



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Entwicklung eines hybriden Indoor-Outdoor-Lokalisierungssystems auf Basis der ARCore Geospatial API und Cloud Anchor

Masterarbeit

Name des Studiengangs
Angewandte Informatik

Fachbereich 4

vorgelegt von
Yannik Morhenn

Datum:
Berlin, 21.03.2023

Erstgutachter: Prof. Dr.-Ing. Thomas Jung

Zweitgutachter: Christoph Holtmann

Abstract

Die präzise globale Lokalisierung von mobilen Endgeräten in Innenräumen ist eine Aufgabe, die noch nicht vollständig gelöst ist. Für die relative Ortung in Relation zu bestimmten Referenzpositionen gibt es bereits einige Lösungen, allerdings können die geografischen Koordinaten innerhalb von Gebäuden bisher nicht genau bestimmt werden.

In dieser Arbeit wurde eine Android-Anwendung entwickelt, die genaue globale Ortung in Gebäuden ermöglicht. Dafür muss mit der Anwendung zunächst ein Gebäudeteil abgelaufen und dabei ausgemappt werden, welcher dann folgenden Nutzern über einen Server zur Verfügung steht. Augmented Reality und die ARCore Cloud Anchor API werden zur genauen Positionierung innerhalb eines Gebäudeplanes verwendet, während mit der ARCore Geospatial API eine globale Referenz zu einem erstellten Gebäudeplan hinzugefügt wird, anhand dessen für jeden Punkt einer Karte die genauen geografischen Koordinaten berechnet werden können.

Die Evaluation der implementierten Anwendung und der zugrundeliegenden Technologien zeigt sehr vielversprechende Ergebnisse im Bereich der Genauigkeit und Zuverlässigkeit. Dabei liegen die größten Limitationen bei dem potenziellen relativen Drift des SLAM-Trackings beim Erstellen eines Gebäudeplanes sowie bei möglichen auftretenden Ungenauigkeiten beim Lokalisieren anhand der Cloud Anchor.

[Englisch]

Precise global localization of mobile devices inside indoor environments is a task that has not been fully solved yet. While localization in relation to known reference points already has a number of viable solutions, indoor positioning with global coordinates is still very hard to accurately obtain.

In this thesis, an android application is built, that enables a user to accurately locate himself inside any indoor environment. To achieve this, the desired area has to first be mapped out with the android app and is then available to all future users. Augmented Reality and the ARCore Cloud Anchor API are used to find a known position within a created map, while the ARCore Geospatial API is used to add a global reference with coordinates to a created map, from which the global coordinates can be calculated for any point of the map.

The analysis of the implemented application and its underlying technologies shows very promising results in terms of both accuracy and reliability. The biggest limitations are the possible drift of the relative SLAM-Tracking while mapping an environment, as well as possible inaccuracies of the Cloud Anchor APIs visual positioning.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Zielstellung	2
1.3	Aufbau der Arbeit	3
2	Grundlagen	4
2.1	Globale Lokalisierung	4
2.1.1	Global Positioning System	4
2.1.2	Darstellung globaler Koordinaten	6
2.2	Methoden für Indoor-Lokalisierung	8
2.2.1	WiFi-Fingerprinting	8
2.2.2	BLE-Beacons	9
2.2.3	Magnetfeld	10
2.3	Lokalisierung mit Augmented Reality	10
2.3.1	Sensorgestützte relative Positionsverfolgung	11
2.3.2	Simultaneous Localization and Mapping	11
2.3.3	Markerbasierte Ortung	12
2.3.4	Cloud Anchor	12
2.3.5	Visual Positioning System	13
3	Analyse	15
3.1	Stand der Wissenschaft	15
3.1.1	Funkbasierte Ortung mit Augmented Reality	15
3.1.2	Building Information Modeling	16
3.1.3	Cloud Anchor Lokalisierung	16
3.1.4	Visuelle Lokalisierung im Stadtraum	17
3.2	Anforderungsanalyse	18
3.2.1	Muss-Kriterien	19
3.2.2	Soll-Kriterien	19
3.2.3	Abrenzungskriterien	20
3.2.4	Limitationen	20
4	Systementwurf	22
4.1	App Übersicht	22
4.2	Datenhaltung	23
4.2.1	Datenmodell	23
4.2.2	Firebase	24
4.2.3	Firebase Authentication	25

4.3	Augmented Reality Komponenten	25
4.3.1	ARCore Render-Engine	26
4.3.2	ARCore APIs	26
4.4	System-Architektur	27
5	Implementierung	28
5.1	Übersicht	28
5.2	Login-Fragment	28
5.3	Gebäudeplan Übersicht	29
5.4	Erstellung eines Gebäudeplanes	30
5.5	Lokalisierung anhand eines Gebäudeplanes	32
5.6	Tracking Test	35
5.7	Datenexport	36
6	Untersuchung	37
6.1	Zielsetzung	37
6.2	Methodik	38
6.3	Testdurchführung	39
6.3.1	Erstellen eines Planes	39
6.3.2	Lokalisierung	50
6.4	Ergebnisse	60
6.5	Weitere Beobachtungen	62
6.5.1	Akkuverbrauch	62
6.5.2	Datennutzung	62
7	Fazit	63
8	Ausblick	64
9	Literaturverzeichnis	66
10	Abbildungsverzeichnis	71
11	Tabellenverzeichnis	71
12	Abkürzungsverzeichnis	72

1 Einleitung

1.1 Motivation

Der Prozess der Lokalisierung eines Mobilgerätes scheint für viele Benutzer und Anwendungsfälle bereits durch die Nutzung von satellitengestützten Navigationssystemen wie GPS gelöst, mit denen vor allem draußen und unter freiem Himmel die Position eines Smartphones ziemlich genau bestimmt werden kann. Aufgrund der physikalischen Limitierungen dieser Systeme leidet die Genauigkeit allerdings drastisch bei der Nutzung in urbanen Gebieten, vor allem in der Nähe von höheren Bauwerken. Innerhalb von Gebäuden ist die Ortung mithilfe dieser Systeme gar keine Option, da die Signale der Satelliten durch den Beton moderner Gebäude nahezu vollständig blockiert werden.

Es gibt bereits diverse Ansätze für genaue Indoor-Ortungssysteme, die allerdings meist nur relative Positionen zu festgelegten Referenzpunkten bestimmen können, während die exakten globalen Koordinaten nicht bestimmbar sind. Augmented Reality auf modernen Smartphones und die damit einhergehenden neuen Möglichkeiten zur Nutzung visueller Tracking-Algorithmen und Methoden zur Abgleichung von Punktwolken bieten ganz neue Optionen zur Lokalisierung an Orten, wo genaue GPS-Ortung keine Option darstellt.

Eine der wichtigsten neuen Technologien, die diese Arbeit in der geplanten Form überhaupt erst möglich macht, ist die im Mai 2022 von Google vorgestellte ARCore Geospatial API [1], die mit ihrem Visual Positioning System (VPS) eine genaue globale Lokalisierung durch Abgleichen der Kamerabilder mit Street View Daten ermöglicht. Solange sich das Gerät in einem Gebiet mit Street View befindet, kann so auch etwa in Häuserschluchten die Position eines Smartphones ca. metergenau bestimmt werden.

Im Rahmen eines vorangegangenen Forschungsprojektes wurde diese Technologie bereits verwendet, um relativ erstellte Routen in Augmented Reality mit einer globalen Referenz zu versehen und so einen simplen und nutzerfreundlichen Übergang von herkömmlichen 2D-Karten mit GPS-Ortung zur Navigation mit Augmented Reality zu ermöglichen. Als Übergang und Ankerpunkt der relativen Routen wurde als weitere Kerntechnologie die ARCore Cloud Anchor Bibliothek [2] verwendet, die es durch Scannen der direkten Umgebung ermöglicht, eine Punktwolke zu erstellen, mit der zu einem späteren Zeitpunkt durch Abgleichen mit dem Kamerabild die genaue Position bestimmt und das dazugehörige relative Koordinatensystem wiederverwendet werden kann.

Aufgrund der sehr positiven Ergebnisse dieses Projektes soll an die Arbeit angeknüpft werden und ein System entwickelt werden, mit dem durch relative Positionsverfolgung in Augmented Reality und Cloud Anchor als Referenzpunkte ein Gebäude genau ausgemappt werden kann. Zusätzlich dazu soll die Geospatial API dazu verwendet werden, eine globale Referenz für einen erstellten Gebäudeplan zu bestimmen, wodurch die globalen geografischen Koordinaten des gesamten Planes errechnet werden können.

Anhand der so erstellten Gebäudepläne soll ein Nutzer sich anschließend genau lokalisieren können, solange er sich innerhalb des ausgemappten Bereiches befindet. Damit ein Nutzer an einer beliebigen Stelle innerhalb des Planes einsteigen kann, sollen bei der Erstellung regelmäßig Cloud Anchor platziert werden, zu denen das Smartphone initial positioniert werden kann und die im Laufe der Lokalisierung als Kalibrierungspunkte verwendet werden können, um Fehlern und Ungenauigkeiten entgegenwirken zu können.

1.2 Zielstellung

Im Rahmen dieser Arbeit soll zunächst eine Android-App entwickelt werden, die es jedem Benutzer ermöglicht, ein beliebiges Gebäude auszumappen und einen entsprechenden Gebäudeplan zu erstellen, der für andere Nutzer später zur Lokalisierung verwendet werden kann, ohne dass dafür weitere externe Geräte oder Signale benötigt werden. Dabei soll ein lokalisierender Nutzer im gemappten Bereich eines Gebäudes lediglich die Kamera seines Smartphones kurz bewegen müssen, um die genau bestimmte eigene Position mit den Koordinaten angezeigt bekommen zu können.

Die Implementation der Anwendung und die zugrundeliegenden Technologien sollen anschließend ausgiebig auf Zuverlässigkeit und Genauigkeit untersucht werden, wobei die folgenden beiden Forschungsfragen primär betrachtet werden sollen:

Forschungsfrage 1:

Kann durch die Kombination von ARCore Cloud Anchor und der Geospatial API eine bessere und zuverlässigere Indoor-Lokalisierung ermöglicht werden, als mit relativem VPS-Tracking nach Verlust von Street View Daten möglich ist?

Forschungsfrage 2:

Können automatisch Cloud Anchor im Hintergrund geladen werden, um damit das System regelmäßig zu kalibrieren und um dem ARCore Drift entgegenzuwirken, aber auch um eine Lokalisierung mit Start im Gebäude zu ermöglichen?

1.3 Aufbau der Arbeit

Zu Beginn dieser Arbeit werden zunächst einige existierende Technologien für die Lokalisierung von Mobilgeräten vorgestellt, während im Anschluss kurz Augmented Reality und dessen verwendete Bild-basierte Methoden für relative Positionsverfolgung und Lokalisierung aufgezeigt werden. In der folgenden Analyse werden einige konkrete Anwendungen aus der Literatur vorgestellt, die genaue Lokalisierung für Augmented Reality Anwendungen implementieren und anschließend die Anforderungen an die zu entwickelnde Anwendung aufgestellt.

Aus diesen Anforderungen wird im Anschluss ein Systementwurf erstellt und die Implementierung und dessen typische Nutzungsabläufe der Anwendung anhand von Screenshots vorgestellt. Darauf folgend wird die entwickelte Anwendung und die zugrundeliegenden Technologien ausgiebig untersucht, wofür einige Hypothesen aufgestellt und evaluiert werden. Zuletzt werden anhand eines Fazits die gewonnenen Kenntnisse formuliert und mögliche Erweiterungen der Anwendung im Ausblick vorgestellt.

2 Grundlagen

Die genaue Positionsbestimmung von Personen ist eine Herausforderung seit vielen Jahren, die heutzutage aufgrund der weiten Verbreitung von Smartphones mit der Positionsbestimmung dieser mobilen Endgeräte gleichgestellt werden kann, wenn gleich diese in zwei grundlegende Arten unterteilt wird. Zunächst gibt es die reine Positionsverfolgung in Abhängigkeit zu einer Ausgangsposition, also lediglich die Bestimmung der relativen Bewegung und Rotation der sechs Freiheitsgrade (6DoF) eines Gerätes, welches im Folgenden als 6DoF-Tracking oder relative Positionsverfolgung beschrieben wird. Auf der anderen Seite steht die Lokalisierung oder Ortung, bei der die konkrete Position eines Gerätes innerhalb eines Koordinatensystems bestimmt wird, wobei hier weiter in die Lokalisierung anhand von globalen geografischen Koordinaten oder der relativen Lokalisierung anhand von einem festgelegten Referenzpunkt und damit einem relativen Koordinatensystem unterschieden werden kann.

Im Rahmen dieses Kapitels wird zunächst das Konzept und die Einschränkungen der globalen Lokalisierung mit satellitengestützten Systemen erläutert und im Folgenden einige Technologien zur Ortung innerhalb von Gebäuden vorgestellt, die in der Regel nur relative Positionen zu festgelegten Referenzpunkten bestimmen können. Im Anschluss wird Augmented Reality und dessen verschiedene Methoden der Bild-basierten Positionsbestimmung präsentiert, wobei zunächst die Ansätze für 6DoF-Tracking betrachtet werden, gefolgt von Methoden für visuelle Lokalisierung.

