziegelwand	6-12
Betonwand	10-20
Doppelverglasung	25-35
Betondecke	20-40

Tabelle 2.3: Typische Dämpfungswerte bei einer Signalfrequenz von 2,4 GHz, [43]

Multipath beschreibt den Effekt der Auftritt, wenn ein Signal an einem Hindernis reflektiert oder gestreut wird [31]. Das Ausmaß der Reflexion oder Streuung hängt dabei von den Materialeigenschaften und der Geometrie eines Mediums ab. Durch Multipath-Effekte können mehrere Signale vom gleichen Sender bei einem Empfänger ankommen. Wird ein reflektiertes Signal zur Positionsbestimmung verwendet, kann dies in einer falschen Position resultieren, da die zurückgelegte Strecke bei einem reflektierten Signal länger ist als bei einem LoS-Signal. Besonders lateration- und angulationbasierte Positionierungsmethoden sind anfällig für Multipath-Effekte.

Eine weitere Herausforderung bei der Positionsbestimmung mittels WLAN-Fingerprinting sind **Signalschwankungen**. Diese können aus verschiedenen Gründen auftreten. Zum

einen unterliegen WLAN-Signale natürlichen Schwankungen, die dafür sorgen, dass sich Signalstärken auf der gleichen Position im Laufe der Zeit verändern können, diese Schwankungen können zusätzlich durch beispielsweise Temperatur oder Luftfeuchtigkeit verstärkt werden [31]. Zum anderen sorgt der Multipath-Effekt dafür, dass Signale aufgrund von Wänden, Personen oder Objekten reflektiert oder gestreut werden können. Für fingerprintbasierte Positionierung sind Multipath-Effekte nur dann ein Problem, wenn es zu kurzfristigen Änderungen in der Umgebung (Personen, öffnen/schließen von Türen, etc.) kommt. Dies kann in unerwarteten Multipath-Effekten resultieren, die ein Signal verstärken oder abschwächen können und damit die Fingerprints an einer Position verändern. Bei gleichbleibender Umgebung sind Multipath-Effekte kein Problem.

Interferenz beschreibt die Änderung der Amplitude eines Signals durch Überlagerung von zwei oder mehr Wellen mit der gleichen Frequenz [50][31]. Dadurch können sich Wellen verstärken oder auslöschen, was dazu führt, dass die Signalstärke falsch gemessen wird oder das Signal gar nicht gemessen werden kann [31]. Dies kann beispielsweise durch den Einsatz verschiedener Geräte im selben Frequenzkanal, wie es bei ESS der Fall ist, passieren. Die Interferenz kann die Fingerprints an einem Referenzpunkt verfälschen, was zu fehlerhaften Daten in der Datenbank führt oder zu fehlerhaften Daten bei der Evaluierung der Fingerprints während der Positionierung. Beides resultiert in einer inkorrekten Positionsbestimmung.

2.3 Framework for Internal Navigation and Discovery

Bei der in dieser Arbeit behandelten Technologie handelt es sich um ein IPS, welches auf der Methode der Szenenanalyse basiert und dort genauer auf dem Fingerprinting. Das IPS trägt den Namen Framework for Internal Navigation and Discovery (FIND). In diesem Abschnitt wird genauer auf FIND eingegangen. Dabei wird kurz erklärt, was FIND ist und wie genau die Standortermittlung mittels FIND funktioniert.

FIND ist ein Framework, mit dessen Hilfe es möglich ist, verschiedene WLAN-fähige Geräte zu lokalisieren. FIND findet sich in der Kategorie der IPSe wieder, da mittels FIND die Lokalisierung innerhalb von Gebäuden oder von Satellitensignalen abgeschotteten Bereichen möglich ist. Genauer lässt sich FIND den szenenanalysebasierten Methoden und dort dem Fingerprinting zuordnen (siehe Kapitel 2.1.2). Bei der Positionsbestimmung verfolgt FIND den Ansatz der Standortvorhersage mittels Klassifizierung von Radiofrequenzsignalen. Dazu nutzt FIND Signale von WLAN-APs oder Bluetooth-fähigen

Geräten. Die aus FIND ermittelten Standortinformationen können für Automatisierung, Wegfindung, Tracking sowie für weitere verschiedene Anwendungsszenarien eingesetzt werden. So ist es beispielsweise möglich, Geräte in einzelnen Räumen einer Wohnung zu lokalisieren.

Im Kontext von FIND existiert der Begriff “Family”. Mit einer FIND-Family lassen sich Geräte-Familien gruppieren, die sich durch verschiedene Charakteristika unterscheiden. Damit wird der Geräteheterogenität entgegengewirkt. Oft haben verschiedene Geräte unterschiedliche Antennen verbaut. Durch die verschiedenen Antennen werden Radiofrequenzsignale verschieden stark empfangen. Dies führt dazu, dass für eine Messung an derselben Position gegebenenfalls verschiedene Fingerprints gemessen werden. Was wiederum die Positionsbestimmung beeinträchtigt. Geräteheterogenität ist ein bekanntes Problem bei fingerprintbasierten IPS.

Wie FIND Radiofrequenzsignale evaluiert, um damit Standortvorhersagen treffen zu können, wird im folgenden Abschnitt beleuchtet.

2.3.1 Funktionsweise

Zur Lokalisierung evaluiert FIND Fingerprints von WLAN-APs und Bluetooth-Geräten.

WLAN bzw. Bluetooth-fähige Geräte sind in der Lage einen Scan nach APs oder Bluetooth-Geräten durchzuführen. So ein Scan wird in regelmäßigen Intervallen initiiert. Bei jedem dieser Scans empfängt das scannende Gerät einen Beacon von jedem AP, der sich in Reichweite befindet (siehe Kapitel 2.1.3). Ein Beacon enthält eine eindeutige Kennung des sendenden AP. Gleichzeitig misst das scannende Gerät die Signalstärke beim Empfangen eines Beacons, die mit der Entfernung zu diesem AP korreliert. Bei der eindeutigen Kennung handelt es sich um den BSSID. Dieser entspricht in der Regel der MAC-Adresse und ermöglicht eine eindeutige Identifizierung eines WLAN-APs. Die Signalstärke wird durch den RSSI beschrieben. Dabei ist die Signalstärke besser, je höher der Wert ist.

Zusätzlich sei noch gesagt, dass es sich bei APs nicht zwingend um Router handeln muss. Genau wie Router würden sich auch Raspberry Pis [33], Repeater oder Ähnliches eignen. Ein weiterer wichtiger Punkt ist, dass keine aktive Verbindung zu einem AP benötigt wird. Es reicht, wenn sich ein AP in Reichweite des scannenden Gerätes befindet.

Die im Kapitel 2.1.2 beschriebene Funktionsweise von szenenanalysebasierten Positionierungsmethoden lässt sich auf die Funktionsweise von FIND übertragen. Die Lokalisierung von FIND lässt sich ebenfalls grob in zwei Schritten beschreiben. Der erste Schritt kann mit der Trainingsphase gleichgesetzt werden. Hier werden alle vorher festgelegten Referenzpunkte aus dem Bereich, in dem später die Lokalisierung stattfinden soll, angesteuert, woraufhin der Referenzpunkt nach sichtbaren APs gescannt wird. Nach dem Scannen werden die gesammelten Daten zum Training von Klassifizierern des maschinellen Lernens [45] verwendet. Der zweite Schritt kann als Trackingphase bezeichnet werden und beinhaltet die eigentliche Positionsbestimmung.

Während der Trainingsphase wird ein Bereich gewählt, in dem später die Lokalisierung stattfinden soll. Dieser Bereich kann beispielsweise eine Wohnung sein, aber auch jeder andere überdachte und auch nicht überdachte Bereich ist denkbar. Vorausgesetzt wird nur, dass mindestens ein AP in diesem Bereich vorhanden ist, wobei die Lokalisierung genauer ist, wenn mehr APs zur Verfügung stehen. Nach dem Festlegen des Bereiches werden Referenzpunkte ausgewählt, diese könnten beispielsweise einzelne Räume in einer Wohnung sein. An diesen Punkten wird nach den sichtbaren APs gescannt. Ein Scan setzt sich dabei aus mehreren Zyklen zusammen. Bei einem Zyklus werden Beacons empfangen, die von allen sich in der Nähe befindenden APs emittiert werden. Das Ergebnis eines solchen Zyklus ist eine Reihe aus BSSID und RSSI Paaren, die als Fingerprint bezeichnet wird. Sind die Referenzpunkte gewählt, beginnen die Scans. Bei einem Scan wird dem Referenzpunkt ein Bezeichner gegeben. Dieser wird dann zusammen mit dem Ergebnis des Scans an den FIND-Server geschickt. Das Ergebnis eines Scans besteht aus mehreren Fingerprints. Ein Scan sollte mindestens zehn Minuten dauern [6]. So kann sichergestellt werden, dass alle sichtbaren APs katalogisiert wurden.

Nachdem an jedem Referenzpunkt ein Scan ausgeführt wurde, können nun die Klassifizierungsalgorithmen trainiert werden. Trainiert werden die Algorithmen mit den vorher gesammelten Daten. Dabei dienen die Referenzpunkt-Bezeichner als Label für die gesammelten Fingerprints. Die Daten bestehen also aus der BSSID (MAC-Adresse), dem RSSI und dem Label beziehungsweise Bezeichner eines Referenzpunktes [6].

Ziel der Algorithmen ist es, mittels Klassifizierung der Fingerprints den zugehörigen Referenzpunkt vorherzusagen. Für die Vorhersage nutzt FIND mehr als einen Klassifizierungsalgorithmus. Genauer gesagt werden zehn Algorithmen eingesetzt [6], um eine möglichst genaue Vorhersage treffen zu können. Jeder dieser Algorithmen wird trainiert den richtigen Standort vorherzusagen. Um die Ergebnisse der Algorithmen vergleich-

bar zu machen wurde der Youden-Index (auch Youden J) [22] implementiert [6]. Der Youden-Index ist eine Metrik zur Messung der Performanz eines Klassifizierers. Mithilfe des Indexes können Performanz-Werte für jeden Algorithmus für jeden Referenzpunkt berechnet werden. Wird dann eine Standortvorhersage angefragt, wird das Ergebnis des Algorithmus genommen, das den höchsten Youden-Index hat.

Sobald die Trainingsphase abgeschlossen wurde, können Geräte lokalisiert werden. In der Trackingphase wird das mobile Gerät im gescannten Bereich positioniert, woraufhin erneut Scans initiiert werden. Diesmal wird jedoch die Information über den Referenzpunkt nicht benötigt. Es werden lediglich die Fingerprints an den FIND-Server geschickt. Der FIND-Server nutzt dann die trainierten Klassifizierer, um den Standort des Gerätes vorherzusagen.

2.4 Robot Operating System

Das Robot Operating System (ROS) ist ein Framework zur Entwicklung von Software für autonome Systeme.

Ziel von ROS ist es, die Entwicklung von Software für autonome Systeme zu vereinfachen, sowie eine gemeinsame Grundlage für die Forschung zu bieten. Begonnen hat die Entwicklung von ROS im Jahr 2007 [4]. 2008 wurde die Entwicklung dann hauptsächlich von dem von Scott Hassan privat finanzierten Unternehmen, beziehungsweise Robotikforschungsinstitut Willow Garage, vorangetrieben [4]. Sechs Jahre später wurde Willow Garage geschlossen und die Verantwortung für ROS an die neu gegründete Open Source Robotics Foundation übergeben [4].

Die ROS-Entwickler bezeichnen ROS als Open Source, Meta-Betriebssystem. Die Bezeichnung Meta-Betriebssystem bedeutet, dass Betriebssystem-Funktionen wie Hardwareabstraktion, Gerätetreiber, Utilityfunktionen, Interprozesskommunikation und Paketmanagement von ROS angeboten werden [9], jedoch kann ROS kein Betriebssystem ersetzen.

ROS soll die Arbeit mit autonomen Systemen simpler machen. Dafür werden folgende Design-Ziele aktiv verfolgt und so gut wie möglich umgesetzt. Zu den wichtigsten Zielen gehören unter anderem verteilte Datenverarbeitung, Code-Wiederverwendbarkeit, leichte Testbarkeit, Leichtgewichtigkeit und Sprachunabhängigkeit [9][39]. Das primäre Ziel von ROS ist jedoch, eine Umgebung zu schaffen, die durch Kollaboration und Wiederverwendung von Code die Robotikforschung und -entwicklung fördert [9]. Mithilfe dieser Zielsetzung und aktiver Arbeit an dem Framework ist ROS zu einem Standard herangewachsen. Damit findet ROS nicht nur Einsatz in der Forschung, sondern wird auch in der Industrie eingesetzt.

2.5 Das autonome mobile System

Im Rahmen der Bachelorarbeit wird ein autonomes mobiles System eingesetzt, um sowohl die Forschungsfrage zu beantworten, den Anwendungsfall zu implementieren als auch die Experimente aus Kapitel 4 durchzuführen. Bei der Forschungsfrage und auch dem Experiment findet das autonome System seinen Einsatz als Scanner nach verfügbaren

APs. Der Anwendungsfall nutzt sowohl das verbaute WLAN-Modul als auch die Mobilität des autonomen Systems. Beides wird in späteren Kapiteln ausgeführt.