2.1 Globale Lokalisierung

Der Prozess der globalen Positionsbestimmung beschreibt die Ortung eines Gerätes und die Beschreibung der bestimmten Position anhand von Koordinaten innerhalb eines festgelegten Koordinatenreferenzsystems. Als Referenzsystem für die Koordinaten ist das „World Geodetic System 1984“ (WGS84) heute weit verbreitet, da es die geodätische Grundlage für GPS ist, auf das im Folgenden näher eingegangen wird. [3]

2.1.1 Global Positioning System

Der Begriff *global navigation satellite system* (GNSS, zu deutsch: Globales Navigationssatellitensystem) umfasst alle Systeme, die Positionsbestimmung und Navigation mittels Satellitensignalen ermöglichen. Das bekannteste GNSS ist das amerikanische „Global Positioning System“ (GPS), welches im Rahmen dieser Arbeit näher

betrachtet wird, allerdings gelten nahezu die gleichen Faktoren auch für die alternativen GNSS: *GLONASS*, *Galileo* und *Beidou*. [3]



Abbildung 1: GPS-Ortung anhand von 4 sichtbaren Satelliten [4]

Die Lokalisierung anhand von GPS basiert auf der Empfängerseite auf einer rein passiven Ortungsmethode, die durch eine Konstellation an Satelliten ermöglicht wird, welche dauerhaft ein Signal mit ihrer aktuellen Position und dem genauen Zeitpunkt aussenden. Da die Ausbreitungsgeschwindigkeit der gesendeten Signale bekannt ist, kann anhand des gemessenen Zeitunterschiedes beim Empfang der Abstand zu einem Satelliten bestimmt werden, woraus sich eine Kugel an möglichen Positionen ergibt. Wie in Abbildung 1 dargestellt, kann so durch Trilateration anhand von mindestens 3 Satelliten die Position des Empfängers als Schnittpunkt aller möglichen Positionen bestimmt werden. Aufgrund von Ungenauigkeiten in der Uhr des Empfängers ist in der Realität für die genaue Lokalisierung mindestens ein vierter sichtbarer Satellit notwendig, um die entsprechende Zeitfehler korrigieren zu können. [3] [4]

Laut dem *National Coordination Office for Space-Based Positioning* [5] liegt die Abweichung der Positionsbestimmung eines Smartphones in einer geeigneten Umgebung unter freiem Himmel im Durchschnitt bei weniger als 4,9 Metern, wobei mit professionellen Geräten und der Nutzung einer zweiten Signalfrequenz sogar eine Genauigkeit im Zentimeterbereich erreicht werden kann.

Die größte Einschränkung von satellitengestützten Ortungssystemen tritt bei der Verwendung in urbanen Gebieten und vor allem in der Nähe von hohen Bauwerken auf, da aufgrund der Beschaffenheit der versendeten Signale eine direkte Sichtlinie zwischen Empfänger und Satellit notwendig ist. Des Weiteren kann es zu einem Phänomen namens Multipathing kommen, wo Satellitensignale aufgrund von Reflexionen auf nicht direktem Weg zum Empfänger gelangen, wie es in Abbildung 2 zu sehen ist. Dadurch wird eine falsche Distanz zum Satelliten bestimmt, wodurch die Berechnung der Empfängerposition entsprechend verfälscht wird. [6]

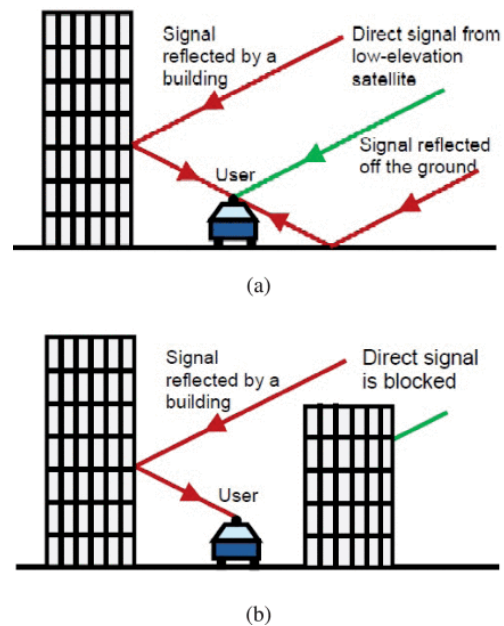


Abbildung 2: Multipathing Effekt bei GPS-Ortung [6]

Da die Signale der Satelliten durch Bauwerke vollständig blockiert werden, ist der Empfang und damit die Nutzung von GPS innerhalb von Gebäuden oder in beispielsweise U-Bahnhöfen nicht möglich.

Moderne Smartphones benutzen heutzutage eine augmentierte Form der GPS-Ortung, die unter dem Begriff A-GPS (Assisted GPS) bekannt ist. Dabei wird beispielsweise in Android neben den Satellitensignalen zunächst eine Ortung mittels naher Mobilfunkmasten und WiFi-Signalen durchgeführt, anhand dessen eine ungefähre Positionsbestimmung zur Überbrückung bis zur mit GPS bestimmten Position stattfinden kann. So kann auch innerhalb von Gebäuden oder während der Suche nach einer ausreichenden Zahl an GPS-Satelliten fast sofort die zunächst ungefähre Position bestimmt werden. [7]

2.1.2 Darstellung globaler Koordinaten

Um globale Koordinaten darzustellen, werden in der Regel zweidimensionale Karten verwendet, die mithilfe einer Kartenprojektion den Globus der Erde auf einer Karte darstellen. Dabei werden von nahezu allen webbasierten Kartendiensten wie auch von Google Maps die sogenannte *Web Mercator* Projektion verwendet. [8]

Da im Rahmen dieser Arbeit eine Anwendung für Indoor-Ortung mit globalen Koordinaten entwickelt werden soll, gibt es einige Optionen an Diensten, um diese Koordinaten innerhalb von Gebäuden sinnvoll darstellen zu können. So bietet beispielsweise Google Maps an ausgewählten Orten wie vor allem in Einkaufszentren, Universitäten und Bahnhöfen die Möglichkeit, die Grundrisse der einzelnen Etagen darzustellen. Da die Daten dieser Gebäudepläne von den Inhabern selbstständig hinzugefügt und anhand von Satellitenbildern in Google Maps ausgerichtet werden müssen, sind nur eine begrenzte Zahl an Indoor-Karten verfügbar. [9]

Neben Google Maps Indoor-Karten gibt es einige auf *OpenStreetMap* [10] basierende Alternativen, die vor allem im Stadtbereich Berlins in ausgewählten Bereichen eine höhere Abdeckung an verfügbaren Gebäudeplänen haben. Vor allem im Bereich von Bahnhöfen verfügt Google Maps in Berlin zum aktuellen Zeitpunkt mit den Bahnhöfen Südkreuz und Ostbahnhof nur über zwei verschiedene Indoor-Karten, während andere OSM-Dienste wie *Openstationmap* [11] oder *Indoor=* [12] eine deutlich höhere Abdeckung anbieten. Dabei stehen allerdings diese Gebäudepläne nur lückenhaft zur Verfügung und auch die Darstellung und der Detailgrad sind häufig nicht konsistent.

Auch direkt in *OpenStreetMap* [10] sind von einigen Bahnhöfen die Indoor-Karten und Schienenpläne integriert, allerdings fehlen dort die Möglichkeiten zur Auswahl einzelner Etagen und es sind immer alle verfügbaren Daten übereinander gelagert sichtbar, woran die Übersichtlichkeit leidet. Des Weiteren sind auch hier Pläne mit sehr unterschiedlichen Detailgraden zu finden, während die in Google Maps verfügbaren Gebäudepläne in der Regel immer vollständig wirken.

Die zu implementierende Anwendung würde zwar durchaus von der größeren Menge an verfügbarem Indoor-Kartenmaterial profitieren, allerdings ist die Vollständigkeit der einzelnen Karten und die visuelle Vergleichbarkeit von sehr großer Bedeutung, um später die Anwendung gut untersuchen zu können. Des Weiteren sind sehr geeignete Gebäudetypen wie Universitäten und Einkaufszentren vor allem in Google Maps weiter verbreitet.

Um die Ergebnisse der Untersuchungen dieser Arbeit später auch anhand von Indoor-Karten visualisieren zu können, kann *Google MyMaps* [13] verwendet werden, ein Dienst von Google, der es ermöglicht, auf die normalen Kartendaten mehrere Ebenen an eigen gezeichneten Informationen darzustellen. Die darzustellenden Informationen können mithilfe des Datenformates *KML* [14] importiert werden, eine für Geoinformationen abgewandelte Form von *XML*, die einfach programmatisch erstellt werden kann.

2.2 Methoden für Indoor-Lokalisierung

Zur Lokalisierung von Personen und Smartphones innerhalb von Gebäuden gibt es diverse verschiedene Lösungsansätze an unterschiedlichsten Technologien, mit denen allerdings die Positionen in der Regel nur relativ zu einem Referenzpunkt im Gebäude und nicht zu einem globalen Koordinatenreferenzsystem, wie etwa bei GPS, bestimmt werden. Im Folgenden werden einige der meist verbreitetsten und geeignetsten Ansätze für diese relative Lokalisierung innerhalb von Gebäuden vorgestellt, wobei der Fokus auf die Technologien gelegt wird, die mit den in modernen Smartphones verbauten Sensoren einfach verwendet werden können.

2.2.1 WiFi-Fingerprinting

Eine für die Lokalisierung innerhalb von Gebäuden häufig am einfachsten umzusetzende Methode ist das sogenannte WiFi-Fingerprinting, was anhand der empfangenen Signalstärken (RSS, received signal strength) der existierenden umliegenden WLAN-Netze die Lokalisierung vornimmt. Dabei müssen im Voraus sogenannte Fingerprints an vielen Orten eines Gebäudes erstellt werden, an denen die einzelnen Signalstärken der verschiedenen WLAN-Signale in Relation zueinander gespeichert werden. So kann bei einer späteren Lokalisierung die aktuelle Kombination an Signalstärken mit den erstellten Fingerprints verglichen und eine bestmögliche Übereinstimmung gefunden werden. [15]

Durch die Abhängigkeit zu den umgebenen WLAN-Netzen hat dieses Verfahren einige Einschränkungen und Probleme, die vor allem von der benötigten Kontinuität der Signalstärken abhängig ist. So können beispielsweise temporäre WLAN-Hotspots die erstellten Fingerprints zu einem späteren Zeitpunkt unbrauchbar machen oder bauliche Veränderungen dafür sorgen, dass die Signalstärken an den gleichen Orten mit anderer Stärke gemessen werden. Viel und Asplund [16] zeigen des weiteren auch stark unterschiedliche Messungen an Signalstärken zwischen verschiedenen Smartphones auf, was es ebenfalls schwierig für die Nutzung zur genauen Indoor-Lokalisierung von Mobilgeräten macht. Laut Davidson und Piché [15] kann mithilfe von WLAN-Fingerprinting in Indoor-Umgebungen eine Genauigkeit von etwa 10 Metern erreicht werden, während Yassin et. al. [17] einen Fehlerbereich von 6 bis 8 Metern beschreiben.

Neben dem Prinzip des RSS-Fingerprintings gibt es auch eine Menge anderer Methoden, um die Position anhand von empfangenen WLAN-Signalen zu bestimmen, wie unter anderem die in [17], [18] und [19] beschriebene *Time of Arrival* (ToA), *Time Difference of Arrival* (TDoA) und *Angle of Arrival* (AoA). Die nähere Erläuterung dieser Methoden übersteigt allerdings den Rahmen dieser Arbeit, sind aber in

Genauigkeit und Problemen mit RSS vergleichbar.

2.2.2 BLE-Beacons

Eine weitere häufig verwendete Methode zur Indoor-Lokalisierung ist die Verwendung von sogenannten BLE-Beacons, also kleinen Hardware-Geräten, die mit Bluetooth Low-Energy ausgestattet sind und die so mit nur sehr geringem Stromverbrauch ebenfalls Signale im 2,4 GHz Bereich aussenden können. Als tatsächliche Methode zur Positionsbestimmung kann ebenfalls eine Form des RSS-Fingerprintings stattfinden, wobei ähnlich wie mit WiFi-Signalen auch andere Methoden neben der Nutzung der empfangenen Signalstärke möglich sind. [17] [18] [19]

Parallel zur Ortung mit RSS-Fingerprinting, beschreiben Davidson und Piché [15] auch die Möglichkeit der direkten Positionsbestimmung bei sehr kurzem Abstand zu einem BLE-Beacon, wodurch eine Sub-Meter genaue Lokalisierung möglich sei.

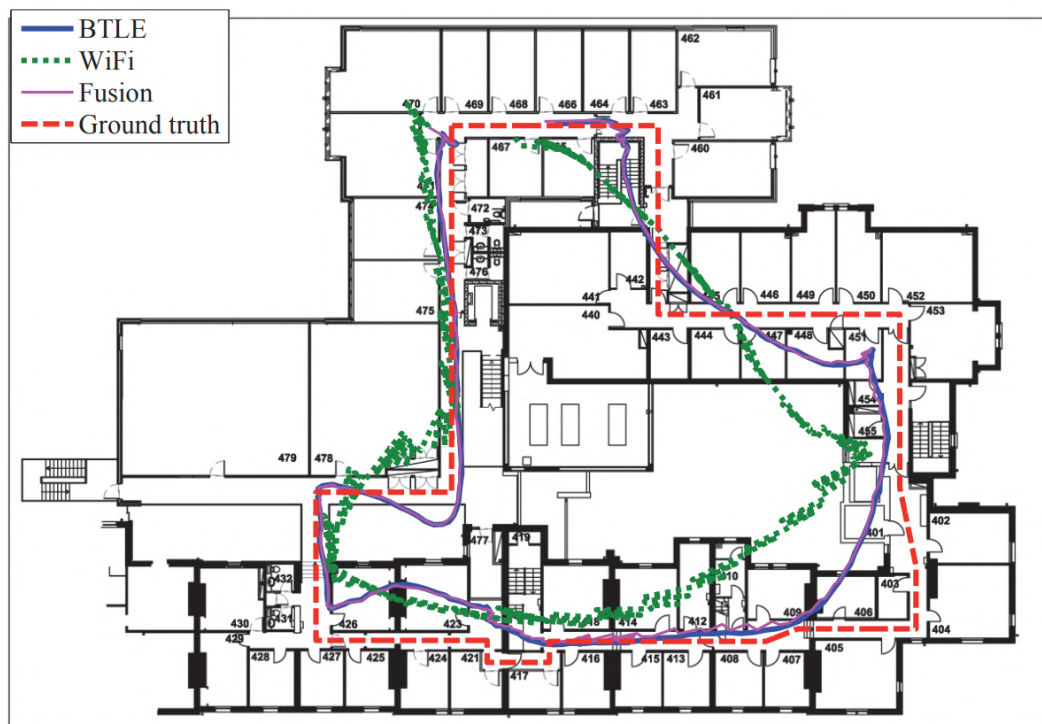


Abbildung 3: Vergleich der Lokalisierung mit BLE und WiFi [20]

Zhao et. al. [20] beschreiben, dass mithilfe von BLE-Beacons eine etwa 27 % genauere Positionsbestimmung im Vergleich zu WiFi-Signalen erfolgen kann, was anhand derer Versuchsergebnisse in Abbildung 3 weiter verdeutlicht wird. Es werden drei primäre Gründe für diese besseren Ergebnisse von BLE im Vergleich zu WiFi

genannt, zum einen der Mechanismus des Kanal-Hoppings, die geringeren Sendeleistungen und dadurch resultierende Reduzierung des Multipathing-Effektes und die deutlich höheren Abtastraten.

Insgesamt ist als größter Nachteil von BLE-gestützten Ortungssystemen der aufwendige und kostspielige Prozess der Platzierung und Wartung der einzelnen Beacons zu sehen, der bei WiFi-Fingerprinting in der Regel aufgrund von bereits existierender Infrastruktur nicht benötigt wird. Dafür kann so im Vergleich zu WiFi für eine deutlich höhere Robustheit des Ortungssystems gesorgt werden, da keine Abhängigkeit zur Beständigkeit von fremden WLAN-Routern besteht.

2.2.3 Magnetfeld

Neben von Geräten produzierten Radio-Signalen kann auch das Magnetfeld der Erde für die Lokalisierung innerhalb von Gebäuden verwendet werden. Laut Davidson und Piché [15] entstehen Störungen und Abweichungen im messbaren Erdmagnetfeld durch die Metallstrukturen von Gebäuden, die von den in modernen Smartphones integrierten Magnetometern gemessen werden können. So kann ähnlich wie bei der RSS-Messung von WiFi und BLE eine Datenbank an Fingerprints an verschiedenen Positionen eines Gebäudes aufgebaut werden, anhand dessen sich später relativ lokalisiert werden kann.

Mendoza-Silva et. al. [18] beschreiben die größten Vorteile dieser Technologie als kostengünstig, mit einer sehr niedrigen Komplexität, bei der ohne die Notwendigkeit weiterer Geräte ein sehr großer Bereich abgedeckt werden kann. Shu et. al. [21] zeigen allerdings auch Probleme dieser Technologie auf, sodass unterschiedliche Smartphones signifikant unterschiedliche Messungen des Magnetfeldes bestimmen können und dass für eine genaue Lokalisierung eine sehr detaillierte initiale Datenbank vor allem auch auf unterschiedlichen Höhen benötigt werden.

2.3 Lokalisierung mit Augmented Reality

Azuma [22] definiert Augmented Reality (AR) als eine Variation der virtuellen Realität, in der ein Nutzer die reale Welt weiterhin sehen kann, während virtuelle Inhalte auf diese echte Welt eingeblendet werden und so eine kombinierte erweiterte Realität entsteht.

Durch die immer performanter werdenden Smartphones können Augmented Reality Anwendungen mit zunehmender Komplexität direkt für mobile Endgeräte entwickelt werden und es ist keine eigenständige Hardware zur Nutzung mehr nötig, wobei in den letzten Jahren das Konzept von tragbaren AR-Brillen immer weiter verbreitet

wurde. So bieten bereits einige Produkte wie beispielsweise die Microsoft Hololens [23] gute Optionen zur freihändigen Nutzung von AR, die vor allem in industriellen Anwendungsgebieten Verwendung finden.

Für die Entwicklung von Augmented Reality Anwendungen für Smartphones gibt es einige unterschiedliche Frameworks, die große Teile der Implementation übernehmen und ein 6DoF-Tracking anbieten. So wird beispielsweise von Apple das nur für IOS nutzbare AR-Framework *ARKit* [24] angeboten, während Google ein konkurrierendes Produkt mit *ARCore* [25] anbietet, was sowohl für die native Implementation in Android verwendet werden kann, aber auch IOS- und Unity-Entwicklung unterstützt. Neben *ARCore* und *ARKit* gibt es auch weitere häufig kostenpflichtige Alternativen wie beispielsweise *Vuforia* [26] oder *Wikitude* [27], die sich eher auf professionelle Unternehmen fokussiert haben.

Um Inhalte innerhalb von Augmented Reality Anwendungen für einen Nutzer angenehm darstellen zu können, wird eine genaue relative Positionsverfolgung benötigt, weshalb im Folgenden zunächst zwei Methoden des 6DoF-Trackings beschrieben werden. Im Anschluss werden einige Bild-basierte Technologien zur Lokalisierung vorgestellt, wobei sowohl Ansätze zu relativen Koordinatensystemen als auch globalen Koordinaten existieren. Konkrete Anwendungen aus der Literatur mit Umsetzungen dieser Technologien in Augmented Reality Anwendungen werden erst anschließend im Rahmen des Kapitels 3.1 vorgestellt.

2.3.1 Sensorgestützte relative Positionsverfolgung

Eine sehr lange existierende Art des 6DoF-Trackings ist die rein sensorgestützte interne Positionsverfolgung eines Mobilgerätes, die nur mit den vorhandenen Sensoren des Gerätes arbeitet. So können die sogenannten *Inertial Measurement Units* (IMU), also Gyroskope und Beschleunigungssensoren, für die Bestimmung von Bewegungsabläufen der Position und Rotation verwendet werden. Das Problem dieser Art der Positionsverfolgung ist ein starker Drift, also ein kontinuierlich stärker werdender Fehler der Positionsbestimmung, der aufgrund von kleinen Berechnungsungenauigkeiten der Sensoren auftritt. Aufgrund dieses Drifts ist die ausschließliche Nutzung von IMU-Sensoren zur Positionsverfolgung für Augmented Reality Anwendungen nicht ausreichend. [28]

2.3.2 Simultaneous Localization and Mapping

Cadena et. al. [29] definieren *Simultaneous Localization and Mapping* (SLAM) als den Prozess des kontinuierlichen Erstellens einer Karte der Umgebung, während innerhalb dieser Karte die Bewegung und Rotation anhand der 6 Freiheitsgrade (6DoF)

getrackt werden. Dabei kann das Prinzip von SLAM auch mit Laserscannern oder anderen Geräten angewandt werden, in diesem Rahmen wird aber hauptsächlich die Umsetzung anhand der visuellen Methode betrachtet.

Moderne AR-Frameworks wie ARCore verwenden das SLAM-Verfahren, um visuelle Merkmalspunkte (feature points) im Sichtfeld der Kamera zu erkennen und daraus die Bewegung des Gerätes zu verfolgen. In Kombination mit SLAM werden außerdem die Messungen der inertialen Messeinheiten (IMU) des Gerätes verwendet, um die Veränderung der Position und Orientierung relativ zur echten Welt laufend bestimmen zu können. [30]

Die Nutzung dieser visuellen Positionsverfolgung ist laut Mendoza-Silva et. al [18] anderen Methoden sowohl in Genauigkeit und Kosten überlegen, da keine weiteren externen Geräte benötigt werden und Smartphones heutzutage performant genug für die benötigten Berechnungen sind. Die Genauigkeit von visueller Positionsverfolgung wird von Cadena et. al. [29] anhand eines sehr geringen Drifts von weniger als 0,5 % der gelaufenen Strecke aufgezeigt, der vor allem bei der ausschließlichen Nutzung von IMUs bedeutend höher ist.

2.3.3 Markerbasierte Ortung

Eine der ältesten und einfachsten Lösungen zur Lokalisierung mithilfe einer optischen Methode ist das Erkennen von physikalischen Markierungen im Bild der Kamera. Dabei beschreiben Sakpere et. al. [31] vor allem QR-Codes, Barcodes oder sogenannte Fiducials als geeignete Markierungen. Durch das Erkennen und Scannen dieser Markierungen können die dazugehörigen Daten geladen werden, die beispielsweise genaue Koordinaten, oder eine eindeutige ID zur Findung innerhalb eines definierten Gebäudeplanes enthalten.

In Kombination mit modernen Augmented Reality Frameworks kann sich ein Smartphone zusätzlich in Relation zu diesen erkannten Markierungen positionieren und durch 6DoF-Tracking die Position des Gerätes anschließend selbstständig weiter verfolgen.

2.3.4 Cloud Anchor

Die größte Einschränkung von SLAM-Tracking in Augmented Reality Anwendungen ist die rein relative Positionsverfolgung, die die Position nur in Abhängigkeit zur ursprünglichen Startposition verfolgen kann und dabei keine globale Referenz existiert.

Google bietet für ihr ARCore-Framework mit der *Cloud Anchor API* [2] eine Lösung für diese Einschränkung an, die es ermöglicht, die Punktwolken der direkten Umgebung anhand eines Cloud Anchors zu hosten und ein lokales Koordinatensystem

daran zu speichern. So kann durch Abgleichen der Umgebung mit der Kamera zu einem späteren Zeitpunkt ein vorher erstellter Cloud Anchor wieder gefunden werden und dessen relatives Koordinatensystem an der global selben Stelle im Raum wieder verwendet werden. Dadurch kann die hohe Genauigkeit des relativen SLAM-Trackings in Abhängigkeit zu einem wiederverwendbaren Referenzort der echten Welt verwendet werden, um eine genaue Lokalisierung zu ermöglichen.

Neben der ARCore Cloud Anchor API bieten auch andere Anbieter ähnliche Lösungen für die Lokalisierung durch Abgleichung von Punktwolken an, wie beispielsweise Microsoft mit ihren *Azure Spatial Anchor* [32], Vuforia mit ihren *Area Targets* [33] oder Neogoma mit ihrer *Stardust Visual Positioning SDK* [34]. Diese Systeme ermöglichen eine sehr ähnliche Option zur Speicherung und Auflösung von den Umgebungsdaten eines Ortes und können ebenfalls als Referenzpunkt für relatives SLAM-Tracking verwendet werden, um eine Lokalisierung zu ermöglichen, wobei beispielsweise mit der Stardust SDK auch größere 3D-Karten erstellt werden können.

Die Implementation dieser APIs weist eine durchaus erwähnenswerte Problematik des Datenschutzes auf, da die Umgebungsdaten des Kamerabildes für die Auflösung dauerhaft an einen externen Server gesendet und dort verarbeitet werden. Des Weiteren sind für alle erstellten Cloud Anchor die Punktwolken der direkten Umgebung auf diesen externen Servern gespeichert und die Verwendung dieser aufgrund der proprietären Architektur nicht einsehbar.

2.3.5 Visual Positioning System

Im Mai 2022 hat Google mit der *Geospatial API* und dessen *Visual Positioning System* (VPS) eine neue Lösung zur globalen Lokalisierung in Augmented Reality veröffentlicht [1], die ebenfalls nach einem ähnlichen Prinzip des Abgleichens von Punktwolken arbeitet, allerdings damit die globale Position und Ausrichtung eines Nutzers bestimmen kann. Dafür wurden aus den Bildern aller existierenden Google Street View Aufnahmen Punktwolken generiert, die mit den Kamerabildern einer Augmented Reality App mithilfe von neuronalen Netzen und Computervision-Algorithmen abgeglichen werden können und woraus innerhalb von Sekunden die genauen geografischen Koordinaten und die Ausrichtung eines Smartphones bestimmt werden kann, die im Gegensatz zu GPS sehr viel genauer ist. [35]

Seit 2019 war diese Technologie innerhalb von Google Maps als *Live View* Funktion verfügbar, die es dort bereits ermöglichte, die genaue Position und Ausrichtung des Smartphones durch einfaches Schwenken der Kamera zu bestimmen [36]. Durch die Veröffentlichung der neuen Geospatial API und dessen freier Verfügbarkeit zur

Nutzung innerhalb von eigenen Anwendungen werden dadurch allerdings neue Anwendungsfälle möglich, die mit GPS-Ortung nicht vorstellbar gewesen wären. [35]

Auch Apple bietet seit Ende 2020 für ihr AR-Framework ARKit ebenfalls eine ähnliche Lösung zur globalen Ortung des Mobilgerätes, das durch das Abgleichen des Kamerabildes mit von Apple gespeicherten Umgebungsdaten arbeitet. Allerdings ist dieses System nur in einer sehr eingeschränkten Anzahl von Orten verfügbar, die sich vor allem auf die größten amerikanischen Städte und wenige weitere internationale Ausnahmen beschränken. [37]

3 Analyse

3.1 Stand der Wissenschaft

In diesem Kapitel werden einige konkrete Implementationen aus der Literatur vorgestellt, bei denen die in Kapitel 2 beschriebenen Technologien innerhalb von Augmented Reality Anwendungen zur Ortung eingesetzt werden. Um im Folgenden die Anforderungen an die zu entwickelnde Anwendung aufstellen zu können, werden dabei primär Anwendungen mit Lösungen für die Lokalisierung innerhalb von Gebäuden betrachtet, aber auch eine relevante Anwendung für die Verwendung in Stadtgebieten vorgestellt.