Von dem Experiment sowie dem implementierten Anwendungsfall werden konkrete Anforderungen an die Hardware gestellt. Die verwendete Hardware muss einen hohen Grad an Mobilität sowie Autonomie vorweisen. Genauer muss das System sich autonom bewegen können. Zusätzlich muss ein WLAN-Modul verbaut sein, sowie die Möglichkeit das System um Funktionen zu erweitern. Dies alles wird vom Segway Loomo Robot (Loomo) realisiert, der sich unter anderem durch Mobilität auszeichnet. Des Weiteren bietet der Loomo ein Android-Software-Development-Kit (Android-SDK) [10], mit dessen Hilfe die Funktionen des Loomos erweitert werden können. Außerdem bietet der Loomo mit der Android-SDK die Möglichkeit eine Schnittstelle für ROS zu implementieren, die weitere Funktionalität wie beispielsweise autonomes Fahren ergänzt.

Bei dem Loomo handelt es sich um einen selbst balancierendes autonomes mobiles System (siehe Abb. 2.6). Dieser ist in der Lage, sich mit Geschwindigkeiten von bis zu 8 km/h im Roboter-Modus zu bewegen. Geliefert wird er bereits mit Funktionen, wie Menschenverfolgung und Hinderniserkennung.



Abbildung 2.6: Segway Loomo Robot (Loomo)

2.6 Verwandte Arbeiten

Wie im vorherigen Kapitel bereits erwähnt, ist WLAN eine Technologie, die zur Positionierung eingesetzt werden kann. Da eine WLAN-Infrastruktur in den vielen Gebäuden bereits gegeben ist, ist der Einsatz von WLAN zur Positionierung mit geringen Kosten verbunden. Aus diesem Grund und der Tatsache, dass die WLAN-Infrastruktur weiter ausgebaut wird, gehört WLAN zu den erforschtesten Technologien für die Indoor-Positionierung [31][35][54].

Grob lässt sich die aktuelle Forschung zu Indoor-Positionierungssystemen in fingerprint- und laterationbasierte Methoden aufteilen [24][29]. Da in dieser Arbeit der Fokus auf fingerprintbasierten Indoor-Positionierung liegt, werden in diesem Kapitel hauptsächlich fingerprintbasierte Methoden vorgestellt. Um den Rahmen der vorliegenden Bachelorthesis nicht zu sprengen, wird nur auf einige ausgewählte Forschungsarbeiten eingegangen.

Guo et al. (2019) [24] entwickelten ein System für Smartphones, welches RTT (siehe Kapitel 2.1.1) und RSS (siehe Kapitel 2.1.1) in Kombination mit einem Signalausbreitungsmodell nutzt, um die Distanz zwischen einem Access Point und einem mobilen Gerät zu bestimmen. Anschließend wird mittels Multilateration die Position des mobilen Gerätes ermittelt. Das System erreicht dabei eine Positionsgenauigkeit von ca. 1,435 m bei einer Update-Rate von 0,19 Sekunden, was das System besonders attraktiv für den Einsatz in Real-Time-Szenarien macht. Zwar nutzt das System WLAN zur Positionsbestimmung, ist aber auf den IEEE 802.11-REVmc2 Standard angewiesen, da dieser das FTM-Protokoll (Fine Time Measurement) einführt, welches sehr genaue Distanzmessung mittels RTT erlaubt. Problem dabei ist, dass nur bestimmte APs das FTM-Protokoll unterstützen.

In [35] stellen Ma et al. (2019) ein ranking-basiertes IPS vor, welches aus drei Teilen besteht. Dabei bedeutet ranking-basiert hier, dass anstelle von rohen RSSI-Werten eine Rangfolge von RSSI-Werten von verschiedenen APs verwendet werden. Im ersten Teil wird mithilfe eines genetischen Algorithmus eine Auswahl von bestimmten Access Points getroffen, die in dem Anwendungsbereich zur Verfügung stehen. Dabei werden APs, die ein hohes Rauschen (engl. noise) [49] verursachen, herausgefiltert. Im zweiten Teil werden der Kendall Tau Korrelations-Koeffizient und ein Faltungsnetz (engl. Convolutional Neural Network, CNN) [45] eingesetzt, um die Position eines Nutzers vorherzusagen. Schließlich wird im dritten Teil ein erweiterter Kalman-Filter (engl. extended Kalman Filter, EKF) [45] eingesetzt, um die vorhergesagten Positionen zu glätten und damit

genauere Trajektorien zu ermitteln. Das IPS erreicht eine durchschnittliche Genauigkeit von 1,42 m.

Xue et al. (2020) [52] stellen das IPS HAIL (High-Adaptability Indoor Localization) vor. HAIL nutzt sowohl einen ranking-basierten Ansatz, ähnlich zu [35], als auch absolute RSSI-Werte, um eine robuste ebenso wie akkurate Positionierung bereitzustellen. Dabei setzten Xue et al. auf den Einsatz einer Kombination aus einem neuronalen Netz [45] und dem Weighted K-Nearest Neighbors (WKNN) [26] zur Referenzpunkterkennung. HAIL erreicht eine durchschnittliche Genauigkeit von 0,87 m, wenn dasselbe Gerät sowohl für die Offline- als auch die Onlinephase verwendet wird und eine durchschnittliche Genauigkeit von 1,5 m, wenn heterogene Geräte verwendet werden. Dabei reduziert HAIL den durchschnittlichen Positionierungsfehler um ca. 30% bis ca. 50%.

In [54] kombinieren Zhang et al. ein tiefes neuronales Netz (engl. Deep Neural Network, DNN) [45] mit mehreren Entscheidungsbäumen [45], genannt Decision Forest, für die Positionsermittlung. Dabei werden mittels neuronalem Netz abstrakte Merkmale aus den Fingerprints extrahiert, die dann an die Entscheidungsbäume weitergereicht werden. Die Entscheidungsbäume berechnen anhand der Merkmale die Wahrscheinlichkeiten der Referenzpunkte und geben diese als Resultat zurück. Dieser neuartige Ansatz wurde als Deep Fuzzy Forest getauft. Wobei ähnliche Deep Forests bereits für andere Problemstellungen, wie Bildklassifizierung, eingesetzt wurden. Zhang et al. erreichen mit ihrem Ansatz eine durchschnittliche Positionierungsgenauigkeit von 1,36 m.

3 Anwendungsszenario

In diesem Kapitel wird das Anwendungsszenario beschrieben. Dabei wird der Aufbau und die Implementierung des Systems vorgestellt. Zusätzlich werden noch Hindernisse beschrieben, die während des Entwicklungsprozesses auftraten.

3.1 System Design

Die Grundidee des Anwendungsszenarios ist es, ein System zu entwickeln, welches ermöglicht, einen Loomo per Smartphone zur aktuellen Position des Smartphones fahren zu lassen. Dabei wird vorausgesetzt, dass der Loomo und das Smartphone in einem bekannten Bereich positioniert werden. Ein bekannter Bereich ist dadurch definiert, dass dieser vorher vom System gescannt wurde.

Das entwickelte System ist eine Komposition aus verschiedenen Teilsystemen, die für dieses Anwendungsszenario miteinander verbunden wurden. Zu den Teilsystemen gehören ROS, FIND sowie ein selbst implementierter Service. Dieser wird im Folgenden Location-Service genannt. Dazu seien noch sowohl der Loomo als auch ein Smartphone als Akteure im System genannt. Jedes Teilsystem übernimmt dabei eine Teilfunktion, um zusammen die oben beschriebene Funktionalität bereitzustellen. In der folgenden Abbildung (siehe Abbildung 3.1) ist eine Modellierung des gesamten Systems zu sehen.

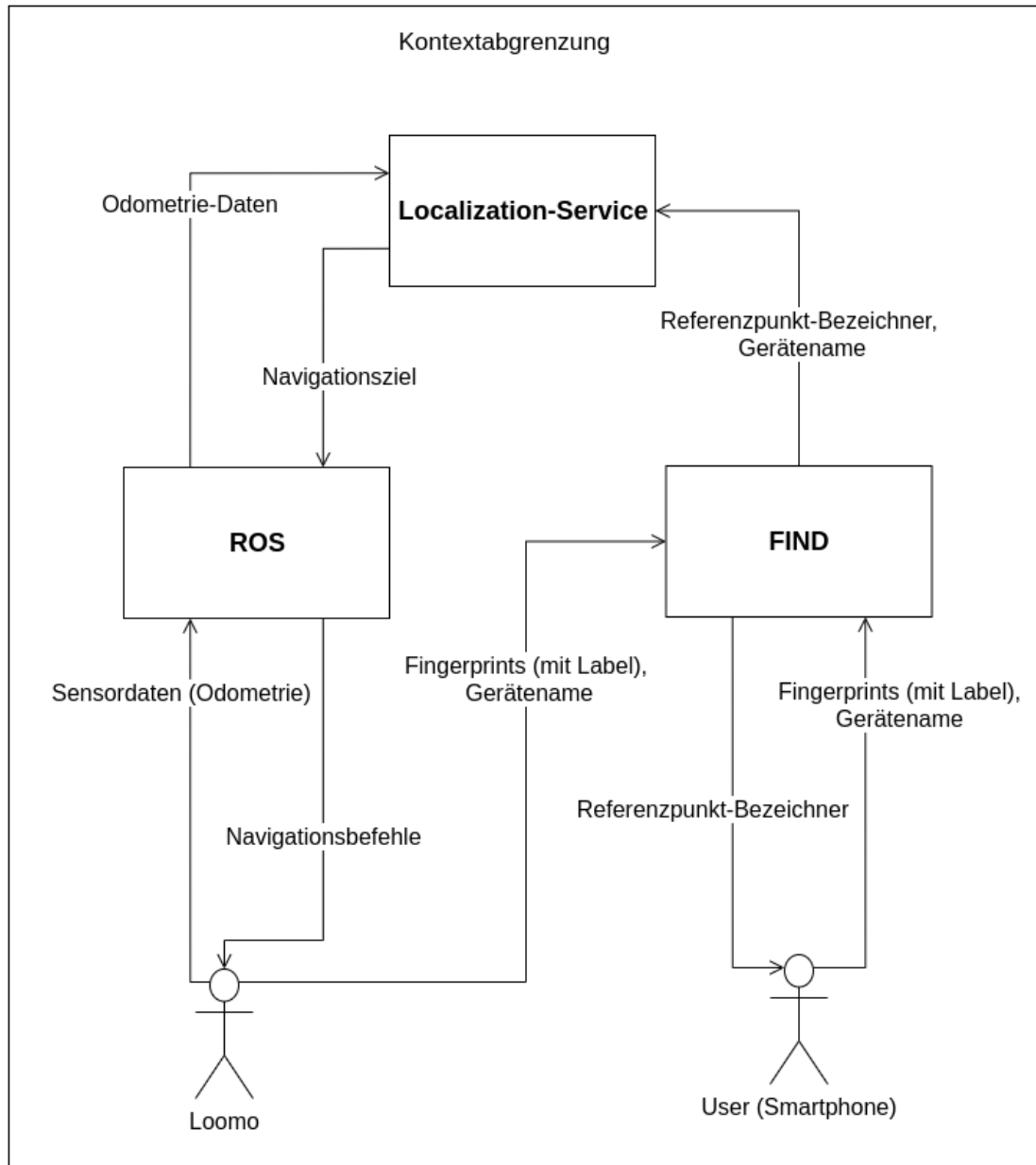


Abbildung 3.1: System Architektur

Teilsysteme

Im Folgenden werden die einzelnen Aufgaben der Teilsysteme genauer beschrieben.

Kern des Systems ist der sogenannte Location-Service. Beim Location-Service handelt es sich um einen selbst implementierten Service. Dieser wird zur Orchestrierung und Bindung der anderen Services genutzt. So bildet der Location-Service den ideellen Mittelpunkt des Systems. Der Location-Service empfängt zum einen Koordinaten von ROS und zum anderen die Standortvorhersagen von FIND. Die Daten aus beiden Subsystemen werden hier in Relation gesetzt und dann als Datenstruktur in einer PostgreSQL-Datenbank [23] persistiert. Außerdem schickt der Location-Service, die für die Navigation benötigten Koordinaten zu ROS, sobald mittels eines Smartphones ein Loomo angefordert wird.

Das ROS übernimmt die Aufgabe der Navigation des Loomos zu einem bestimmten Koordinatenpunkt im festgelegten Bereich. Dafür werden mehrere Funktionen von ROS eingesetzt. Zum einen wird eine SLAM-Implementierung [47] von ROS genutzt, um eine Karte vom festgelegten Bereich zu zeichnen. Diese Karte wird dazu benötigt, um den Loomo innerhalb des Bereiches zu orten und zu navigieren. Die zweite eingesetzte Funktion ist die Navigation des Loomos in der gezeichneten Karte.