3.1.1 Funkbasierte Ortung mit Augmented Reality

Einige Ansätze aus der Literatur benutzen funkbasierte Ortungsmethoden als Referenz für Augmented Reality Anwendungen. So beschreiben Huang et. al. [38] ein auf Augmented Reality basierendes Indoor-Navigationssystem, das die in Kapitel 2.2.2 beschriebenen BLE-Beacons als primäre Ortungsmethode verwendet. Diese BLE-Beacons werden an jeder Kreuzung, Treppe oder sonstigen festgelegten Wegpunkten im Gebäude platziert, anhand dessen durch Messung der Signalstärken (RSS) die Position des Mobilgerätes bestimmt werden kann. Zusätzlich werden Messungen der IMU-Sensoren zur Anzeige von AR-Wegmarkierungen für die Navigationskomponente verwendet. Laut den Autoren konnte mit dieser Methodik eine Genauigkeit der Positionsbestimmung zwischen 3 und 5 Metern erreicht werden, wobei dabei ein großer Arbeits- und Kostenaufwand durch die Platzierung der BLE-Beacons benötigt wurde.

Eine weitere funkbasierte Anwendung aus der Literatur wird von Ahmand et. al. [39] vorgestellt, bei der RSS Fingerprinting von WiFi-Signalen zur Lokalisierung verwendet wird. Für die Nutzung der Anwendung muss zunächst ein 2D-Grundriss des Gebäudes verfügbar sein, der von einem Nutzer mithilfe einer Augmented Reality Brille (Hololens) manuell an die Umgebung ausgerichtet werden muss und auf den ein Raster an Referenzpunkten projiziert wird. Das relative 6DoF-Tracking der Hololens wird verwendet, um für jedes Feld des Rasters einen WiFi-Fingerprint zu erstellen und der entsprechenden Position auf der Karte zuzuordnen. Anhand der so erstellten Datenbank an Messungen kann sich ein Nutzer anschließend durch die WiFi-Signale lokalisieren lassen und dadurch die entsprechende Position auf dem Raster des Gebäudeplanes erhalten. Laut den Autoren konnte das System eine Genauigkeit von unter 1,5 Metern bei der Lokalisierung erreichen, wobei insgesamt 174 Fingerprints entlang von vier je etwa 45 Meter langen Flurstücken erhoben wurden.

3.1.2 Building Information Modeling

Eine weitere Methode, um eine Referenz in relativer Indoor-Lokalisierung zu ermöglichen, wird von Herbers und König [40] vorgestellt, wofür existierende digitale Gebäudepläne und Informationen aus sogenannten *Building Information Models* (BIM) verwendet werden. Für die Lokalisierung werden laufend Punktwolken der Kamera erstellt und zu zweidimensionalen Plänen der Umgebung transformiert, um so mit den Gebäudeplänen des BIM abgeglichen werden zu können. So kann eine genaue Referenz innerhalb des existierenden Gebäudeplanes gefunden werden, während das AR SLAM-Tracking in Kombination mit IMU-Messungen zur weiteren relativen Bewegungsverfolgung verwendet wird. Die größte Schwierigkeit dieses Systems wurde als der initiale Findungsprozess einer Umgebung beschrieben, die einzigartig genug für eine erfolgreiche Übereinstimmung mit dem Gebäudeplan sein muss. So musste das Gerät im Durchschnitt etwa 15-20 Meter bewegt werden, bevor eine zuverlässige Position bestimmt werden konnte, anschließend konnte aber eine sehr hohe kontinuierliche Genauigkeit von 5 bis 30 cm erreicht werden.

Einige weitere Anwendungen mit Ansätzen des BIM-Matching, also der Nutzung von existierenden Gebäudeplänen zum Abgleichen mit den aktuellen Punktwolken als Referenz für AR-Indoor-Ortung, werden von Baek et. al. [41], Chen et. al [42] und Mahmood et. al.[43] vorgestellt.

3.1.3 Cloud Anchor Lokalisierung

Varelas et. al. [44] stellen ein AR-Indoor Ortungs- und Navigationssystem vor, das auf den in Kapitel 2.3.4 beschriebenen ARCore Cloud Anchorn als Grundlage an globaler Referenz basiert. Um die Anwendung in einer Umgebung nutzen zu können, muss zuvor ein Nutzer eine Karte erstellen, in dem er die gewünschte Strecke abläuft und dabei regelmäßig Cloud Anchor platziert. Die Autoren bezeichnen einen 2 Meter Abstand zwischen den Ankern als optimal, während zusätzlich *points of interest* als mögliche Wegpunkte für die spätere Navigation hinzugefügt werden sollen.

Sobald so eine vollständige Karte erstellt wurde, kann ein Nutzer durch Scannen der Umgebung mit der Kamera und Abgleichung mit der Cloud Anchor API mögliche Übereinstimmungen finden und so einen der Cloud Anchor der Karte auflösen. Bei Erfolg kann so dessen genaue Position und die gesamte restliche gespeicherte Karte und dessen Ankerpositionen in Augmented Reality abgebildet und dem Nutzer angezeigt werden. Die Position des Nutzers relativ zu den Cloud Anchorn wird mithilfe von Trilateration bestimmt, also der Berechnung anhand von Distanzen zu drei Ankern im Sichtfeld. Mit der bestimmten Position und der existierenden Karte bietet die Anwendung des Weiteren die Möglichkeit, sich zu einem der erstellten Wegpunkten navigieren zu lassen, wofür eine Strecke mithilfe eines A* Algorithmus entlang der

Ankerpunkte bestimmt werden kann.

In den Untersuchungen der Arbeit wurde beschrieben, dass das Auflösen eines Cloud Anchors anhand von den umliegenden Punktwolken nur in einem maximalen Abstand von 10 Metern funktioniert und dass bei einer kurzen Entfernung von weniger als 3 Metern eine deutlich höhere Genauigkeit beobachtet werden konnte. Insgesamt wurde die Verwendung der Cloud Anchor zur globalen Referenz als eine sehr genaue Option beschrieben, wobei die größten Einschränkungen der Anwendung als der zeitaufwändige Prozess der Erstellung einer Karte und der ebenfalls häufig sehr lange dauernde Teil des Findens eines ersten Cloud Anchors bei der Lokalisierung genannt wurden.

3.1.4 Visuelle Lokalisierung im Stadtraum

Platinsky et. al. [45] stellen eine eigene Lösung für mobile globale Lokalisierung innerhalb von urbanen Gebieten vor, die auf einer ähnlichen Technologie zur Positionsbestimmung wie das in Kapitel 2.3.5 beschriebene Visual Positioning System der Geospatial API basiert. Dafür mussten zunächst genaue 3D-Karten der gesamten Stadt erstellt werden, wofür mit je 8 Kameras und damit einer 360° Sicht ausgestattete Autos verwendet wurden, die über einen längeren Zeitraum, alle Straßen der Stadt abgefahren sind. Um eine genaue Karte zu erreichen, musste jeder Straßenzug laut den Autoren mehrfach abgefahren werden, sodass die Chance einer späteren erfolgreichen Lokalisierung von etwa 70 % nach der ersten Fahrt zu 97,3 % durch Kombination der Daten von insgesamt 7 Fahrten angestiegen ist.

Die vorgestellte zugehörige AR-Anwendung kann sich anhand dieser so erstellten 3D-Karten anschließend lokalisieren, in dem zunächst die eigenen Punktwolken der Sicht der Kamera auf dem Gerät erstellt werden und anschließend über eine Internetverbindung an den zugehörigen Server gesendet werden. Durch Abgleichen der Punktwolken des Gerätes mit der 3D-Karte der Stadt wird auf dem Server eine Übereinstimmung gesucht, mit der so in 73 % aller Fälle eine korrekte Lokalisierung ermöglicht wurde, wobei die erreichte Genauigkeit davon im Durchschnitt bei 42 cm lag. Insgesamt konnte in der vorgestellten Arbeit durch einen intensiven Mapping-Prozess über 3 Monate und mit insgesamt mehr als 1000 gefahrenen Meilen eine 3D-Karte für die gesamte Stadt San Francisco erstellt werden, mit der entlang der Straßen im Stadtbereich eine deutlich genauere Lokalisierung möglich wurde, als es mit GPS-Ortung im Stadtraum möglich wäre. Eine der größten Einschränkungen dieses Systems sind neben dem intensiven Mapping-Prozess auch die benötigten Rechenleistungen, um die Karte initial zu erstellen, aber auch um schnelle Reaktionszeiten bei einer Lokalisierung zu ermöglichen.

Das Visual Positioning System der Geospatial API bietet eine sehr ähnliche Mög-

lichkeit der globalen Lokalisierung innerhalb des Stadtraumes und hat durch die Verwendung der Street View Daten eine bedeutend größere Abdeckung an Orten. Allerdings sind aufgrund der so kurzen Zeit seit der Veröffentlichung der Geospatial API zum aktuellen Zeitpunkt noch keine sinnvollen Anwendungen in der Literatur zu finden, die in diesem Rahmen vorgestellt werden könnten. Lediglich Julien Mercier et. al [46] beschreiben die vielversprechenden Möglichkeiten von VPS für ihre ortsbasierte AR-Anwendung *BiodivAR*, allerdings handelt es sich dabei lediglich über eine Erwähnung im Ausblick der Arbeit und es wurde noch keine Implementation vorgenommen.

3.2 Anforderungsanalyse

Der aktuelle Stand der Wissenschaft beinhaltet einige verschiedene Anwendungen für AR-Lokalisierung innerhalb von Gebäuden. Dabei bietet das relative SLAM-Tracking der AR-Frameworks eine sehr genaue Möglichkeit für 6DoF-Tracking, allerdings ist eine Kombination mit einer globalen Referenz für die Nutzung zur Lokalisierung nötig. Für diese Referenz kommen herkömmliche funkbasierte Ortungsmethoden in Frage, dessen Genauigkeiten allerdings nicht sonderlich hoch sind und wofür häufig ein großer initialer Aufwand und Kostenfaktor notwendig ist.

Als alternative Referenz verwenden einige Anwendungen aus der Literatur die digital modellierten Bauwerksdaten (BIM), die bei der Architektur von modernen Gebäuden in der Regel erstellt werden. Durch Abgleichen der Punktwolken der Kamera kann so eine sehr hohe Genauigkeit für die initiale Lokalisierung als Referenz erreicht werden, allerdings ist für diese Form der Lokalisierung die Verfügbarkeit von digitalen Gebäudemodellen notwendig, die vor allem bei älteren Gebäuden oft nicht existieren. Für diese Gebäude oder für den Stadtraum können ähnliche Modelle in einem sehr aufwändigen und zeitintensiven Prozess mit spezieller Hardware selber erstellt werden. Alternativ bieten Bibliotheken wie die ARCore Cloud Anchor API eine weitere Möglichkeit, um eine globale Referenz für das relative 6DoF-Tracking der SLAM-Algorithmen zu ermöglichen, wobei nur die Kamera eines modernen Smartphones ausreichend ist, um mit überschaubarem Zeitaufwand eine Karte zur Lokalisierung zu erstellen.

Alle diese Methoden aus der Literatur bieten keine Möglichkeit, um globale Koordinaten bei der Indoor-Lokalisierung zu bestimmen, da nur in Relation zu den jeweiligen Referenzsystemen geortet wird. Für diese lokalen Referenzsysteme müsste eine globale Referenz bestimmt werden, wofür die Nutzung des Visual Positioning System der Geospatial API geeignet wäre, womit die genauen globalen Koordinaten auch in engen Straßen im Stadtraum bestimmt werden können.

Im Folgenden werden die Anforderungen an die zu implementierende Anwendung aufgestellt, anhand dessen der grundlegende Aufbau des Systems entworfen werden soll. Diese Anforderungen werden zunächst anhand von Muss- und Soll-Kriterien beschrieben, während im Anschluss Abgrenzungen zu den Ansätzen aus der Literatur gezogen und voraussichtliche Limitierungen aufgezeigt werden.

3.2.1 Muss-Kriterien

Im Rahmen dieser Arbeit soll eine Augmented Reality Anwendung entwickelt werden, mit der eine genaue Lokalisierung des Benutzers innerhalb von Gebäuden ermöglicht wird. Dabei muss die Anwendung zunächst einen einfachen Prozess zur Erstellung von Gebäudeplänen durch Ablaufen eines Gebäudeabschnitts bieten, bei dem Cloud Anchor als globale Referenz zur Lokalisierung platziert und an den mithilfe der Geospatial API die genauen geografischen Koordinaten verknüpft werden.

Die Lokalisierung anhand eines erstellten Gebäudeplanes muss an jedem Ort entlang der gemappten Strecke möglich sein, indem ein beliebiger Cloud Anchor aufgelöst werden kann. Anschließend muss die relative Position und die genauen Koordinaten des Nutzers bestimmt und mithilfe des relativen SLAM-Trackings weiter aktualisiert werden können. Entlang des gesamten erstellten Gebäudeplanes müssen laufend neue Cloud Anchor aufgelöst werden können, um dem relativen SLAM-Drift entgegenzuwirken und um potenzielle Ungenauigkeiten der Cloud Anchor zu korrigieren.

Da Ziel der Arbeit auch die detaillierte Untersuchung der Genauigkeit und Zuverlässigkeit des zu entwickelnden Ortungssystems ist, muss die Anwendung Möglichkeiten zum Datenexport während der einzelnen Prozesse haben. So sollen bei der Erstellung und der Lokalisierung jeweils alle relevanten Informationen in einer Datei gespeichert werden, um später für die Evaluierung visualisiert werden zu können.

3.2.2 Soll-Kriterien

Um die Nutzung der Anwendung zur Lokalisierung für möglichst viele Benutzer zu ermöglichen, soll eine zentrale Datenhaltung für die Gebäudepläne implementiert werden, auf die jeder Nutzer Zugriff zum Hinzufügen und Lesen hat und lediglich das Bearbeiten und Löschen auf den erstellenden Nutzer limitiert ist.

Des Weiteren sollte die Anwendung Möglichkeiten zur Visualisierung der einzelnen Gebäudepläne haben, damit ein Nutzer besser einschätzen kann, ob ein Plan für den aktuellen Ort geeignet ist und in welchen Bereichen des Gebäudes diese Lokalisierung verfügbar ist.

Während der Lokalisierung sollte einem Nutzer außerdem die bestimmte Position auf einer (Indoor-) Karte angezeigt werden, um sich anhand dessen im Gebäude einordnen zu können und um zur Wegfindung verwendet werden zu können.

Das Erstellen eines neuen Gebäudeplanes sollte möglichst simpel und schnell möglich sein, damit ein beliebiger Nutzer einfach ein neues Gebäude hinzufügen kann, ohne dass Vorwissen oder Übung benötigt sind.

Als konkrete Anwendung der Lokalisierung soll eine prototypische Navigation implementiert werden, die während der aktiven Ortung dem Nutzer die Möglichkeit gibt, sich zu einem beliebigen Cloud Anchor als Wegpunkt navigieren zu lassen. Dabei solle die bestimmte Route sowohl als Markierung des Weges in AR, als auch auf der 2D-Karte dargestellt werden.

3.2.3 Abrenzungskriterien

Im Gegensatz zu den vorgestellten Anwendungen aus der Literatur soll mit dem zu entwickelnden System eine Indoor-Lokalisierung ermöglicht werden, die globale geografische Koordinaten für die Verwendung mit 2D-Karten bestimmen kann. Des Weiteren soll keine Abhängigkeit zu externen Signalen, teurer Hardware oder bereits existierenden digitalen Modellen bestehen und ein Gebäude soll von jedem Nutzer mithilfe der zu entwickelnden App einfach ausgemappt werden können. Es soll eine initiale Genauigkeit der Lokalisierung erreicht werden, die deutlich über die von funkbasierten Methoden hinausgeht und anschließend die Bewegung des Smartphones mithilfe des relativen SLAM-Trackings zuverlässig weiter bestimmt werden.

3.2.4 Limitationen

Aufgrund der Abhängigkeit zu den ARCore Cloud Anchor und Geospatial APIs ist für die Verwendung der Anwendung zu jedem Zeitpunkt eine aktive Internetverbindung notwendig, was vor allem für die Nutzung in U-Bahnhöfen oder anderen großen Gebäuden durchaus zu Einschränkungen führen kann. Des Weiteren ist für die Erstellung eines Gebäudeplanes die Verfügbarkeit von Street View Daten notwendig, da nur so mit der Geospatial API der initiale Referenzpunkt genau lokalisiert werden kann und ansonsten die Koordinaten nur mit ungenauer GPS-Ortung bestimmt werden könnten. Diese Abhängigkeit zu Street View ist vor allem für die Nutzung in Deutschland eine größere Einschränkung, da nur für die größeren Städte diese Daten überhaupt existieren und zusätzlich dazu sehr veraltet sind.

Durch die Ortung anhand von Punktwolken der Umgebung innerhalb von Gebäuden ist das zu entwickelnde System von baulichen Veränderungen betroffen und

es ist zu erwarten, dass Gebäudepläne nach größeren Änderungen zu aktualisieren sind. Vor allem für die draußen platzierten Cloud Anchor können auch unterschiedliche Licht- und Wetterverhältnisse die erfolgreiche Auflösung behindern. Des Weiteren sind ARCore Cloud Anchor auf eine maximale Lebensdauer von einem Jahr limitiert [47], weshalb nach Ablauf dieser Zeit ein Gebäudeplan ebenfalls aktualisiert werden muss.

Neben den beschriebenen Limitationen profitiert die Anwendung sehr von der Verfügbarkeit von Indoor-Karten, da diese sowohl für die Erstellung als auch die Lokalisierung als Hilfsmittel verwendet werden sollen. Zwar kann sich die Anwendung auch problemlos ohne Indoor-Karten lokalisieren, allerdings leidet dadurch die Übersichtlichkeit der Gebäudepläne und bei der Lokalisierung fehlt die sonst übersichtliche Darstellung der bestimmten Koordinaten auf der Karte des Gebäudes. Da diese Indoor-Karten von den Eigentümern der einzelnen Gebäude selbstständig an Google Maps gesendet werden müssen, ist die Verfügbarkeit sehr unterschiedlich und vor allem ortsabhängig, sollte aber zunehmend immer mehr werden.

4 Systementwurf

Im Rahmen dieses Kapitels wird das zu entwickelnde System anhand der im vorherigen Kapitel formulierten Anforderungen entworfen und näher vorgestellt. Es soll zunächst die geplante Architektur der Anwendung erläutert werden und dabei auf verschiedene mögliche Optionen an Technologien eingegangen werden, aus denen die zu verwendenden Lösungen gewählt werden.

4.1 App Übersicht

Das zu entwickelnde System soll primär aus einer Android-Anwendung bestehen, die selbst in mehrere Komponenten zu unterteilen ist.

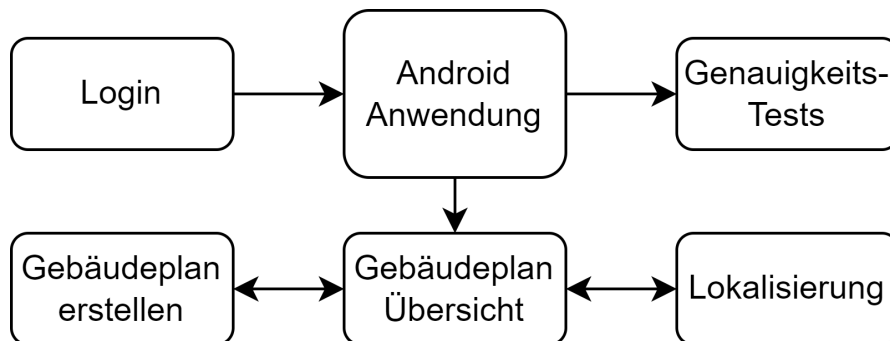


Abbildung 4: Übersicht der zu implementierenden Anwendung

In Abbildung 4 sind die einzelnen Komponenten der Android-Anwendung dargestellt, die benötigt werden, um die festgelegten Anforderungen zu erfüllen. Dabei steht zu Beginn eine Login-Komponente, die dafür sorgen soll, dass alle Nutzer der Anwendung authentisiert sind und im System eine klare und eindeutige Nutzer-ID haben.

Der Kern und gleichzeitig das Hauptmenü der eigentlichen Anwendung soll anschließend eine Übersicht aller verfügbaren Gebäudepläne sein, die mit Möglichkeiten zur Suche und Sortierung der Pläne, wie auch mit der Option der Visualisierung eines einzelnen Gebäudeplanes ausgestattet sein soll.

Falls ein Gebäudeplan für den gewünschten Abschnitt eines Gebäudes bereits verfügbar ist, soll ein Nutzer von dieser Übersicht einfach die Lokalisierung starten können, wodurch er auf eine der Augmented Reality Komponenten weitergeleitet wird. Mit dieser AR-Lokalisierung soll nun einfach die aktuelle Position des Nutzers bestimmt werden können und auf einer eingeblendeten 2D-Karte dargestellt werden.

Für die Lokalisierung muss natürlich zunächst auch die Möglichkeit bestehen, einen neuen Gebäudeplan zu erstellen. Dies soll ebenfalls aus der Plan-Übersicht gestartet werden können und eine weitere Augmented Reality Komponente öffnen, bei der der Nutzer durch die Schritte der Erstellung eines Planes geleitet wird.

Aufgrund der aufgestellten Anforderung, dass die Anwendung und die zugrundeliegenden Technologien auf ihre Genauigkeit untersucht und verglichen werden sollen, soll eine weitere AR-Komponente implementiert werden, mit der Genauigkeitsuntersuchungen einfach durchgeführt werden können.

4.2 Datenhaltung

Als zentraler Bestandteil der zu implementierenden Anwendung sind die Gebäudepläne zu sehen, die einen beliebigen Abschnitt der echten Welt als virtuelles Netz an Informationen und Koordinaten betrachten. Es wird zunächst der geplante Aufbau des Datenmodelles vorgestellt und im Anschluss die zu verwendende Datenhaltung festgelegt.

4.2.1 Datenmodell

Als Datenmodell eines Gebäudeplanes soll eine Klasse vom Typ `FloorPlan` implementiert werden, die zunächst einen Namen, einen weiteren Beschreibungs-Text und eine eindeutige ID des erstellenden Nutzers als Attribute hat, um damit gesucht und einfach identifiziert werden zu können. In Abbildung 5 ist anhand eines UML-Diagrammes der geplante Aufbau eines Gebäudeplanes zu sehen, der neben den grundlegenden Informationen vor allem auch die zur Lokalisierung benötigten Daten beinhaltet. Jeder Gebäudeplan hat zunächst einen Cloud Anchor als initialen Startpunkt, der bei der Erstellung als erstes festgelegt wird und dessen genaue globale Position und Orientierung mithilfe der Geospatial API bestimmt wird. Anhand dieses initialen Cloud Anchors soll die abgelaufene Strecke anhand einer Liste an `MappingPoints`, also relativen 3D-Positionen gespeichert werden, wobei zur Lokalisierung

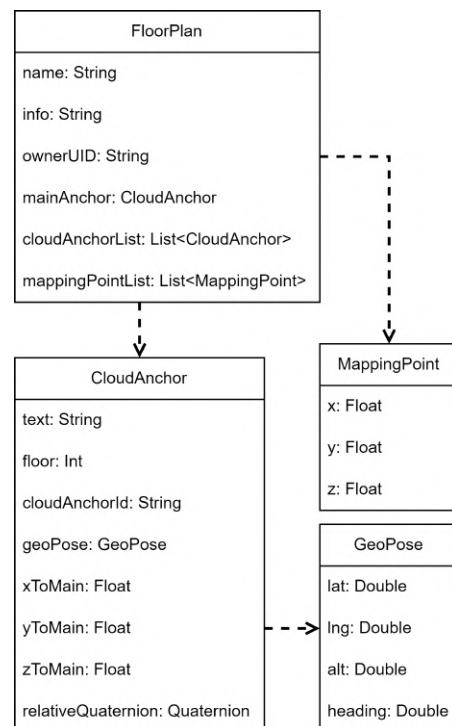


Abbildung 5: UML-Diagramm: Aufbau Gebäudeplan

vor allem die Liste an weiteren platzierten Cloud Anchoren verwendet werden sollen. Diese weiteren Cloud Anchor haben jeweils einen kurzen Beschreibungstext und die Nummer der Etage als Attribute, wobei zusätzlich sowohl die relativen Koordinaten in Relation zum initialen Cloud Anchor des Planes, als auch die daraus errechneten globalen Koordinaten gespeichert werden sollen. Aus dieser Liste an Cloud Anchoren mit relativen Positionen spannt sich so ein Netz des ausgemessenen Bereiches auf, innerhalb dessen die Lokalisierung stattfinden kann.

4.2.2 Firebase

Damit mehrere Nutzer die Anwendung zur selben Zeit nutzen und dabei auf Gebäudepläne von anderen Nutzern zugreifen können, soll eine zentrale Datenhaltung für die Pläne implementiert werden. Dabei soll die Lösung so gestaltet werden, dass die Anwendung die Gebäudepläne in einem lokalen Speicher zwischenspeichert und lediglich nach dem Hinzufügen oder bei Aktualisierung der Daten auf der Serverseite eine Internetverbindung notwendig ist. Jeder Nutzer der Anwendung soll Lesezugriff auf alle existierenden Gebäudepläne haben, damit die Anwendung an möglichst vielen Orten verwendet werden kann, allerdings soll nur der jeweilige erstellende Nutzer seine Pläne bearbeiten oder löschen dürfen.