Es wurden außerdem zwei weitere Funktionen zu ROS hinzugefügt. Dafür wurden zwei ROS-Nodes [8] implementiert. Dabei handelt es sich bei einer ROS-Node um einen aktiven Prozess. Eine dieser Nodes dient dem Lesen von Koordinaten aus der Odometrie [47] des Loomos und dem Weiterleiten dieser Daten an den Location-Service, wo diese dann persistiert werden. Die andere Node dient dem Weiterleiten von Koordinaten vom Location-Service an ROS, sodass der Loomo zu den gegebenen Koordinaten navigiert werden kann. Beide Nodes nutzen zur Kommunikation mit dem Location-Service Apache Kafka [2].

FIND ist ein weiteres Teilsystem und übernimmt die Vorhersage des Standortes eines Gerätes, wie in Kapitel 2.3 beschrieben.

Loomo und Smartphone implementieren einen FIND-Client. Dieser scannt die Umgebung nach sichtbaren Access Points und sendet die gesammelten Fingerprints dann an FIND. Dafür wurde sowohl für das Smartphone als auch für den Loomo Software entwickelt, die die benötigte Funktionalität zum Sammeln der Fingerprints bietet. Bei der Software für den Loomo handelt es sich um eine Erweiterung der bereits bestehenden Android-Applikation "LoomoCADSService" [13]. Genauer gesagt wurde hier

ein Hintergrund-Service implementiert, der die benötigten Aufgaben erledigt. Für das Smartphone wurde eine prototypische App entwickelt.

Es wurde eine Anbindung an eine Lambda-Architektur implementiert. Dort werden neben den Relationen auch Daten über die einzelnen Access Points sowie empfangene RSSI-Werte persistiert. Diese Daten werden außerhalb dieser Thesis für Analysen verwendet.

3.1.1 System Anwendung

Im Folgenden werden die Schritte beschrieben, mit denen das System einsatzbereit gemacht sowie eingesetzt wird.

Zu Beginn wird ein Bereich gewählt, in dem die Lokalisierung von Smartphone und Loomo stattfinden soll. In diesem Bereich werden dann Referenzpunkte gewählt und Scans für FIND ausgeführt. Diese Arbeitsschritte wurden in 2.3.1 erläutert. Sind alle Referenzpunkte gelernt, wird ROS eingesetzt, um eine Karte des Bereiches zu zeichnen. Das Scannen der Referenzpunkte sowie das Zeichnen der Karte wird mit dem Loomo ausgeführt. Zusätzlich zu den Scans mit dem Loomo müssen die Referenzpunkte ein zweites Mal mit dem Smartphone gescannt werden. Die zusätzlichen Scans mit dem Smartphone werden unter einer neuen (Geräte-) Familie ausgeführt (Kapitel 2.3) und dienen der Verbesserung der Lokalisierung des Smartphones. Eine Erläuterung ist dem Abschnitt Hindernisse (3.2) zu entnehmen.

Das Erstellen der Karte erfolgt mithilfe der SLAM-Implementierung von ROS. Zeitgleich zum Zeichnen der Karte wird der Standort des Loomos mittels FIND ermittelt und die Koordinaten des Loomos mittels Odometrie bestimmt. Sowohl der von FIND bestimmte Standort als auch die durch Odometrie berechneten Koordinaten werden an den Location-Service geschickt. Beide Daten werden im Location-Service in Relation gesetzt und in der Datenbank abgespeichert.

Wurde der Bereich nun sowohl vom SLAM als auch von FIND vollständig erfasst, kann nun der Loomo zu den Referenzpunkten gerufen werden. Dies funktioniert folgendermaßen: Zunächst wird ein Smartphone mit der für das Anwendungsszenario implementierten FIND-Applikation benötigt. Das Smartphone wird in dem gescannten Bereich positioniert. Sobald nun die Lokalisierung des Smartphones mittels App gestartet wird, wird der Referenzpunkt vorhergesagt, an dem sich das Smartphone befindet. Da vorher FIND-Standort und Odometrie Koordinaten zusammen persistiert wurden, können nun

die zum FIND-Standort passenden Koordinaten zurückgegeben und an ROS gesendet werden. ROS übernimmt dann die Navigation des Loomos.

3.2 Hindernisse

Während der Implementierung des Anwendungsfalles kam es zu verschiedenen Problemen. Diese Probleme und deren Lösung, falls eine gefunden wurde, werden im folgenden Abschnitt beschrieben.

Zur Lokalisierung mittels FIND wird ein FIND-Server sowie ein FIND-Client benötigt. Der FIND-Server ist eine Komponente von FIND und übernimmt sowohl die Evaluierung von Fingerprints als auch die darauf basierende Berechnung der Standortvorhersage. Der FIND-Client ist verantwortlich für das Sammeln und Zustellen von Fingerprints an den FIND-Server und muss gegebenenfalls selbst implementiert werden. Beides zusammen repräsentiert dann FIND.

Zur Lokalisierung von Smartphones stellt der Entwickler von FIND eine Android-Applikation zur Verfügung, die die benötigte Funktionalität bietet. Beim Ausführen der Applikation auf der aktuellen Android-Version 11 (Android 11) kommt es jedoch zu Abstürzen. Diese Abstürze machen die Verwendung der Applikation unmöglich. Aus diesem Grund wurde die Applikation neu implementiert, sodass diese auf der aktuellsten Android-Version (Android 11) lauffähig ist. Bei der Implementierung wurde die Applikation des Entwicklers [53] als Vorlage genommen. Die aktualisierte Applikation wurde in Kotlin geschrieben. Aufbau und Funktionalität wurden weitestgehend übernommen. Es sei noch angemerkt, dass es sich bei der implementierten Applikation um einen Prototyp handelt.

Bei der Implementierung der Applikation wurde eine weitere Problematik offengelegt. In Android Versionen ab Android 9 (Android Pie) wurden Beschränkungen bezüglich der Anzahl von AP-Scans eingeführt [17]. Dabei werden Foreground-Applikationen [5] und Background-Applikationen [11] unterschieden. Die Anzahl der erlaubten Scans wurde auf maximal vier Scans in einer zwei Minuten Periode für Foreground-Applikationen und einen Scan pro 30 Minuten für die Summe aller Background-Applikationen [17] beschränkt. Dies macht eine Echtzeit-Standortverfolgung nahezu unmöglich. Der aktuelle Standort beziehungsweise eine Standortänderung wäre dann erst nach zwei Minuten sichtbar. Mit Android 10 wurde jedoch die Option eingeführt, die Scan-Restriktionen

abzuschalten [17]. Dies ist jedoch keine Option für Endverbraucher, da dies Änderungen in den Developer Options [3] von Android erfordert und diese nur erfahrenen Nutzern vorbehalten sind.

Ein weiteres Hindernis beziehungsweise eine weitere Herausforderung war die technische Umsetzung der Relation zwischen den Odometrie- und den FIND-Daten. Wie genau dies realisiert wurde, wird im Folgenden erläutert.

Odometrie-Daten vom Loomo und Referenzpunkt-Bezeichner von FIND werden mit verschiedenen Frequenzen an den Location-Service übermittelt. Da die Frequenz der Odometrie-Daten deutlich höher ist als die der FIND-Daten, wurde eine Variable angelegt, die immer mit den aktuellsten Odometrie-Daten aktualisiert wird. Sobald der Location-Service eine Position von FIND empfängt, wird die aktuelle Koordinate aus der Variable gelesen und zusammen mit der Position aus FIND in einer Datenstruktur gebündelt. Die Datenstruktur speichert den FIND-Standort und die Odometrie-Daten in einer String-Variable.

Da in diesem Anwendungsfall die Referenzpunkte jeweils einen eigenen Raum repräsentierten, wurden mehrere Odometrie-Koordinaten mit dem gleichen Referenzpunkt-Bezeichner assoziiert, wodurch bei einer Abfrage nach einem Navigationsziel gleich mehrere Koordinaten zurückgegeben wurden. Um dem entgegenzuwirken, wird bei der Abfrage nach einem Ziel für den Loomo alle zu einem Referenzpunkt-Bezeichner gehörenden Odometrie-Koordinaten aus der Datenbank gelesen. Aus jeweils den x- und y-Werten jeder Koordinate wird dann der Median berechnet und die beiden ermittelten Werte werden als Koordinate an den Loomo übermittelt. Damit wird das Ziel des Loomos so mittig wie möglich gehalten.

Folgendes Hindernis deckt sich mit den für fingerprintbasierte Positionierungssysteme bekannten Problemen der Geräteheterogenität (Kapitel 2.1.2).

Bei den ersten Tests des Systems wurde festgestellt, dass die Positionsbestimmung von Loomo und Smartphone verschieden präzise war. Dabei wurden die Scans in der Offline-Phase mit dem Loomo ausgeführt und beide Geräte nutzten die auf dieser Datenbasis trainierten Klassifizierer für die Standortvorhersage. Der Unterschied in der Präzision war teilweise so groß, dass für das Smartphone Referenzpunkte vorhergesagt wurden, die maximal weit entfernt zu dem tatsächlichen Referenzpunkt waren, während die Vorhersage für den Loomo mit dem tatsächlichen Referenzpunkt übereinstimmte.

Diese Diskrepanz zwischen den beiden Vorhersagen ließ sich auf das Problem der Geräteheterogenität zurückführen. Verschiedene Geräte haben verschiedene WLAN-Antennen und Chips verbaut [31][52][46]. Diese Unterschiede wirken sich unter anderem darauf aus, wie viele Signale ein Gerät bei einem Scan nach verfügbaren APs empfängt und wie stark die empfangenen Signale sind. Dies kann darin resultieren, dass zwei Geräte am selben Referenzpunkt verschiedene Signalstärken messen oder die Signale von bestimmten APs gar nicht empfangen. Im Kontext von FIND haben diese Unterschiede Auswirkungen auf die Genauigkeit der Positionsvorhersagen.

Diese Heterogenität wurde beim Scannen mit dem Loomo und einem Smartphone beobachtet. Das Smartphone war in der Lage die Signale von mehr APs zu empfangen. Damit unterschieden sich die vom Smartphone empfangenen Fingerprints von den, die vom Loomo empfangenen wurden. Wodurch die Positionsvorhersagen für das Smartphone ungenau waren.

Hier kam dann die im Kapitel 2.3 beschriebene Familien-Funktion von FIND zum Einsatz. Durch die Angabe einer Familie beim Scannen der Referenzpunkte, war es möglich, die Scans nach Gerät zu gruppieren. So konnten sowohl für das Smartphone als auch für den Loomo unabhängige Scans durchgeführt werden. Es sei noch erwähnt, dass mehr als zwei Familien angelegt werden können und die Bezeichner für eine Familie frei wählbar sind. Außerdem muss für jede angelegte Familie der gesamte Bereich neu gescannt werden, was den Zeitaufwand der Offline-Phase nochmals erhöht.

3.3 Implementierung

3.3.1 Benutzeroberfläche: Loomo

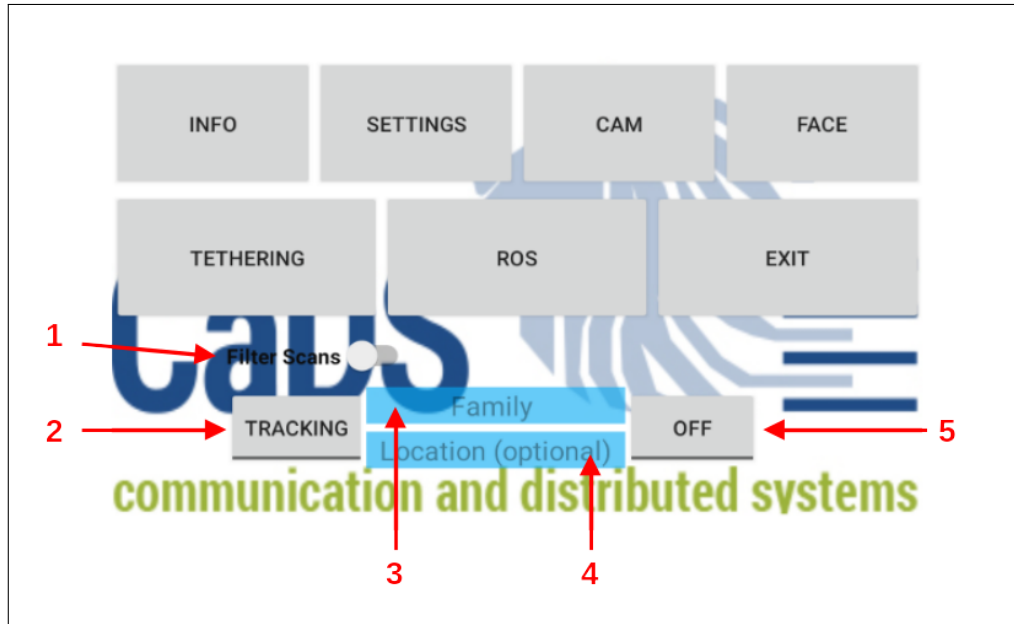


Abbildung 3.2: Loomo Benutzeroberfläche

Im Rahmen dieser Arbeit wurde die Loomo-Benutzeroberfläche um zwei Eingabefelder, einen Schalter und zwei Buttons erweitert 3.2. Alle Komponenten befinden sich im unteren Teil der Benutzeroberfläche. Die Funktionen der hinzugefügten grafischen Komponenten werden nun erläutert.

Der Schalter (engl. Switch) mit der Beschriftung “Filter Scans” (siehe Abbildung 3.2 Nummer 1) dient der Aktivierung eines Filters für die gesammelten Fingerprints. Wird der Filter aktiviert, werden lediglich die zehn besten Fingerprints an den FIND-Server gesendet. Als die zehn besten Fingerprints gelten die Fingerprints, die den höchsten RSSI-Wert haben. Werden weniger als zehn Fingerprints empfangen, werden alle an den FIND-Server geschickt.

Das obere Eingabefeld mit dem Bezeichner “Family” (Nummer 3) dient der Eingabe einer FIND-Family. So können beispielsweise alle Loomos als eine Familie gruppiert werden.

Das untere Eingabefeld mit der Nummer 4 und dem Bezeichner “Location” wird nur in der Offline-Phase (siehe Kapitel 2.3.1) eingesetzt. In das Eingabefeld wird der Bezeich-

ner des Referenzpunktes eingetragen, an dem der Loomo sich aktuell befindet. Dieser Bezeichner wird zusammen mit den dort gesammelten Fingerprints an den FIND-Server gesendet. Dieser verwendet die Bezeichner beim Trainieren der Klassifizierer als Label für die Trainingsdaten [37].

Der sich links befindende Button mit der Beschriftung “TRACKING” (Nummer 2) ist ein sogenannter “Togglebutton” [15] und erlaubt den Wechsel zwischen zwei Zuständen. In diesem Fall erlaubt der Button den Wechsel zwischen den Scan-Typen “TRACKING” und “LEARNING”, die angeben, ob die aktuellen Scans im Rahmen der Offline- oder Online-Phase durchgeführt werden.

Ist der Scan-Typ “LEARNING” ausgewählt, muss das “Location”-Eingabefeld ausgefüllt sein, da die gesammelten Fingerprints zum Training der Klassifizierer verwendet werden und deswegen ein Label zu den gegebenen Daten benötigen.

Bei dem rechts liegenden Button mit der Nummer 5 handelt es sich ebenfalls um einen “Togglebutton”. Mit diesem wird zwischen den Zuständen “Off” und “SCANNING” gewechselt, was das Starten und Beenden eines Scans erlaubt.

3.3.2 Benutzeroberfläche: FIND-Applikation

In Abbildung 3.3 ist die Benutzeroberfläche der FIND-Applikation zu sehen. Diese wird im Folgenden beschrieben.

Die FIND-Applikation, wie sie hier zu sehen ist, ist ein Nachbau der originalen FIND-Applikation. Diese ist online [53] sowie im Google Play Store zu finden. Die Benutzeroberfläche der FIND-Applikation ist simpel gehalten. Sie besteht aus drei beziehungsweise vier Eingabefeldern, vier Schaltern (Switches) und zwei Buttons.

Das oberste Eingabefeld mit dem Bezeichner “Family” dient der Eingabe der Gerätefamilie, zu der das Gerät, auf dem die Applikation ausgeführt wird, gehört. Darauf folgt ein Eingabefeld für den Gerätenamen. Hier wird ein Bezeichner für das zu lokalisierende Gerät eingetragen. Im nächsten Eingabefeld werden die IP-Adresse und der Port des FIND-Servers eingetragen. Das letzte Eingabefeld wird erst sichtbar, wenn der Schalter mit der Beschriftung “Learning” aktiviert wird. Dieses Eingabefeld erwartet einen Bezeichner für einen Referenzpunkt. Da der Referenzpunkt-Bezeichner nur nötig ist, wenn Trainingsdaten für den FIND-Server gesammelt werden, ist das Eingabefeld standardmäßig nicht aktiviert.

The image shows a mobile application interface for the FIND app. It features three text input fields at the top, labeled 'Family', 'Device Name', and 'Server URL'. Below these fields are four toggle switches arranged in a 2x2 grid. The top row contains 'Submit GPS' and 'Submit Bluetooth', both of which are currently turned off. The bottom row contains 'Learning' and 'Filter Scans', both of which are currently turned on. At the bottom of the screen, there are two prominent buttons: a blue button labeled 'START SCANNING' and a red button labeled 'STOP SCANNING'.

Abbildung 3.3: FIND App Benutzeroberfläche

Unter den Eingabefeldern befinden sich Schalter. Über diese Schalter lassen sich verschiedene Einstellungen bezüglich der Scans vornehmen. Der “Learning”-Schalter wurde bereits diskutiert. Über die Schalter “Submit Bluetooth” und “Submit GPS” lässt sich einstellen, ob beim Scannen Bluetooth-APs berücksichtigt beziehungsweise ob GPS-Daten mit an den Server geschickt werden sollen (falls diese vorhanden sind). Beide Funktionen wurden im Rahmen der Bachelorarbeit nicht genutzt. Der Schalter mit der Beschriftung “Filter Scans” aktiviert beziehungsweise deaktiviert den Filter, der nach den zehn Fingerprints mit den höchsten RSSI-Werten filtert.

Ganz unten im Bildschirm befinden sich zwei Buttons. Mit diesen Buttons wird ein Scan gestartet beziehungsweise beendet.

3.3.3 Technologien

In diesem Abschnitt wird die technische Umsetzung der Relation zwischen Odometrie- und FIND-Daten beschrieben. Außerdem werden kurz die verwendeten Technologien vorgestellt.

Apache Kafka (Kafka) ist eine verteilte, Open-Source Event-Streaming-Plattform [2]. Kafka erlaubt das Austauschen von Nachrichten zwischen Services mittels dem Publish-Subscribe-Messaging-Pattern [48]. Im Rahmen der Bachelorarbeit wird Kafka zur Kommunikation zwischen ROS und dem Location-Service eingesetzt. Dabei findet die Kommunikation über zwei Kafka-Topics [2] statt. Das erste Topic wird für die Kommunikation vom Location-Service zu ROS verwendet. Es hat den Bezeichner “ros-output”. Das zweite Topic ermöglicht die Kommunikation in die andere Richtung und hat den Bezeichner “ros-input”. Übermittelt werden die mittels Loomo-Odometrie ermittelten Koordinaten. Diese werden zum Location-Service geschickt und dort persistiert, um diese dann später als Navigationsziel zurück an den Loomo zu schicken. Außerdem wird Kafka zur Kommunikation mit der Lambda-Architektur eingesetzt. Dabei dient die Kommunikation zur Lambda-Architektur lediglich späteren Analysen. Diese Analysen werden in dieser Thesis nicht behandelt.

Als Event-Streaming-Plattform eignet sich Kafka für die Erfassung von Daten in Echtzeit. So eine Echtzeit-Datenquelle sind die Odometrie-Daten beziehungsweise die Sensoren des Loomos. Aus diesem Grund und der Tatsache, dass Kafka als Anbindung für die Lambda-Architektur eingesetzt wird, wurde sich in dieser Arbeit für Kafka als Kommunikationstechnologie entschieden.

Für die Implementierung des Location-Service wurde die Programmiersprache Java eingesetzt. Des Weiteren wurde das Spring-Boot Framework [14] verwendet, um den Service zu implementieren.

Wie in Abschnitt 3.1 bereits erwähnt, wurden zwei ROS-Nodes implementiert. Dafür wurde die Programmiersprache Python [16] verwendet, da ROS ein Application Programming Interface (API) für Python anbietet.

4 Experiment

Die im Folgenden beschriebenen Experimente dienen der Evaluierung der Genauigkeit von FIND im Kontext autonomer mobiler Systeme. Für die Experimente wird ein Testbed aufgebaut, indem ein Raum in mehrere Teilbereiche aufgeteilt wird. Dadurch entsteht ein virtuelles Raster (siehe Abbildung 4.1), wobei die Anzahl der Teilbereiche die Granularität des Rasters bestimmt. Werden alle Teilbereiche erkannt, wird eine Genauigkeit entsprechend der Granularität des Rasters angenommen.

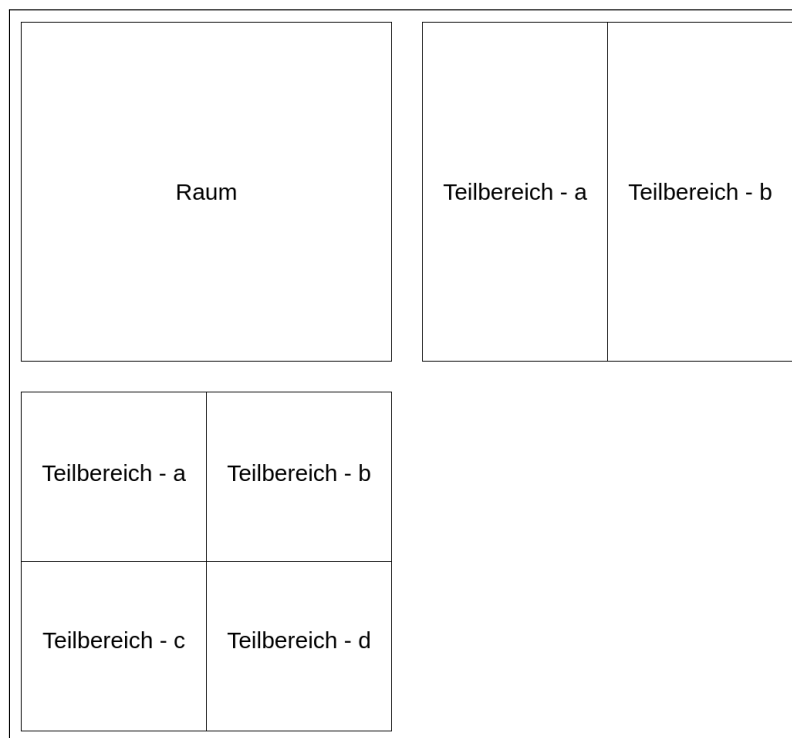


Abbildung 4.1: Skizze des virtuellen Rasters

Die Experimente werden in zwei Varianten ausgeführt. Einmal unter der Berücksichtigung aller, vom Gerät sichtbaren, APs und ein zweites Mal unter der Berücksichtigung der zehn

besten APs. Die Güte eines APs wird anhand der Signalstärke bestimmt. Außerdem wird angenommen, dass mindestens zehn APs sichtbar sind. Sollten weniger APs sichtbar sein, werden alle sichtbaren berücksichtigt.

Grund für ein Experiment in der zweiten, eben beschriebenen, Variante ist die These, dass durch das Filtern störende beziehungsweise stark rauschende Signale (Access Points) ignoriert werden und dies in einer besseren Standortvorhersage resultiert.

Für das Experiment wurden folgende Komponenten verwendet:

- Ein Loomo, mit dem die einzelnen Teilbereiche des Raumes gescannt werden. Nach dem Scannen wird versucht, den Loomo in jedem Teilbereich zu lokalisieren.
- FIND
- Ein Raum. Dieser muss mithilfe von Markierungen am Boden in Teilbereiche geteilt werden.

Zusätzlich zu den eben genannten Komponenten wurde ein Python-Skript für das Experiment geschrieben. Das Skript empfängt Standortvorhersagen von FIND und vergleicht diese mit dem erwarteten Referenzpunkt-Bezeichner. Die Ergebnisse des Vergleichs sowie die Wahrscheinlichkeit, mit der die Standortvorhersage getroffen wurde, werden vom Skript auf die Konsole geschrieben.

4.1 Aufbau

In diesem Abschnitt werden die Schritte zur Vorbereitung der Experimente beschrieben. Der Experimentaufbau setzt sich aus drei Schritten zusammen. Als Erstes muss das Gerät, welches zur Lokalisierung eingesetzt wird, mit der Software zum Scannen nach APs ausgestattet werden. Danach muss ein FIND-Server aufgesetzt werden. Schließlich muss ein Raum für das Experiment ausgesucht und vorbereitet werden. Die Reihenfolge der eben beschriebenen Schritte ist dabei nicht relevant.

Für die Experimente wurde der Loomo eingesetzt. Um den Loomo einsatzfähig zu machen wurde eine Version der “LoomoCADSService“-Applikation installiert, die eine Implementierung des FIND-Clients enthält.

Der FIND-Server kann mithilfe der Dokumentation von FIND aufgesetzt werden [12]. Hier ist lediglich zu beachten, dass der Loomo den FIND-Server erreichen kann. Beispielsweise dadurch, dass sich beide Systeme in einem Netzwerk befinden.

Nachdem ein Raum ausgewählt wurde, wird dieser ausgemessen. Entsprechend der benötigten Anzahl an Teilbereichen werden Markierungen auf dem Boden angebracht, die die Teilbereiche voneinander trennen. Die Markierungen können im Laufe des Experimentes erweitert werden, um neue Teilbereiche hinzuzufügen.

4.2 Durchführung

Im Folgenden werden die einzelnen Arbeitsschritte zum Durchführen des Experimentes beschrieben.