Um die serverseitige Datenhaltung für die Gebäudepläne zu gewährleisten, gibt es eine Vielzahl unterschiedlicher Implementierungsmöglichkeiten, die von selbst gehosteten Rest-Servern bis zu verschiedenen Cloud-Lösungen reichen. Um einen schnellen und jederzeit verfügbaren Server zu haben und dabei diverse Sicherheitsregeln einfach implementieren zu können, soll die *Firebase Cloud Firestore* [48] Plattform von Google verwendet werden, die eine auf Dokumenten basierte NoSQL Datenbank anbietet und im Rahmen von kleinen Projekten kostenfrei zu verwenden, aber trotzdem auch für große Nutzerzahlen einfach zu skalieren ist. Dabei ist die Integration von Firebase in Android-Anwendungen sehr einfach und gut dokumentiert und unterstützt dabei eine native Cache-Funktion für Offline-Funktionalität.

Um den Zugriff auf die einzelnen gespeicherten Daten innerhalb von Firebase einzuschränken, werden einfache Sicherheitsregeln verwendet, die anhand der verschiedenen Zugriffsarten auf die Daten festgelegt werden [49]. Da jeder Nutzer der Anwendung alle Gebäudepläne sehen und selber neue hinzufügen können soll, soll die Einschränkung zum Lesen und Schreiben lediglich auf der erfolgreichen Authentisierung beruhen, die im nächsten Abschnitt näher beschrieben wird. Daraus ergibt sich die folgende einfache Regel:

```
allow read, create: if request.auth != null;
```

Um einen existierenden Gebäudeplan verändern oder löschen zu können, muss

dieser von demselben Nutzer erstellt worden sein, wofür das Feld der `ownerUID` verwendet wird, welches der eindeutigen ID der Firebase Authentication entspricht und wofür die folgende Regel verwendet wird:

```
allow update, delete: if request.auth.uid == resource.data.ownerUID;
```

4.2.3 Firebase Authentication

Um die zu implementierende Anwendung verwenden zu können, soll ein Nutzer sich zunächst einloggen, damit eine eindeutige ID bestimmt werden kann, anhand dessen der Eigentümer eines Gebäudeplanes festgelegt werden kann. Zum Login soll die *Google One Tap-Anmeldung* [50] verwendet werden, mit der ein neuer Nutzer lediglich seinen Google-Account auswählen muss, der zur initialen Registrierung und Authentisierung verwendet wird. *Firebase Authentication* [51] bietet eine große Vielzahl an verschiedenen Sign-In Providern an, allerdings sollte die Verwendung des Google Accounts bei der zu implementierenden Android-Anwendung vollständig ausreichen und so kann dafür gesorgt werden, dass die Anmeldung nur bei der ersten Nutzung der App getätigt werden muss und in folgenden Fällen einfach eine automatische Anmeldung stattfindet. Innerhalb von Firebase wird für einen so erstellten Account die entsprechende E-Mail-Adresse und die zugehörige UID gespeichert, wodurch Nutzer einfach identifiziert werden können, ohne dass sicherheitsrelevante Daten gespeichert werden müssen, da die Anmeldung weiterhin direkt über Google stattfindet.

4.3 Augmented Reality Komponenten

Mit dem Erstellen eines Gebäudeplanes und dem Lokalisieren benötigen die beiden Kernkomponenten der Anwendung Funktionen eines Augmented Reality Frameworks. Wie in Kapitel 2.3 bereits beschrieben, gibt es eine Vielzahl verschiedener Frameworks zur Entwicklung mit Augmented Reality. Da es sich bei der zu entwickelnden Anwendung um eine native Android-App handeln soll, bietet sich an dieser Stelle die Verwendung des ARCore Frameworks an, das eine native Integration für Android-Anwendungen hat und ebenfalls wie das Betriebssystem selbst auch von Google entwickelt wird. Ein weiterer wichtiger Grund für die Nutzung von ARCore ist die Verwendung der für die Anwendung benötigten ARCore Geospatial API für die genaue initiale globale Lokalisierung eines Ankerpunktes, dessen Nutzung mit vielen anderen Frameworks nicht möglich ist.

4.3.1 ARCore Render-Engine

Um Anwendungen mit ARCore zu implementieren und vor allem um 3D-Inhalte darstellen zu können, ist eine weitere Ebene der Entwicklung notwendig. So wird in der Dokumentation des ARCore Frameworks die Nutzung von OpenGL [52] vorgeschlagen, um selbständig eine Implementierung zur Darstellung von 3D-Inhalten vorzunehmen.

Bis Anfang 2020 hat Google mit der Sceneform SDK ein ergänzendes Framework zur einfachen Darstellung von 3D-Inhalten ohne die Notwendigkeit von diversem Boilerplate-OpenGL Code angeboten, was im Rahmen einer Code-Veröffentlichung seit dem nicht weiter gepflegt und empfohlen wird. [53]

Im Rahmen eines Open-Source Projektes wurde die Sceneform SDK zunächst von Thomas Gorisse weiterentwickelt und aktuell gehalten [54]. Zum Ende 2021 wurde vom gleichen Entwickler das neue Framework SceneView [55] zur Erweiterung von ARCore vorgestellt, was auf die Programmiersprache Kotlin umgestellt wurde und zur Darstellung von Inhalten die von Google entwickelte Rendering Engine Filament [56] verwendet.

Das SceneView Framework soll für die Implementation der Anwendung verwendet werden, da es eine einfach nutzbare, aber mächtige Rendering-Engine für die AR-Entwicklung bietet und dabei ohne viel Boilerplate-Code verwendet werden kann, der bei der Nutzung von nativem OpenGL benötigt würde.

4.3.2 ARCore APIs

Die Genauigkeit der initialen Positionsbestimmung des ersten Cloud Anchors eines Gebäudeplanes ist von enormer Wichtigkeit, weil die Koordinaten des gesamten restlichen Planes in Relation zu dieser bestimmt werden. Wie in Kapitel 2.1.1 beschrieben, ist die Ortung mithilfe von GPS-basierten Systemen vor allem in urbanen Regionen nicht sehr genau und zuverlässig möglich, weshalb für die zu implementierende Anwendung die in Kapitel 2.3.5 vorgestellte ARCore Geospatial API verwendet werden soll. Die Verwendung dieser API ermöglicht eine etwa metergenaue Positionsbestimmung, solange Street View Daten der direkten Umgebung verfügbar sind, weshalb der Startpunkt eines jeden Gebäudeplanes draußen und in der Nähe einer Straße platziert werden sollte.

Für die Verknüpfung der relativen Koordinaten eines Gebäudeplanes mit wiederfindbaren Positionen muss eine weitere API verwendet werden. Hier würden alle der in Kapitel 2.3.4 beschriebenen APIs infrage kommen, um eine Position der echten Welt in Augmented Reality genau wiederfinden zu können.

Da die native Funktionalität mit Android, ARCore und der Geospatial API vor allem für die Cloud Anchor API gegeben ist und die Funktionalität dessen auch bereits in vorherigen Projekten sichergestellt werden konnte, soll in dieser Anwendung die ARCore Cloud Anchor API verwendet werden. Prinzipiell wäre eine Verwendung von etwa den Azure Spatial Anchors durchaus auch vorstellbar, allerdings wurde die technische Machbarkeit und die Genauigkeit für das geplante System nicht näher untersucht, könnte aber in einem folgenden Projekt sinnvoll mit den Untersuchungen aus diesem Projekt verglichen werden.

4.4 System-Architektur

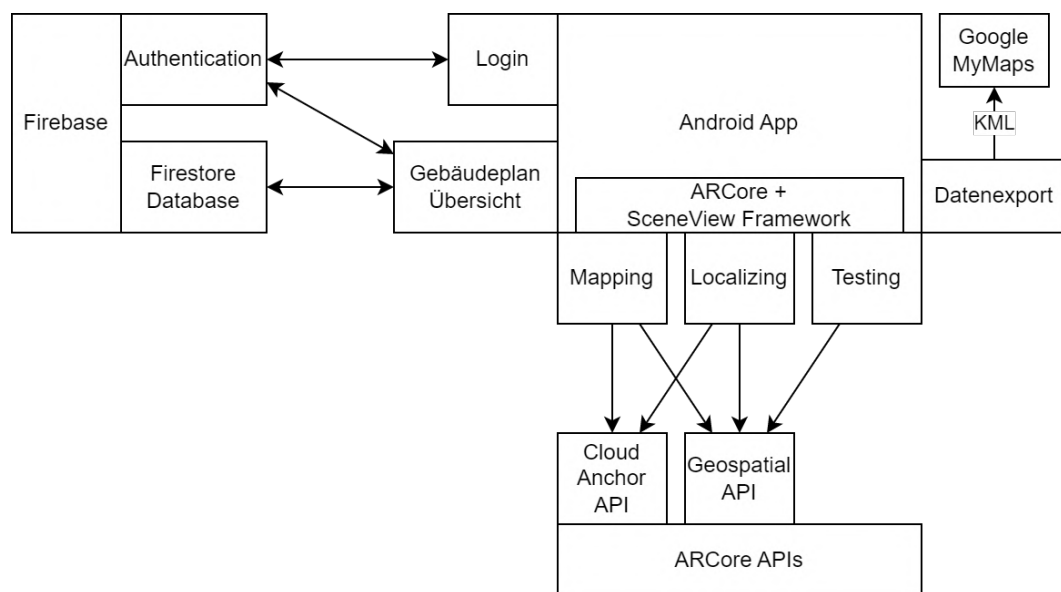


Abbildung 6: Übersicht der entworfenen System-Architektur

In der Abbildung 6 sind alle ausgewählten Komponenten und Systeme der zu implementierenden Anwendung abgebildet und die Abhängigkeiten und Kommunikationswege der einzelnen Teile eingezeichnet. Dabei besteht das System aus 3 grundlegenden Teilen, wobei der Hauptteil aus der Android-App mit ihren verschiedenen Funktionen und Komponenten besteht, wobei zusätzlich ein Datenexport für die spätere Visualisierung der Untersuchungen in Google MyMaps implementiert werden soll. Die anderen beiden Teile bestehen aus den verwendeten Cloud Services, wobei auf der einen Seite der Firebase Service mit der Datenhaltung und Authentisierung steht, während auf der anderen Seite die Google Cloud Services mit der Geospatial und Cloud Anchor API stehen, die von den AR-Komponenten direkt verwendet werden.

5 Implementierung

5.1 Übersicht

In diesem Kapitel wird die Umsetzung des Systementwurfes und die im Rahmen dieser Arbeit entwickelte Anwendung vorgestellt. Dies wird anhand der in Kapitel 4.1 festgelegten Komponenten der Anwendung strukturiert und mithilfe von Screenshots der verschiedenen Fragments der Android-App visualisiert, wobei Fragments in Android Anwendungen als Container für die einzelnen unterschiedlichen Bildschirm-inhalte fungieren.

Um eine bessere Vorstellung von den beiden primären Prozessen der Anwendung erhalten zu können, wurden zwei Bildschirmaufnahmen erstellt, die als Videos verlinkt sind:

Im ersten Video¹ wird der Prozess des Erstellens eines neuen Gebäudeplanes gezeigt, welcher an der TU Berlin durchgeführt wurde und dem in Abbildung 8b dargestellten Plan entspricht.

Das zweite Video² zeigt den Prozess der Lokalisierung nach diesem Gebäudeplan, der an einem Punkt innerhalb des Gebäudes startet.

Konkrete Beschreibungen der Funktionalität der in den beiden Videos gezeigten Prozesse werden in den folgenden Kapiteln 5.4 und 5.5 näher vorgestellt.

5.2 Login-Fragment

Beim ersten Start der Anwendung wird dem Nutzer zunächst ein Fragment für die initiale Registrierung beziehungsweise dem Login gezeigt, was in Abbildung 7 zu sehen ist. Dieses Fragment ist sehr simpel gehalten