Zunächst wird der Loomo in einem der Teilbereiche positioniert. Um einen Scan zu starten, müssen die beiden Felder “Family” und “Location” ausgefüllt sowie der Scan-Modus auf “SCANNING” geschaltet sein. Alles lässt sich auf der Loom-Benutzeroberfläche (siehe 3.3.1) einstellen.

In das “Family”-Eingabefeld wird der Bezeichner für den aktuellen Experimentdurchlauf eingetragen. Die “Family”-Funktion wird hier zur Kapselung einzelner Experimentdurchläufe eingesetzt.

In das “Location”-Eingabefeld wird der Bezeichner für den Teilbereich, in welchem der Loomo sich aktuell befindet, eingetragen.

Schließlich wird der sich links von den Eingabefeldern befindende Button betätigt, sodass der Text “LEARNING” zu sehen ist.

Mit dem sich rechts von den Eingabefeldern befindenden Button kann nun ein Scan gestartet werden. Beim Betätigen wechselt der Text auf dem Button von “Off” zu “SCANNING” und indiziert damit den Start des Scans. Zeitgleich zum Scan Start wird ein Timer gestartet. Dieser ist auf zehn Minuten eingestellt. Nach Ablauf des Timers wird der Scan durch erneuten Knopfdruck des Buttons beendet.

Dieser Prozess wird für jeden Teilbereich wiederholt. Sind alle Teilbereiche gescannt, beginnt die eigentliche Evaluierung von FIND. Hierfür wird der Loomo erneut in jedem der Teilbereiche positioniert und es werden erneut Scans durchgeführt. Diesmal wird in

der Loomo-Benutzeroberfläche der Scan-Typ auf “TRACKING” gestellt. Dies bedeutet, die gesammelten Fingerprints werden nicht zum Trainieren der Klassifizierer eingesetzt, weshalb auch kein Bezeichner in des “Location”-Eingabefeld eingegeben werden muss. Die Fingerprints werden stattdessen mit den bereits trainierten Klassifizierern evaluiert und es wird eine Standortvorhersage getroffen.

Um ein aussagekräftiger Ergebnis zu erhalten, werden je Teilbereich zehn Fingerprints evaluiert. Jede Evaluation wird mit dem erwarteten Standort verglichen und es wird notiert, ob beide Werte übereinstimmen. Dabei gilt ein Teilbereich als erkannt, sobald sechs von zehn Fingerprints den erwarteten Standort vorausgesagt haben. Zusätzlich werden die Vorhersage-Wahrscheinlichkeiten aus allen korrekt vorhergesagten Standorten gemittelt und notiert. Wird ein Fingerprint falsch klassifiziert, wird diesem eine Wahrscheinlichkeit von null zugewiesen. Die Ergebnisse des Experimentes werden im Kapitel 5 vorgestellt und diskutiert.

4.3 Erwartung

Aufgrund von Erfahrungen, die bei der Implementierung des Anwendungsfalls gemacht wurden und der vom Entwickler zur Verfügung gestellten Dokumentation von FIND, wird eine erfolgreiche Unterscheidung der Teilbereiche lediglich bei einer Zweiteilung des Raumes erwartet.

In den FAQs zu FIND [6] wird erläutert, dass eine Genauigkeit von ca. einem Quadratmeter möglich ist. Dies würde zu der Annahme führen, dass mehr als eine Zweiteilung des Raumes durch FIND erkannt wird. Jedoch wird weiterhin angemerkt, dass die Genauigkeit abhängig ist von Faktoren wie der Anzahl der sichtbaren APs am Referenzpunkt oder den Umgebungsbedingungen am Referenzpunkt.

Dies bedeutet unter anderem, dass sich durch eine Wand getrennte Referenzpunkte positiv auf die Genauigkeit der Positionsbestimmung von FIND auswirken. Dies ist darauf zurückzuführen, dass sich die gesammelten Fingerprints der beiden, durch die Wand getrennten, Referenzpunkte stärker voneinander unterscheiden als die Fingerprints von zwei Referenzpunkten, die in gleicher Entfernung zueinander liegen, jedoch nicht durch eine Wand oder ähnliches getrennt werden. Da Signaldämpfungen und Multipath-Effekte durch Wände oder andere Hindernisse verstärkt beziehungsweise erzeugt werden und das Fehlen eines solchen Hindernisses die Signale ungehindert durch mehrere Referenzpunkte

passieren lässt, können Fingerprints an solchen Referenzpunkten ähnlich oder sogar gleich aussehen. Ähnliche oder gleiche Fingerprints wiederum resultieren in Schwierigkeiten bei der Unterscheidung der Referenzpunkte und damit ungenaueren Standortvorhersagen, da die klassifizierenden Algorithmen weniger unterscheidende Merkmale in den Fingerprints eines Referenzpunktes erkennen.

Bezogen auf die hier aufgebaute Testumgebung sollte ein Mangel an APs kein Problem darstellen. Bei Messungen wurden bereits über zehn sichtbare APs beobachtet. Außerdem wurden weitere Router aufgebaut, die ebenfalls als APs genutzt werden. Das Hauptkriterium für die hier beschriebene Erwartung ist die Konstellation der Referenzpunkte, wie sie für das Experiment benötigt ist. Da es sich bei den Referenzpunkten um Teilbereiche ein und desselben Raumes handelt, liegen alle Referenzpunkte unmittelbar nebeneinander. Zwischen den Referenzpunkten befinden sich keine Hindernisse, wie beispielsweise Wände, was zu den oben beschriebenen Problemen führt.

Bei dem Vergleich der Genauigkeit der Lokalisierung unter Berücksichtigung aller APs und unter der Berücksichtigung der besten zehn APs, wird erwartet, dass erstere Methode bessere Ergebnisse liefert. Grund für die Erwartung ist: Da beim Scannen mit Filter immer nur zehn APs behalten und an den FIND-Server gesendet werden, ist bei den gefilterten Scans die Anzahl der gesammelten Daten geringer als bei den ungefilterten Scans. Jedoch muss hier beachtet werden, dass nicht die Anzahl der gesammelten Fingerprints geringer ist, sondern die Anzahl der BSSID-RSSI-Paare je Fingerprint. Dies resultiert in einer höheren Wahrscheinlichkeit für ähnliche Fingerprints. Außerdem werden beim Filtern APs ignoriert, die gegebenenfalls nur in einem der Teilbereiche sichtbar sind und damit die Unterscheidung der einzelnen Referenzpunkte unterstützen würden. Umgekehrt werden beim Filtern nach den APs mit den höchsten RSSI-Werten jeweils die gleichen APs für jeden Referenzpunkt ermittelt, da sich die Referenzpunkte unmittelbar nebeneinander befinden. Dies macht das Differenzieren der Referenzpunkte schwieriger.

5 Evaluation

In diesem Kapitel werden die Ergebnisse aus den Experimenten vorgestellt und anschließend diskutiert.

5.1 Ergebnisse

Im folgenden Abschnitt werden die Ergebnisse des Experimentes aus Kapitel 4 vorgestellt. Die Ergebnisse sind in Form von zwei Tabellen dargestellt. Dabei präsentiert die erste Tabelle (siehe Tabelle 5.1) die Ergebnisse ohne den Einsatz des Filters beim Scannen und die zweite Tabelle (siehe Tabelle 5.2) die Ergebnisse mit Einsatz des Filters. Nach der Erläuterung der Tabellen und der Präsentation der Werte werden diese diskutiert.

Beide Tabellen sind äquivalent aufgebaut. Eine Tabelle besteht aus fünf Spalten: “Teilbereiche”, “Bezeichner”, “Daten”, “Scans erkannt”, “Durchschn. Wahrscheinlichkeit” (Durchschnittliche Wahrscheinlichkeit). Die Anzahl der Zeilen ist abhängig von der Anzahl der Experimentdurchläufe. Die Spalten werden im Folgenden von links nach rechts vorgestellt und es wird ihre Semantik besprochen.

Die erste Spalte hat den Bezeichner “Teilbereiche”. Ein Eintrag in dieser Spalte repräsentiert einen Experimentdurchlauf. In der Spalte selbst ist die Anzahl der Teilbereiche zu sehen.

In der nächsten Spalte, “Bezeichner” sind die einzelnen Bezeichner der Teilbereiche zu sehen. Die Anzahl der Bezeichner entspricht der Anzahl der Teilbereiche, die in der vorherigen Spalte angegeben sind. Hier findet eine Teilung der Zeile statt, da die folgenden Spalten Informationen für den jeweiligen Teilbereich enthalten. Trotzdem sind die Zeilen einem Experimentdurchlauf zuzuordnen.

In der dritten Spalte “Daten” wird die Anzahl der Fingerprints angegeben, die während des Scannens des Referenzpunktes aggregiert wurden.

Die vierte Spalte mit dem Bezeichner “Scans erkannt” enthält die Anzahl der Fingerprints, die bei der Klassifizierung richtig erkannt wurden. Der Spalte ist außerdem zu entnehmen, dass jeweils zehn Fingerprints klassifiziert wurden. Diese Spalte dient der Evaluation von FIND. An den Einträgen in dieser Spalte wird entschieden, ob FIND einen Teilbereich richtig erkannt hat oder nicht. Wie in Kapitel 4 bereits beschrieben, gilt ein Teilbereich als erkannt, wenn mindestens sechs von zehn Fingerprints richtig klassifiziert wurden.

In der letzten Spalte sind die durchschnittlichen Wahrscheinlichkeiten notiert. Die durchschnittliche Wahrscheinlichkeit gibt die Sicherheit der Standortvorhersage in Form einer Prozentzahl wieder. Hier sei noch angemerkt, dass eine falsche Standortvorhersage mit einer Wahrscheinlichkeit von null gewertet wurde.

Nun folgt die Vorstellung der Resultate, die sich mit Einsatz des Filters und ohne Einsatz des Filters ergeben haben. Zur Evaluation der Ergebnisse wurden folgende Tabellen erstellt (siehe Tabelle 5.1 und 5.2).

Teilbereiche	Bezeichner	Daten	Scans erkannt	Durchschn. Wahrscheinlichkeit
2	a	120	10/10	0,845
	b	121	8/10	0,547
4	a	120	9/10	0,65
	b	120	0/10	(0,718)
	c	120	10/10	0,59
	d	120	10/10	0,808

Tabelle 5.1: Ergebnisse ohne Filter

Teilbereiche	Bezeichner	Daten	Scans erkannt	Durchschn. Wahrscheinlichkeit
2	a	120	10/10	0,83
	b	121	9/10	0,702
4	a	121	1/10	0,07
	b	121	1/10	0,034
	c	120	6/10	0,286
	d	120	10/10	0,78

Tabelle 5.2: Ergebnisse mit Filter

In der Spalte “Teilbereiche” ist zu sehen, dass für beide Versuche sowohl eine Zweiteilung als auch eine Vierteilung des Bereichs vorgenommen wurde. Dies bedeutet, dass bei den Experimenten jeweils zwei Durchläufe ausgeführt wurden.

In der nächsten Spalte sind die Bezeichner der einzelnen Teilbereiche zu sehen. Für die Bezeichner der einzelnen Teilbereiche wurden die Buchstaben aus dem lateinischen Alphabet gewählt. Beide Spalten sind sowohl für Tabelle 5.1 als auch für Tabelle 5.2 identisch.

In der Spalte "Daten" ist die Anzahl der Fingerprints wiederzufinden. Diese ist in jedem Durchlauf ungefähr gleich, da die Scan-Frequenz sowie die Dauer der Scans identisch war. Die Anzahl variiert zwischen 120 und 121 Fingerprints. Diese Spalte hat Relevanz für die Reproduzierbarkeit der Experimente und kann genutzt werden, um Korrelation zwischen der Anzahl der Fingerprints und der Genauigkeit der Scans zu untersuchen. Auch diese Spalten ähneln sich in beiden Tabellen.

Erst in der Spalte "Scans erkannt" unterscheiden sich die Werte aus den Tabellen wesentlich. In Tabelle 5.1 ist zu sehen, dass bei einer Zweiteilung des Raumes beide Teilbereiche erkannt wurden. Für Teilbereich "a" wurden zehn von zehn Fingerprints richtig erkannt. Für Teilbereich "b" wurden hingegen acht von zehn Fingerprints korrekt vorhergesagt, jedoch gilt jeder Teilbereich, bei dem die Anzahl der korrekten Vorhersagen größer fünf ist, als erkannt.

Bei den folgenden Zeilen zum zweiten Experimentdurchlauf ist zu sehen, dass einer der Teilbereiche nicht erkannt wurde. Für den Teilbereich mit dem Bezeichner "b" wurde der Standort kein einziges Mal korrekt vorhergesagt. Die anderen Teilbereiche wurden korrekt vorhergesagt, jedoch ist zu erkennen, dass die Sicherheit der Vorhersagen für Teilbereich "a" und "c" (Durchschnittliche Wahrscheinlichkeit) nahe 50 % liegt. Dennoch gilt die Vierteilung des Bereiches somit als nicht erkannt. Das Experiment wurde an dieser Stelle beendet.

Tabelle 5.2 zeigt, dass auch mit Einsatz des Filters eine Teilung des Bereichs in zwei Hälften erfolgreich erkannt wurde. Beide Teilbereiche haben die korrekten Standorte mit einer hohen durchschnittlichen Wahrscheinlichkeit vorhergesagt. Teilbereich "a" mit 83% und zehn von zehn korrekt vorhergesagten Fingerprints und Teilbereich "b" mit 70,2% und neun von zehn korrekten Vorhersagen.

Beim nächsten Experimentdurchlauf verschlechterten sich die Ergebnisse stark. Hier wurden nur für zwei Teilbereiche die Fingerprints mehr als fünfmal korrekt klassifiziert. Dies ist zum einen der Teilbereich "d" mit zehn von zehn korrekten Klassifizierungen und einer durchschnittlichen Wahrscheinlichkeit von 78% und zum anderen der Teilbereich "c" mit sechs von zehn korrekt klassifizierten Fingerprints und einer durchschnittlichen

Wahrscheinlichkeit von 28,6%. Die restlichen zwei Teilbereiche weisen nur einen korrekt klassifizierten Fingerprint auf.

5.2 Diskussion

FIND stellt Nutzern Diagramme zu jedem erstellten Referenzpunkt zur Verfügung. Mittels dieser Diagramme können die beim Scannen erhobenen Daten angesehen werden, um beispielsweise feststellen zu können, wie gut die Positionsbestimmung ist.

In einem Diagramm werden zehn Graphen gezeigt, die jeweils die zehn relevantesten APs repräsentieren. Die relevantesten APs sind die, die die höchste Varianz zwischen den Referenzpunkten haben und somit den größten Beitrag zur Unterscheidbarkeit der Referenzpunkte leisten. Auf der x-Achse jedes Diagramms sind RSSI-Werte dargestellt. Auf der y-Achse ist die Wahrscheinlichkeit dargestellt, mit der ein RSSI-Wert an einem bestimmten Referenzpunkt vorzufinden ist. Die BSSIDs der abgebildeten APs werden rechts im Diagramm, in Form von einer Legende, aufgelistet.

Hohe y-Werte im Diagramm bedeuten, dass entsprechende RSSI-Werte auf der x-Achse mit einer hohen Wahrscheinlichkeit an dem entsprechenden Referenzpunkt vorgefunden werden. Andersherum bedeutet dies, wird bei einem Scan ein RSSI-Wert gemessen, der einem hohen y-Wert zugeordnet ist, wird die Standortvorhersage stärker zu diesem Referenzpunkt gelenkt, als dies beim Messen eines RSSI-Wertes mit einem niedrigen y-Wert der Fall ist.

Ausschlaggebend für die Unterscheidung der einzelnen Referenzpunkte ist die Varianz in den dort empfangenen Fingerprints. Sprich, wenn ein AP an einem Referenzpunkt RSSI-Werte im Bereich von -80 dBm bis -90 dBm hat und an einem anderen Referenzpunkt Werte im Bereich von -30 dBm bis -40 dBm, existiert eine starke Variation in den RSSI-Werten, die förderlich für die Differenzierung von Fingerprints ist. Ebenso förderlich ist die Sichtbarkeit eines APs an nur einem Referenzpunkt. Diese Art von Variation lässt sich gut an dem Gesamtbild der Graphen aus einem Diagramm festmachen. Je stärker sich die Gesamtbilder der verschiedenen Referenzpunkte unterscheiden, desto sicherer sind die Standortvorhersagen.

Beim ersten Experimentdurchlauf wird der Test-Bereich in zwei Teilbereiche geteilt. Dies resultiert in zwei Referenzpunkten, die jeweils repräsentativ für einen der Teilbereiche ste-

hen. Dementsprechend werden zwei Diagramme für jeweils einen Referenzpunkt erzeugt.

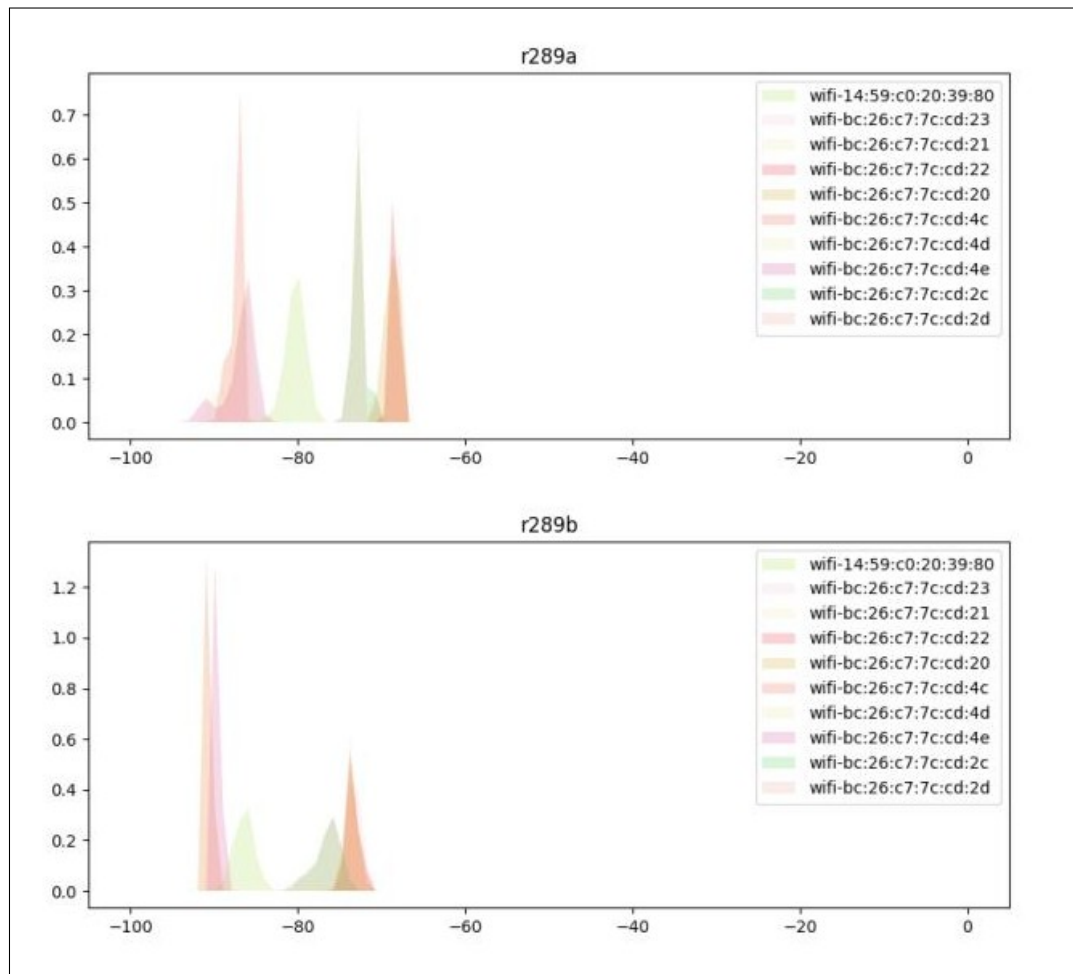


Abbildung 5.1: Erster Experimentdurchlauf

Wie in Abbildung 5.1 zu sehen ist, haben beide Diagramme ein ähnliches Aussehen. Dies deutet auf eine geringe Anzahl an unterscheidbaren Merkmalen zwischen den Referenzpunkten und damit auf eine unpräzise Standortvorhersage hin. Trotz der Ähnlichkeit fällt eine leichte Verschiebung entlang der x-Achse von den Graphen aus "r289a" auf. Diese Verschiebung wirkt sich positiv auf die Unterscheidung der beiden Referenzpunkte aus. Durch diese Verschiebung kommt es im Bereich um -80 dBm zu einer Divergenz. Dies ist ebenfalls positiv für die Unterscheidbarkeit der Referenzpunkte. Trotz der Unterschiede macht sich die Ähnlichkeit der beiden Diagramme in der Anzahl der korrekt klassifizierten

Fingerprints und damit der durchschnittlichen Wahrscheinlichkeit der Standortvorhersagen in Tabelle 5.1 sichtbar. Hier wird der Teilbereich “b” mit einer Sicherheit von 0,547% und acht von zehn korrekt klassifizierten Fingerprints vorhergesagt, was sehr nah am Zufall ist.

Es sei noch angemerkt, dass im Diagramm “r289b” (siehe Abbildung 5.1) zwei Graphen die logische Grenze von 1 für Prozent Angaben überschreiten. Dies ist auf einen Fehler in der Implementierung von FIND zurückzuführen und wird in dieser Thesis nicht weiter behandelt.

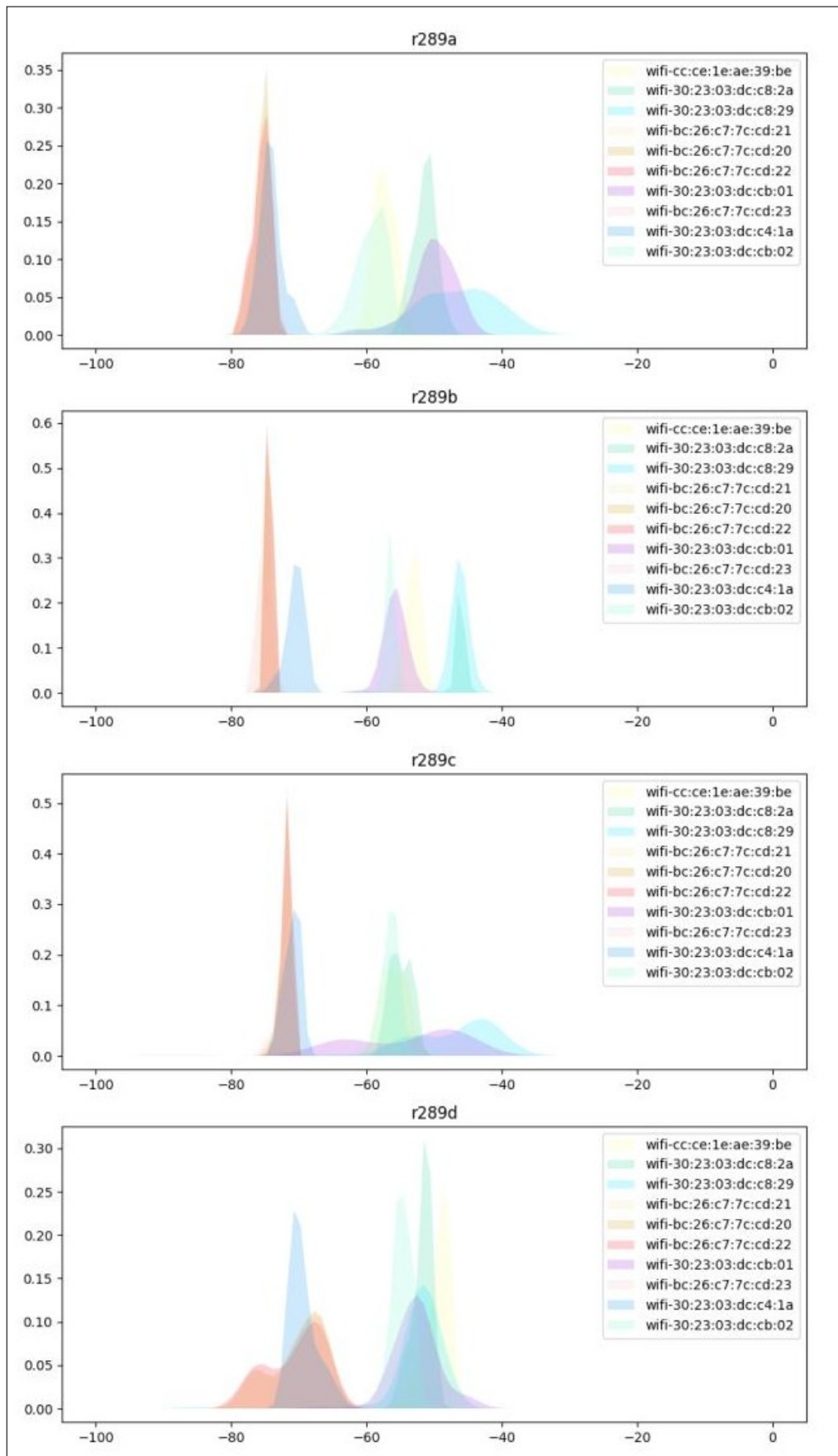


Abbildung 5.2: Zweiter Experimentdurchlauf

In Abbildung 5.2 werden die Diagramme zum zweiten Experimentdurchlauf abgebildet. Beim Betrachten der Diagramme fällt auf, dass die Graphen in allen vier Diagrammen einen RSSI-Wertebereich von circa -80 dBm bis circa -40 dBm haben. Werden die einzelnen Diagramme verglichen, sticht das Diagramm zum Referenzpunkt “r289b” heraus, da die Graphen in diesem Diagramm stärker gestaucht sind als in den anderen Diagrammen. Damit decken die Graphen einen geringeren RSSI-Wertebereich ab. Dies führt zu einer schlechteren Standorterkennung an diesem Referenzpunkt, da es weniger wahrscheinlich ist, bei einem Scan einen RSSI-Wert im Wertebereich dieser Graphen zu messen. Im Kontrast dazu haben die Graphen in den anderen Diagrammen eher einen gestreckten Verlauf.