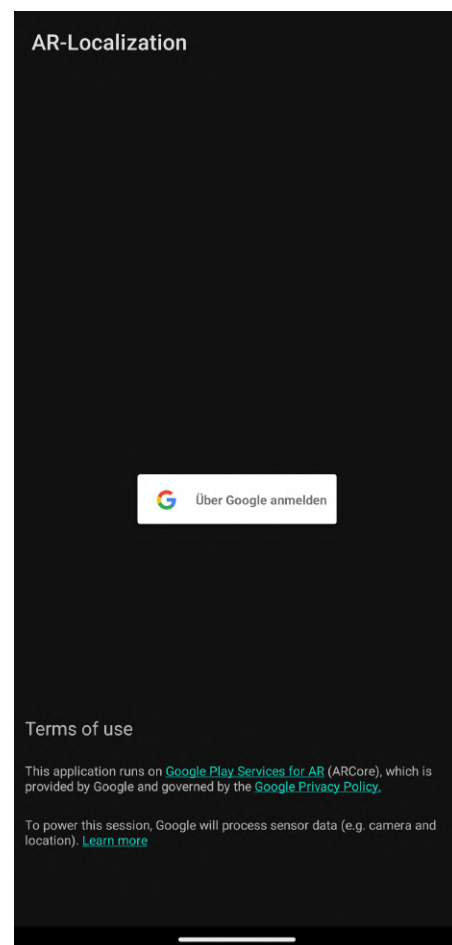


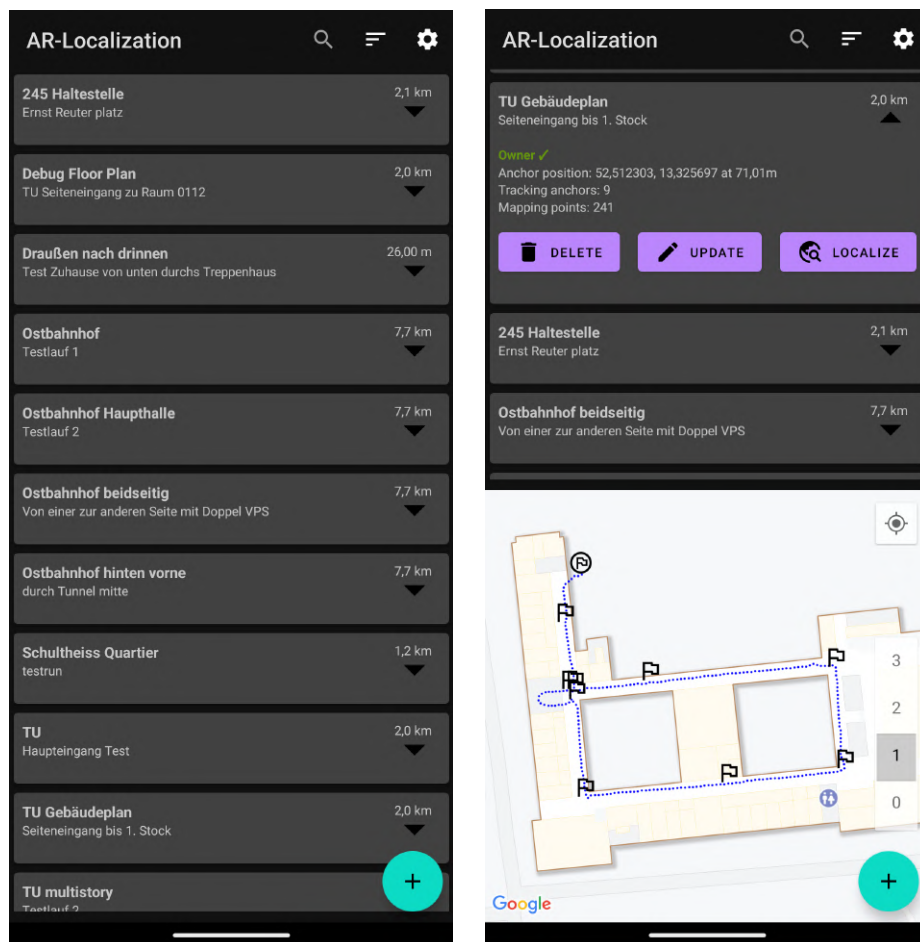
Abbildung 7: Fragment: Login

¹Demo Erstellung: https://www.youtube.com/watch?v=NUa_R3DZ12U

²Demo Lokalisierung: <https://www.youtube.com/watch?v=5aMPy0TZhQ>

und beinhaltet neben den Nutzungsbedingungen von ARCore und dessen verwendeter APIs nur einen Button zum Anmelden mit dem Google Account, der mit der *Google One Tap-Anmeldung* [50] implementiert wurde. Nach der ersten Anmeldung wird dieses Fragment bei späteren Verwendungen übersprungen, solange eine erfolgreiche automatische Authentisierung mit dem Google Account und Firebase stattfinden konnte.

5.3 Gebäudeplan Übersicht



(a) Übersicht der Gebäudepläne

(b) Übersicht der Gebäudepläne mit ausgewähltem Plan

Abbildung 8: Fragment: Übersicht der verfügbaren Gebäudepläne

Der eigentliche Hauptbildschirm der Anwendung ist das Fragment zur Übersicht aller verfügbaren Gebäudepläne. Dies ist zunächst sehr simpel gehalten und zeigt alle

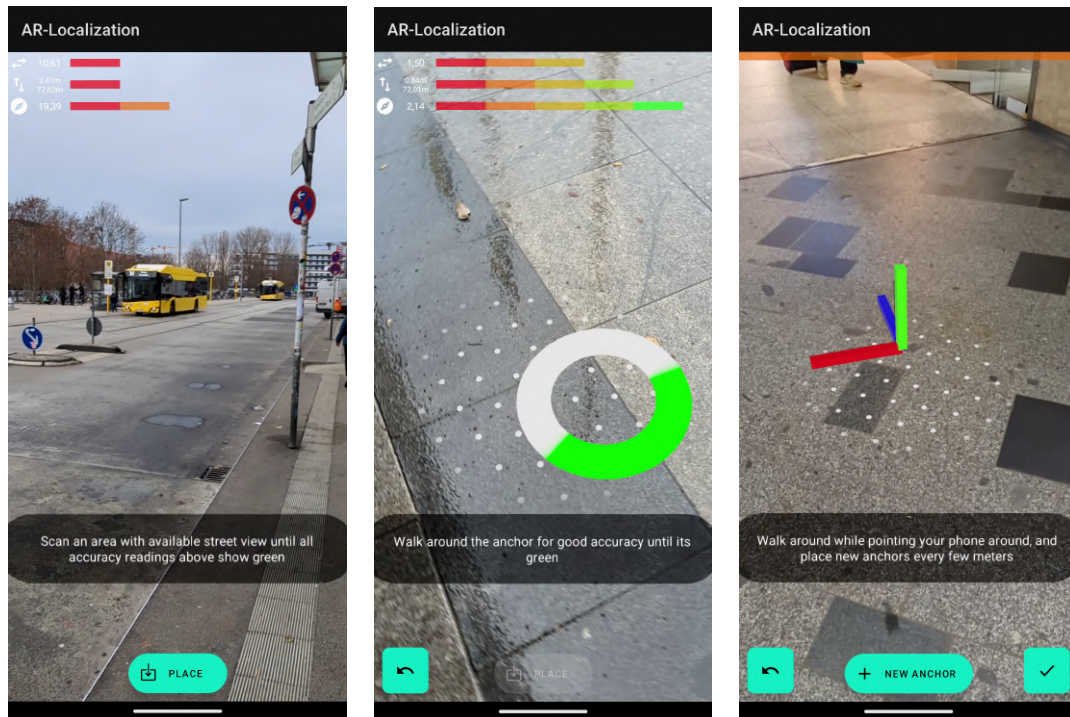
Gebäudepläne in einer Liste an, wobei jeweils der Name, die Beschreibung und die aktuelle GPS-Entfernung angezeigt wird. Wie in Abbildung 8a zu sehen ist, gibt es am oberen Bildschirmrand einen Button zur Suche nach konkreten Namen oder Beschreibungen sowie die Möglichkeit, die Sortierung der Liste zwischen dem Namen und der jeweiligen aktuellen Entfernung umzustellen. Über den Einstellungs-Button kann außerdem der Datenexport ausgeschaltet oder das Fragment für die Genauigkeitsuntersuchungen gestartet werden, während der + Button in der unteren rechten Ecke das Fragment zur Erstellung eines neuen Gebäudeplanes startet.

Um eine bessere Vorstellung von einem konkreten Gebäudeplan erhalten zu können, kann ein Element der Liste durch Tippen aufgeklappt werden, woraufhin der Bildschirm auf der Hälfte geteilt und eine Google Maps Karte eingeblendet wird, die bei Verfügbarkeit die entsprechende Indoor-Karte anzeigt. Auf dieser Karte wird der gesamte Gebäudeplan und die Positionen der zugehörigen Cloud Anchor dargestellt, anhand dessen ein Nutzer das verfügbare Gebiet und die gemappte Strecke besser nachvollziehen kann. Wie in Abbildung 8b zu sehen ist, werden außerdem weitere Informationen wie die Koordinaten des initialen Ankers und die Anzahl der zugehörigen Elemente sowie ein Button zum Start der Lokalisierung angezeigt, mit dem ein weiteres entsprechendes AR-Fragment gestartet wird. Falls ein ausgewählter Gebäudeplan von dem aktuellen Nutzer erstellt wurde, werden an dieser Stelle ebenfalls auch Buttons zum Bearbeiten und Löschen angezeigt.

5.4 Erstellung eines Gebäudeplanes

Wenn ein Nutzer auf den Button zum Erstellen eines neuen Gebäudeplanes tippt, wird zunächst ein kurzer Dialog geöffnet, in dem der Name und die weitere Beschreibung des Planes eingegeben werden soll. Anschließend wird das eigentliche Fragment zur Erstellung geöffnet, dessen anfänglicher Zustand in Abbildung 9a zu sehen ist. Zunächst muss ein Nutzer die genaue globale Position mithilfe der Geospatial API bestimmen lassen, indem das Smartphone in Richtung einer Straße mit Street View gehalten wird, wobei die im oberen Bereich dargestellten drei Balken die aktuelle Genauigkeit dieser Positionsbestimmung anzeigen. Sobald eine Genauigkeit im Sub-Meter Bereich erreicht wurde, werden die Balken grün dargestellt und der Nutzer wird dazu aufgefordert, den initialen Cloud Anchor des Gebäudeplanes auf dem Boden zu platzieren und die Position mit dem primären Button zu bestätigen. Wie in Abbildung 9b zu sehen ist, muss dieser platzierte Cloud Anchor nun von jeder Seite gescannt werden, damit eine möglichst gute Abdeckung der Umgebung zur späteren Auflösung entsteht. Dieser Prozess wird anhand eines zunächst weißen Kreises dargestellt, dessen 10 Seitenstücke nach dem Scannen von der

jeweiligen Seite grün gefärbt werden. Sobald der gesamte Kreisring grün ist, wird der Hosting-Prozess des Cloud Anchors automatisch gestartet und die Umgebungsdaten an die Cloud Anchor API gesendet, was innerhalb des in diesem Zustand ausgegrauten primären Buttons mit einer Fortschrittsanzeige visualisiert wird. Die Implementation der VPS-Genauigkeitsanzeige und des Kreisringes wurde aus einem vorherigen Projekt übernommen.



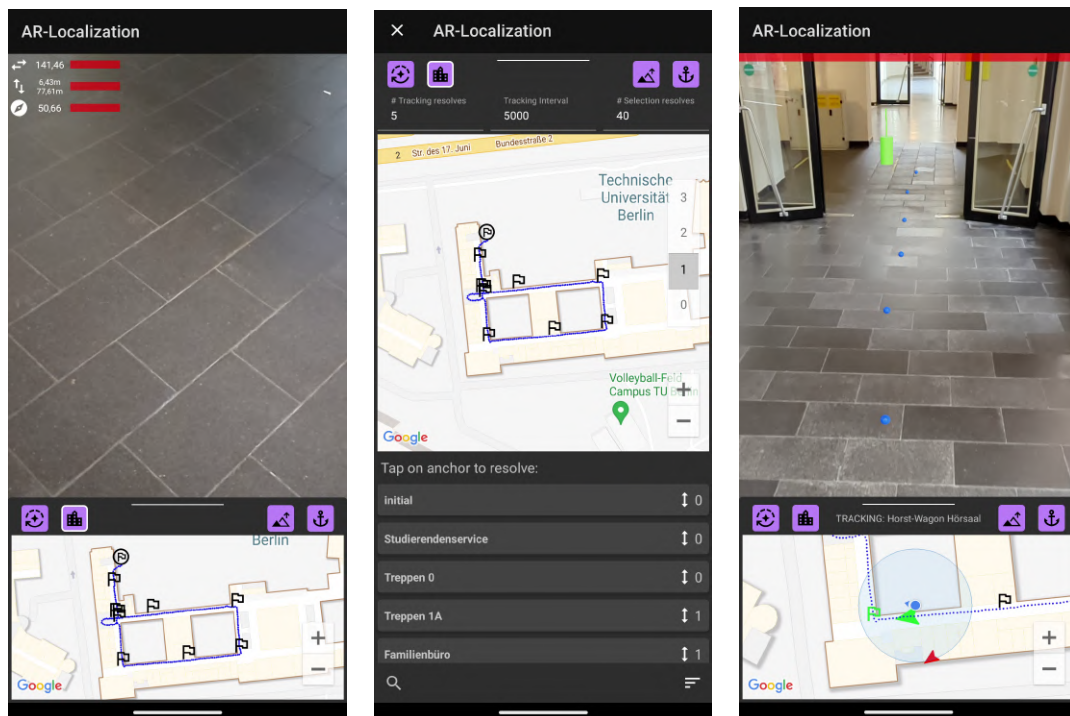
(a) Platzierung des initialen Cloud Anchors (b) Scannen des initialen Cloud Anchors (c) Zustand nach erfolgreichem Hosten des initialen Cloud Anchors

Abbildung 9: Fragment: Erstellen eines neuen Gebäudeplanes

Sobald der Cloud Anchor erfolgreich gehostet wurde, wird ein anderes Modell an diese Stelle platziert, was den Achsen des lokalen Koordinatensystems entspricht. Wie in Abbildung 9c zu sehen ist, verändert sich das User-Interface dieses Fragmentes ebenfalls entsprechend, sodass nun ein Button zum Rückgängig machen des letzten Schrittes sowie ein weiterer Button zum Abschließen der Erstellung sichtbar werden. Da die Genauigkeiten der VPS-Ortung nicht weiter benötigt werden, wird dessen Darstellung zu einem einzelnen farbigen Balken eingeklappt und der Nutzer wird dazu aufgefordert, die gewünschte Strecke abzulaufen und dabei weitere Cloud Anchor zu platzieren. Diese weiteren Cloud Anchor werden nach demselben

Prinzip wie der initiale Anker platziert, nur dass der Nutzer für jeden neuen Anker zunächst einen kurzen Namen als Ortsbeschreibung und dessen Etage innerhalb des Gebäudes in einem Dialog eingeben muss. Dieser Prozess des Erstellens von neuen Cloud Anchorn entlang der gewünschten Strecke wird so lange wiederholt, bis der Nutzer mit dem gewünschten Gebäudeplan fertig ist und die Erstellung über den entsprechenden Button abschließt. Anschließend werden die Informationen und die relativen Koordinaten aller erstellten Cloud Anchor und des gelaufenen Weges anhand von MappingPoints mit dem in Kapitel 4.2.1 beschriebenen Datenmodell an den Firebase Server gesendet und in der Übersicht aller Gebäudepläne dargestellt.