Beides, der gestauchte Verlauf der Graphen aus “r289b” und die Tatsache, dass sich die Graphen aller Diagramme einen RSSI-Wertebereich teilen, deutet auf eine schlechte Performanz bei der Erkennung der Referenzpunkte und besonders bei der Erkennung vom Referenzpunkt “r289b” hin. Dies wird durch die Ergebnisse aus Tabelle 5.1 bestätigt. In Tabelle 5.1 ist abzulesen, dass der Referenzpunkt “r289b” keinmal korrekt vorhergesagt wurde und die Sicherheit, mit der die anderen Referenzpunkte in diesem Durchlauf vorhergesagt wurden, ist ebenfalls gering.

An dieser Stelle wurde das Experiment mit den ungefilterten Scans beendet, da nicht alle Teilbereiche erkannt wurden. Es wurde mit den gefilterten Scans fortgefahren.

In Abbildung 5.3 sind die Diagramme zum ersten Experimentdurchlauf mit Einsatz des Filters zu sehen. Werden diese mit den Diagrammen in Abbildung 5.1 verglichen, ist deutlich zu erkennen, dass die APs aus Abbildung 5.3 durchschnittlich höhere RSSI-Werte haben. Ebenso sind die abgebildeten BSSIDs in beiden Diagrammen verschieden, da für die Diagramme in Abbildung 5.1 die APs ausgesucht wurden, die am meisten Wert für die Standortvorhersage haben und nicht die APs, die die höchsten RSSI-Werte besitzen. Der Wert eines APs wird anhand der Varianz seiner Signalstärke an den verschiedenen Referenzpunkten festgemacht. Je mehr Varianz zwischen den Referenzpunkten besteht, desto mehr Wert hat der AP für die Standortbestimmung. Während im ersten Experimentdurchlauf mit Filter ebenso wie im ersten Experimentdurchlauf ohne Filter die Referenzpunkte ungefähr gleich oft erkannt wurden, geht die Anzahl an korrekten Standortvorhersagen im zweiten Experimentdurchlauf mit Filter deutlich runter. Dies zeigt, dass die APs mit dem stärksten Signal, nicht zwingend die APs mit dem größten Wert für eine korrekte Standortvorhersage sind. Im Gegenteil kann aus der Verschlechterung im zweiten Experimentdurchlauf mit Filter gefolgert werden, dass die APs mit den höch-

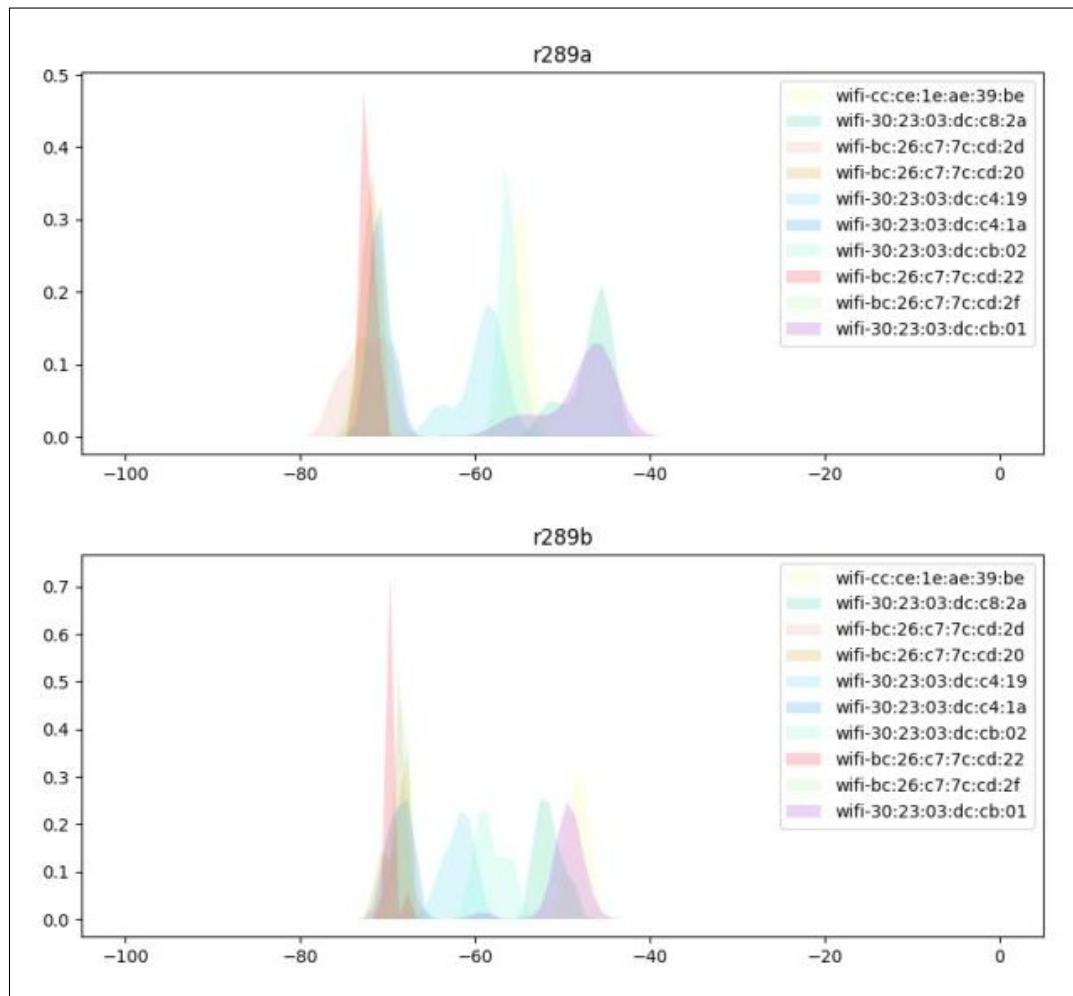


Abbildung 5.3: Erster Experimentdurchlauf mit Einsatz des Filters

ten RSSI-Werten in allen Teilbereichen sichtbar sind und in allen Teilbereichen einen hohen RSSI-Wert vorweisen. Dies gilt besonders dann, wenn die Teilbereiche sich unmittelbar nebeneinander befinden. Dadurch haben die APs wenig Varianz zwischen den Referenzpunkten und damit auch wenig Relevanz für die Standortvorhersagen. Zudem kann anhand des Vergleiches der beiden ersten Experimentdurchläufe festgestellt werden, dass die Signalstärke für die Positionsbestimmung mittels Fingerprinting keine Relevanz hat.

Werden die Ergebnisse aus den gefilterten Scans mit den Ergebnissen aus den ungefilterten Scans verglichen, ist zu erkennen, dass der Einsatz der Filter zu schlechteren Ergebnissen geführt hat. Zwar wurde in beiden Fällen nur eine Zweiteilung des Raumes

korrekt erkannt, dennoch wurden bei den Experimenten ohne Filter und einer Vierteilung des Raumes häufiger korrekte Standortvorhersagen getroffen als beim Einsatz des Filters und einer Vierteilung des Raumes. Dies entspricht der in Kapitel 4.3 verfassten Erwartung.

Der Raum, in dem die Experimente durchgeführt worden sind, ist in der Länge sowie in der Breite sechs Meter lang. Bei der Zweiteilung wurde der Raum in zwei Teilbereiche mit den Maßen drei Meter mal sechs Meter aufgeteilt. In der Tabelle 5.1 ist zu erkennen, dass eine Zweiteilung erfolgreich erkannt wurde. Dies resultiert in einer Granularität von circa 18 Quadratmetern. Eine weitere Teilung in vier Teilbereiche wurde vorgenommen, jedoch ist an den Ergebnissen zu erkennen, dass die Teilbereiche von FIND nicht erkannt wurden.

Hier sei nochmal angemerkt, dass die ermittelte Granularität mittels Teilbereichen festgestellt wurde, die nicht sichtbar voneinander getrennt waren. Mit anderen Worten: Zwischen den Teilbereichen befanden sich keine Hindernisse wie Wände oder ähnliches. Diese Art von Hindernissen sind vorteilhaft für die Unterscheidung und damit auch die Erkennung von Referenzpunkten, da dadurch eine größere Variation in den RSSI-Werten sowie den sichtbaren BSSIDs entsteht. Dies bedeutet, bei Referenzpunkten, die physisch voneinander separiert sind, könnte eine höhere Granularität beobachtet werden. Der Entwickler von FIND gibt eine Genauigkeit von ca. 10 Quadratfuß an [6], dies sind ungefähr 0,93 Quadratmeter.

6 Fazit

In der vorliegenden Thesis wurde ein Überblick über die Indoor-Positionsbestimmung gegeben. Dabei wurden verschiedene Positionierungsmethoden und Technologien vorgestellt. Es wurde außerdem auf aktuelle fingerprintingbasierte Indoor-Positionierungssysteme eingegangen. Im Hauptteil der Thesis wurde ein konkretes Indoor-Positionierungssystem behandelt. Das behandelte IPS nennt sich FIND. FIND basiert ebenfalls auf der Methode des Fingerprinting. Mithilfe von FIND, ROS und dem Loomo wurde ein System implementiert, das die Navigation des Loomos zu einer von FIND bestimmten Position ermöglicht. Dabei wurde festgestellt, dass FIND sich für den Einsatz mit dem Loomo eignet und damit auch für den Einsatz mit anderen Systemen, die eine Android-Schnittstelle bieten.

Im Vergleich zu aktuellen Ansätzen verfolgt FIND einen eher simplen Ansatz des Fingerprinting. Dadurch kommt es beim Einsatz von FIND zu bekannten Problemen. Zu diesen gehören Geräteheterogenität, Anfälligkeit gegenüber Signalschwankungen und geringe Genauigkeit der Positionsbestimmung. Dies bedeutete für den Anwendungsfall, dass aufeinander folgende Standortvorhersagen zwischen Referenzpunkten schwankten, die gegebenenfalls nicht direkt nebeneinander lagen. Dies führt dazu, dass der Loomo nicht immer zum Referenzpunkt der Smartphones navigiert werden konnte. Außerdem kann rückblickend gesagt werden, dass es sinnvoll ist, dass das Aufbauen der Fingerprint-Datenbank mit demselben Gerät erfolgt, für das später der Standort bestimmt werden soll. Bei verschiedenen Geräten sollte außerdem die "Family"-Funktion genutzt werden, um die Geräte voneinander abzukapseln. Dabei muss bedacht werden, dass für jede "Family" ein erneutes Scannen der Referenzpunkte erforderlich ist. Des Weiteren wurde durch Experimente eine Genauigkeit von ca. 18 Quadratmetern für FIND ermittelt. Ein erster Versuch, die Positionsgenauigkeit zu verbessern, wurde ebenfalls im Rahmen der Experimente evaluiert. Dabei wurden beim Scannen nach sichtbaren APs nur die zehn APs mit den höchsten RSSI-Werten berücksichtigt. Dies hat jedoch in einer leicht geringeren Positionierungsgenauigkeit resultiert.

Abschließend kann gesagt werden, dass vor dem Einsatz von FIND genau überlegt werden sollte, wie hoch die Anforderungen an die Positionsgenauigkeit und gegebenenfalls auch an die Zuverlässigkeit der Positionsbestimmung sind. Aufgrund der geringen Genauigkeit und der Anfälligkeit für Geräteheterogenität eignet sich FIND nur unter der Berücksichtigung der eben genannten Kriterien für den Einsatz in Produktionssoftware oder der Industrie. Trotz dessen kann FIND problemlos für den Privatgebrauch eingesetzt werden, um beispielsweise Home-Automation-Anwendungen zu realisieren, da die Software durchaus brauchbare Ergebnisse liefert und auch Open Source ist.

7 Ausblick

Mit den hier durchgeführten Untersuchungen konnte gezeigt werden, dass FIND mittels Fingerprinting eine Positionierung mit Meter-Genauigkeit realisieren kann. Es kann gesagt werden, dass FIND viel Potenzial für Verbesserungen hat. Im Kapitel 2.6 sind aktuelle Lösungsansätze für IPS beschrieben. Die dort vorgestellten Optimierungen können auf FIND angewendet werden, um sowohl die Positionsgenauigkeit zu verbessern als auch anderen Schwächen entgegenzuwirken.

Konkret können rangbasierte Fingerprints anstelle von rohen Fingerprints verwendet werden, um der Geräteheterogenität entgegenzuwirken. Weiter kann nach dem Aufbauen der Fingerprint-Datenbank eine Filterung nach APs, die ein besonders hohes Rauschen haben und somit wenig Mehrwert für die Wiedererkennung von Referenzpunkten mit sich bringen, ausgeführt werden. Dies hat in [52] zu einer Verbesserung der Robustheit des IPS geführt. Während FIND die Ergebnisse mehrere Machine Learning Algorithmen evaluiert und nur ein Ergebnis eines Algorithmus als Positionsvorhersage verwendet, setzen andere IPS auf nur einen Algorithmus beziehungsweise kombinieren mehrere Algorithmen dahingehend, dass die Ausgabe eines Algorithmus im nächsten weiterverarbeitet wird [52][54][35]. Dieser Ansatz bringt bessere Resultate als sie bei FIND beobachtet wurden. Abschließend kann gesagt werden, dass die Implementierung eines von Grund auf neuen IPS durchaus sinnvoll ist.

Literaturverzeichnis

- [1] *2019Vfg135_UWB.Pdf*. https://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/Telekommunikation/Unternehmen_Institutionen/Frequenzen/Allgemeinzuteilungen/FunkanlagenGeringerReichweite/2019Vfg135_UWB.pdf?__blob=publicationFile&v=2. – (Zuletzt besucht am 24.11.2021)
- [2] *Apache Kafka*. <https://kafka.apache.org/intro>. – (Zuletzt besucht am 29.11.2021)
- [3] *Configure On-Device Developer Options*. <https://developer.android.com/studio/debug/dev-options>. – (Zuletzt besucht am 04.12.2021)
- [4] *Einführung in das Robot Operating System | heise Developer*. <https://www.heise.de/developer/artikel/Einfuehrung-in-das-Robot-Operating-System-3273655.html>. – (Zuletzt besucht am 07.07.2021)
- [5] *Foreground Services*. <https://developer.android.com/guide/components/foreground-services>. – (Zuletzt besucht am 04.12.2021)
- [6] *Frequently Asked Questions · FIND3*. <https://www.internalpositioning.com/doc/faq.md>. – (Zuletzt besucht am 03.07.2021)
- [7] *IEEE 802.11 / WLAN-Topologie (IBSS BSS ESS)*. <https://www.elektronik-kompodium.de/sites/net/0907071.htm>. – (Zuletzt besucht am 13.10.2021)
- [8] *Nodes - ROS Wiki*. <https://wiki.ros.org/Nodes>. – (Zuletzt besucht am 08.12.2021)
- [9] *ROS/Introduction - ROS Wiki*. <http://wiki.ros.org/ROS/Introduction>. – (Zuletzt besucht am 07.07.2021)

- [10] *SDK Platform Release Notes | Android Developers.* <https://developer.android.com/studio/releases/platforms>. – (Zuletzt besucht am 02.12.2021)
- [11] *Services Overview.* <https://developer.android.com/guide/components/services>. – (Zuletzt besucht am 04.12.2021)
- [12] *Setting up the Server · FIND3.* https://www.internalpositioning.com/doc/server_setup.md. – (Zuletzt besucht am 31.07.2021)
- [13] *Smart4cads / LoomoCaDSService.* <https://git.haw-hamburg.de/smart4cads/LoomoCaDSService>. – (Zuletzt besucht am 03.12.2021)
- [14] *Spring Boot.* <https://spring.io/projects/spring-boot>. – (Zuletzt besucht am 07.12.2021)
- [15] *Toggle Buttons.* <https://developer.android.com/guide/topics/ui/controls/togglebutton>. – (Zuletzt besucht am 04.12.2021)
- [16] *Welcome to Python.Org.* <https://www.python.org/about/>. – (Zuletzt besucht am 11.12.2021)
- [17] *Wi-Fi Scanning Overview | Android Developers.* <https://developer.android.com/guide/topics/connectivity/wifi-scan>. – (Zuletzt besucht am 04.12.2021)
- [18] *Eine Übersicht Über In- Und Outdoor Positionierungssysteme.* <https://docplayer.org/57322352-Eine-uebersicht-ueber-in-und-outdoor-positionierungssysteme.html>. 2005. – (Zuletzt besucht am 23.09.2021)
- [19] BATISTIĆ, Luka ; TOMIĆ, Mladen: Overview of Indoor Positioning System Technologies. In: *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Mai 2018, S. 0473–0478
- [20] CHINTALAPUDI, Krishna ; PADMANABHA IYER, Anand ; PADMANABHAN, Venkata N.: Indoor Localization without the Pain. In: *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*. New York, NY, USA : Association for Computing Machinery, September 2010 (MobiCom '10), S. 173–184. – ISBN 978-1-4503-0181-7

- [21] FARAGHER, Ramsey ; HARLE, Robert: Location Fingerprinting With Bluetooth Low Energy Beacons. In: *IEEE Journal on Selected Areas in Communications* 33 (2015), November, Nr. 11, S. 2418–2428. – ISSN 1558-0008
- [22] FLUSS, Ronen ; FARAGGI, David ; REISER, Benjamin: Estimation of the Youden Index and Its Associated Cutoff Point. In: *Biometrical Journal* 47 (2005), Nr. 4, S. 458–472. – ISSN 1521-4036
- [23] GROUP, PostgreSQL Global D.: *PostgreSQL*. <https://www.postgresql.org/>. 2021-12-07T21:50:58.054306. – (Zuletzt besucht am 07.12.2021)
- [24] GUO, Guangyi ; CHEN, Ruizhi ; YE, Feng ; PENG, Xuesheng ; LIU, Zuoya ; PAN, Yuanjin: Indoor Smartphone Localization: A Hybrid WiFi RTT-RSS Ranging Approach. In: *IEEE Access* 7 (2019), S. 176767–176781. – ISSN 2169-3536
- [25] GUSTRAU, Frank: *Hochfrequenztechnik: Grundlagen der mobilen Kommunikationstechnik ; mit 58 Beispielen und 39 Übungsaufgaben*. 2., überarb. und erw. Aufl. München : Fachbuchverl. Leipzig im Carl-Hanser-Verlag, 2013. – ISBN 978-3-446-43245-1
- [26] HECHENBICHLER, Schliep: Weighted K-Nearest-Neighbor Techniques and Ordinal Classification. (2004), S. 17
- [27] HÖFLINGER, Fabian ; BORDOY, Joan ; SIMON, Nikolas ; WENDEBERG, Johannes ; REINDL, Leonhard M. ; SCHINDELHAUER, Christian: Indoor-Localization System for Smart Phones. In: *2015 IEEE International Workshop on Measurements Networking (M N)*, Oktober 2015, S. 1–6
- [28] KAPLAN, Elliott D. ; HEGARTY, Christopher: *Understanding GPS/GNSS: Principles and Applications, Third Edition*. Artech House, Mai 2017. – ISBN 978-1-63081-442-7
- [29] KLEINE, Jonathan: *Mobiles WLAN-basiertes Positionierungssystem für den Indoorbereich des Departments für Geodäsie und Geoinformation*, Wien, Thesis, 2021
- [30] KÖLMEL, Bernhard: *Location based services*. Gesellschaft für Informatik e.V., 2003. – ISBN 978-3-88579-354-0
- [31] LEB, Alexander: *Studie für ein campusweites Positionierungs- und Informationssystem im Indoor- und Outdoorbereich*, Wien, Thesis, 2020

- [32] LIU, Hui ; DARABI, Houshang ; BANERJEE, Pat ; LIU, Jing: Survey of Wireless Indoor Positioning Techniques and Systems. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37 (2007), November, Nr. 6, S. 1067–1080. – ISSN 1558-2442
- [33] LTD, Raspberry P.: *Buy a Raspberry Pi 4 Model B.* <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>. – (Zuletzt besucht am 02.12.2021)
- [34] LUNTOVSKYY, Andriy ; GÜTTER, Dietbert ; MELNYK, Igor: *Planung und Optimierung von Rechnernetzen.* Wiesbaden : Vieweg+Teubner Verlag, 2012. – ISBN 978-3-8348-1458-6 978-3-8348-8242-4
- [35] MA, Zixiang ; WU, Bang ; POSLAD, Stefan: A WiFi RSSI Ranking Fingerprint Positioning System and Its Application to Indoor Activities of Daily Living Recognition. In: *International Journal of Distributed Sensor Networks* 15 (2019), April, Nr. 4, S. 1550147719837916. – ISSN 1550-1477
- [36] MAINETTI, Luca ; PATRONO, Luigi ; SERGI, Ilaria: A Survey on Indoor Positioning Systems. In: *2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, September 2014, S. 111–120
- [37] MÜLLER, Andreas C. ; GUIDO, Sarah: *Introduction to Machine Learning with Python: A Guide for Data Scientists.* First edition. Sebastopol, CA : O'Reilly Media, Inc, 2016. – ISBN 978-1-4493-6941-5
- [38] MÜLLER, Manuel ; MÜLLER, Timo ; TALKHESTANI, Behrang A. ; MARKS, Philipp ; JAZDI, Nasser ; WEYRICH, Michael: Industrial Autonomous Systems: A Survey on Definitions, Characteristics and Abilities. In: *at - Automatisierungstechnik* 69 (2021), Januar, Nr. 1, S. 3–13. – ISSN 2196-677X
- [39] O’KANE, Jason M.: *A Gentle Introduction to ROS.* Version 2.1.1 (3e3d9c5), generated in November 20, 2014. Columbia, SC : Jason M. O’Kane, 2014. – ISBN 978-1-4921-4323-9
- [40] PASKU, Valter ; DE ANGELIS, Alessio ; DE ANGELIS, Guido ; ARUMUGAM, Darmindra D. ; DIONIGI, Marco ; CARBONE, Paolo ; MOSCHITTA, Antonio ; RICKETTS, David S.: Magnetic Field-Based Positioning Systems. In: *IEEE Communications Surveys Tutorials* 19 (2017), Nr. 3, S. 2003–2017. – ISSN 1553-877X

- [41] RETSCHER, Guenther ; TATSCHL, Thomas: Indoor Positioning Using Wi-Fi Lateration — Comparison of Two Common Range Conversion Models with Two Novel Differential Approaches. In: *2016 Fourth International Conference on Ubiquitous Positioning, Indoor Navigation and Location Based Services (UPINLBS)*, November 2016, S. 1–10
- [42] RETSCHER, Günther ; KISTENICH, Michael: Vergleich von Systemen zur Positionsbestimmung und Navigation in Gebäuden. (2006), S. 11
- [43] RETSCHER, Günther ; TATSCHL, Thomas: Positionierung in Gebäuden mit differenziellem WLAN. In: *zfv – Zeitschrift für Geodäsie, Geoinformation und Landmanagement* (2017), Nr. 2/2017, S. 111–125. – ISSN 1618-9050
- [44] ROPPEL, Carsten: *Grundlagen Der Digitalen Kommunikationstechnik: Übertragungstechnik – Signalverarbeitung – Netze*. München : Carl Hanser Verlag GmbH & Co. KG, März 2006. – ISBN 978-3-446-22857-3 978-3-446-40879-1
- [45] RUSSELL, Stuart J. ; NORVIG, Peter: *Artificial Intelligence: A Modern Approach*. Fourth edition. Hoboken : Pearson, 2020 (Pearson Series in Artificial Intelligence). – ISBN 978-0-13-461099-3
- [46] SACCOMANNO, Nicola ; BRUNELLO, Andrea ; MONTANARI, Angelo: Let's Forget About Exact Signal Strength: Indoor Positioning Based on Access Point Ranking and Recurrent Neural Networks. In: *MobiQuitous 2020 - 17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. New York, NY, USA : Association for Computing Machinery, Dezember 2020 (MobiQuitous '20), S. 215–224. – ISBN 978-1-4503-8840-5
- [47] SICILIANO, Bruno (Hrsg.) ; KHATIB, Oussama (Hrsg.): *Springer Handbook of Robotics*. Berlin : Springer, 2008. – ISBN 978-3-540-23957-4
- [48] STEEN, Maarten van ; TANENBAUM, Andrew S.: *Distributed Systems*. Third edition, Version 3.01. Erscheinungsort nicht ermittelbar : Maarten van Steen, 2017. – ISBN 978-1-5430-5738-6
- [49] TANENBAUM, Andrew S. ; WETHERALL, David J.: *Computernetzwerke*. Pearson Deutschland, August 2012. – ISBN ISBN 9783868941371
- [50] TATSCHL, Thomas: *Indoor Positionierung mit differenziellem WLAN*, Wien, Thesis, 2015

- [51] VALDERAS, Daniel ; SANCHO, Juan I. ; PUENTE, David ; LING, Cong ; CHEN, Xiaodong: *Ultrawideband Antennas: Design and Applications*. IMPERIAL COLLEGE PRESS, Oktober 2010. – ISBN 978-1-84816-491-8 978-1-84816-492-5
- [52] XUE, Jianzhe ; LIU, Junyu ; SHENG, Min ; SHI, Yan ; LI, Jiandong: A WiFi Fingerprint Based High-Adaptability Indoor Localization via Machine Learning. In: *China Communications* 17 (2020), Juli, Nr. 7, S. 247–259. – ISSN 1673-5447
- [53] ZACK: *FIND3 Android App*. November 2021
- [54] ZHANG, Le ; CHEN, Zhenghua ; CUI, Wei ; LI, Bing ; CHEN, Cen ; CAO, Zhiguang ; GAO, Kaizhou: WiFi-Based Indoor Robot Positioning Using Deep Fuzzy Forests. In: *IEEE Internet of Things Journal* 7 (2020), November, Nr. 11, S. 10773–10781. – ISSN 2327-4662

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort

Datum

Unterschrift im Original