5.5 Lokalisierung anhand eines Gebäudeplanes



(a) Initialer Zustand

(b) Initialer Zustand mit ausgeklapptem BottomSheet

(c) Zustand während des laufenden Trackings

Abbildung 10: Fragment: Lokalisierung mit einem Gebäudeplan

Sobald ein Nutzer die Lokalisierung anhand eines Gebäudeplanes startet, öffnet sich erneut ein Augmented Reality Fragment, wo im oberen Bereich die Genauigkeit der Geospatial API anhand der 3 Balken abgebildet ist. Wie in Abbildung 10a zu sehen ist, ist im unteren Bereich des Bildschirms ein sogenanntes *BottomSheet* platziert, auf dem eine Google Maps Karte der Umgebung abgebildet ist, die zunächst nur

die aktuelle GPS-Position des Nutzers und den ausgewählten Gebäudeplan anzeigt, dessen abgelaufene Strecke in Blau gezeichnet und die Positionen der Cloud Anchor als kleine schwarze Markierungen dargestellt werden.

Um die Lokalisierung zu starten, muss die Anwendung nun zunächst einen beliebigen Cloud Anchor des Gebäudeplanes finden, wofür einige verschiedene Möglichkeiten implementiert sind, die über die 4 Buttons ausgewählt werden können, welche über der Karte angeordnet sind. Der erste Button von links startet das automatische Auflösen aller verfügbaren Cloud Anchor, wobei die maximale Zahl an gleichzeitig aufzulösenden Ankern von der API auf 40 limitiert ist [57] und dies deshalb nicht bei sehr großen Plänen möglich ist. Der zweite Button steht für die Nutzung der VPS-Ortung und ist standardmäßig aktiv, sodass für den Fall, dass mithilfe von Street View Daten eine genaue Position bestimmt werden kann, die Positionen der umliegenden Cloud Anchor anhand von Vorschau-Modellen angezeigt und automatisch versucht, aufgelöst zu werden. Falls sich der Nutzer nicht in einer Umgebung mit Street View befindet, kann der dritte Button verwendet werden, um das aktuelle Stockwerk innerhalb eines Dialogs auszuwählen, woraufhin alle Cloud Anchor dieser Etage versucht, geladen zu werden. Alternativ bietet der letzte Button auch die Möglichkeit, einen konkreten Cloud Anchor aus der Liste aller auszuwählen, wofür durch Tippen das BottomSheet ausgeklappt wird und, wie in Abbildung 10b zu sehen ist, den gesamten Bildschirm verdeckt. Hier wird zunächst die Karte etwas größer dargestellt und einige Optionen zur Konfiguration des Lokalisierungsprozesses sichtbar, aber vor allem wird im unteren Bereich eine Liste mit allen zugehörigen Cloud Anchorn angezeigt, die sowohl durchsucht als auch anhand des Namens oder der Etage sortiert werden kann. Durch Tippen auf einen Cloud Anchor in der Liste oder auch auf eine Markierung auf der Karte werden dieser und alle umliegenden versucht, aufgelöst zu werden, was anhand einer Fortschrittsanzeige neben dem Ankernamen entsprechend dargestellt wird.

Zur gleichzeitigen Auflösung von mehreren Cloud Anchorn wurde die SceneView Bibliothek um eine Methode entsprechend erweitert, sodass für die bis zu 40 möglichen Cloud Anchor gleichzeitig Anfragen zum Auflösen an die API gesendet werden und nach einem ersten erfolgreichen Treffer dieser Cloud Anchor zurückgegeben und das Auflösen der weiteren abgebrochen wird.

Sobald ein erster Cloud Anchor erfolgreich aufgelöst wurde, wird der gesamte Gebäudeplan auf die echte Welt projiziert und alle umliegenden weiteren Cloud Anchor als rote Zylinder und auch die zugehörige Strecke anhand von blauen kugelförmigen MappingPoints dargestellt. Die genauen geografischen Koordinaten und die Ausrichtung des Smartphones können ab diesem Moment in Relation zum aktuel-

len Cloud Anchor mithilfe des relativen SLAM-Trackings von ARCore berechnet und genau verfolgt werden und werden auf der Karte durch einen grünen Pfeil entsprechend dargestellt, wie in Abbildung 10c zu sehen ist. Zusätzlich dazu wird in Rot die durch die Geospatial API bestimmte Position ebenfalls dargestellt, die vor allem für den Vergleich und die folgenden Untersuchungen hilfreich ist. Anhand der mit dem System bestimmten Position innerhalb des Gebäudeplanes werden die jeweils nächsten Cloud Anchor versucht, aufgelöst zu werden. Bei jeder neuen Auflösung wird die Projektion des Gebäudeplanes aktualisiert und die Vorhersagen der weiteren Cloud Anchor und der MappingPoints in Relation zu dem neuen Anker dargestellt, um möglichst genaue Positionen zu erreichen und um potenziellen Drift-Fehlern entgegenzuwirken. Zur Visualisierung für den Nutzer wird immer der aktuell aufgelöste Cloud Anchor sowohl auf der Karte als auch dessen virtuelles AR-Modell in Grün dargestellt.

Während der aktiven Lokalisierung kann ein Nutzer ebenfalls eine Navigation zu einem beliebigen Cloud Anchor starten, indem dieser durch Tippen auf eine Markierung auf der Karte oder in der Liste ausgewählt wird. Entlang der MappingPoints wird eine Route bestimmt, die von der aktuellen Position bis zum ausgewählten Cloud Anchor verläuft und die Strecke als grüne Kugeln in Augmented Reality darstellt. Wie in Abbildung 11 zu sehen ist, wird zusätzlich der ausgewählte Zielpunkt in AR visuell gelb hervorgehoben und dieser sowie die bestimmte Route auf der 2D-Karte ebenfalls farblich dargestellt. Da die Route zur Navigation entlang einer bei der Erstellung abgelaufenen Strecke verläuft, ist dafür gesorgt, dass diese immer begehbar ist. Es kann allerdings nicht in jedem Fall der optimale Weg bestimmt werden, da nur der gemappte Bereich für die Wegfindung und Navigation verwendet werden kann.

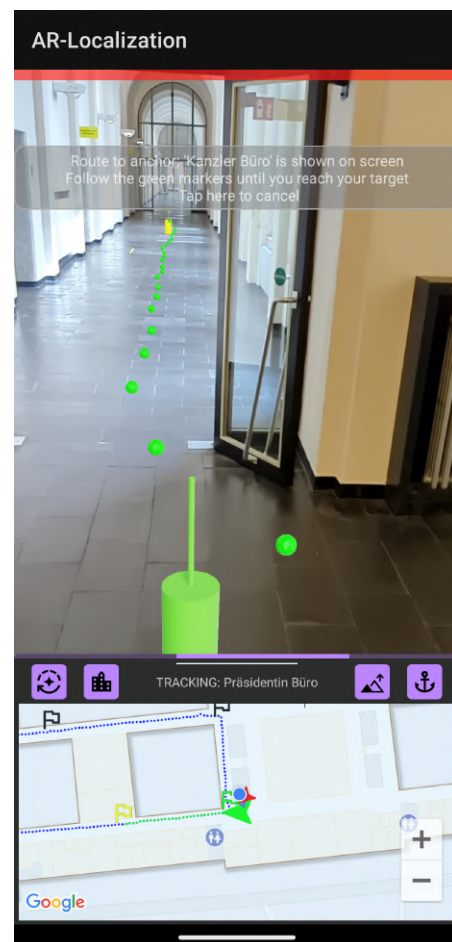


Abbildung 11: Navigation während aktiver Lokalisierung

5.6 Tracking Test

Da die Genauigkeit des entwickelten Ortungssystems stark von der Zuverlässigkeit des relativen SLAM-Trackings in Relation zu platzierten ARCore-Ankern als Referenzpunkt abhängig ist, sollte dies in einer gesonderten Testumgebung genauer untersucht werden. Dafür wurde ein weiteres Fragment entwickelt, das ausschließlich für die Durchführung von Experimenten zu verwenden ist, die im folgenden Kapitel 6 näher erläutert werden.

Wie in Abbildung 12 zu sehen ist, besteht dieses AR-Fragment lediglich aus der Genauigkeitsanzeige für die Geospatial API, einem Button zum Starten und Beenden des Testlaufes und Konfigurationsmöglichkeiten für das gewünschte Log-Intervall und den Abstand zwischen den Anker Platzierungen. Sobald ein Test gestartet wird, wird ein ARCore-Anker an die Startposition platziert, der als Referenz für das relative SLAM-Tracking verwendet wird und vergleichbar mit dem Tracking des Mapping-Prozesses der Anwendung ist. Anschließend werden in dem festgelegten Abstand entlang der abgelaufenen Strecke jeweils neue Anker platziert, die als neue Referenz des relativen Trackings verwendet werden, wobei bei der Auswahl eines Abstandes von 0 Metern nur der initiale Anker verwendet wird. Da laut ARCore Dokumentation [58] die Distanz zum letzten Anker das relative SLAM-Tracking beeinflussen kann und für den Mapping-Prozess relevant ist, soll dies anhand verschiedener Abstände im folgenden Kapitel genauer untersucht werden, wobei das relative Tracking des Visual Positioning System nach Verlust des Street View Fixes als Vergleichswert verwendet wird.



Abbildung 12: Fragment: Genauigkeitstests

5.7 Datenexport

Um im Rahmen des folgenden Kapitels 6 die Genauigkeit der implementierten Lokalisierung auch zu GPS und der Geospatial API vergleichen zu können, wurde ein automatischer Datenexport implementiert, der bei der Erstellung, der Lokalisierung und während der Tracking Tests automatisch alle relevanten Daten speichert und diese nach Abschluss des jeweiligen Prozesses in eine Datei schreibt. Als Datenformat wird die *Keyhole Markup Language* (KML) verwendet, einer für Geoinformationen abgewandelten Form von XML, das für den direkten Import in Google MyMaps Karten zur späteren Visualisierung geeignet ist.

Bei der Erstellung eines Gebäudeplanes bestehen diese Daten sowohl aus den bestimmten Koordinaten aller erstellten Cloud Anchor als auch aus den Positionen der gesamten abgelaufenen Strecke, die mit einer Auflösung von je einem Meter gespeichert werden.

Für den Prozess der Lokalisierung werden deutlich mehr Daten gespeichert, da bei diesem Prozess mit der Ortung von GPS und VPS verglichen werden soll. Dafür werden mehrfach pro Sekunde die jeweils aktuellen bestimmten Koordinaten von GPS, VPS und der entwickelten Ortungsmethode (CPS) gespeichert, die später anhand von Linien in verschiedenen Farben dargestellt werden. Zusätzlich dazu werden die jeweiligen Koordinaten zum Zeitpunkt eines erfolgreichen Auflösens eines Cloud Anchors als Markierung mit dem Namen des entsprechenden Ankers hinzugefügt.

Da bei dem entwickelten Tracking Test nur die Positionen des Visual Positioning Systems mit den mit relativem SLAM-Tracking bestimmten Koordinaten verglichen werden, werden bei diesem Prozess in dem festgelegten Intervall nur die jeweiligen beiden Koordinaten gespeichert, die anschließend ebenfalls anhand von Linien in verschiedenen Farben dargestellt werden können.

6 Untersuchung

6.1 Zielsetzung

Die im Rahmen dieser Arbeit implementierte Anwendung soll in diesem Kapitel auf ihre Funktionalität und Genauigkeit untersucht werden. Dabei ist der Kern der Anwendung in zwei Teile zu unterteilen, wobei der erste Teil dem Prozess des Ausmappens eines Gebäudeplanes entspricht, dessen Präzision vor allem von der relativen Positionsverfolgung zur Ausgangsposition abhängig ist, die durch das relative ARCore SLAM-Tracking durchgeführt wird. Des Weiteren sind die genau bestimmten globalen Koordinaten und die Orientierung der Ausgangsposition von großer Wichtigkeit, da sich dessen mögliche Ungenauigkeit auf alle Punkte eines gesamten Gebäudeplanes übertragen.

Der zweite Teil der Anwendung besteht aus dem Lokalisieren anhand eines existierenden Gebäudeplanes und ist zwar auch von der Genauigkeit des relativen 6DoF-Trackings zwischen den einzelnen Cloud Anchors abhängig, allerdings liegt der Schwerpunkt hier eher auf der initialen Positionsbestimmung. So ist die Genauigkeit und Reproduzierbarkeit des Auflörens eines initialen Cloud Anchors von größter Bedeutung, ebenso aber auch, wie genau und schnell das automatische Laden von weiteren Cloud Anchors entlang der Strecke funktioniert. Dabei ist ebenfalls die Präzision bei der Erstellung eines Gebäudeplanes von enormer Wichtigkeit, da sich daraus die maximal mögliche Genauigkeit von der Lokalisierung ergibt.

Die Untersuchung der implementierten Anwendung soll anhand der folgenden zwei Forschungsfragen abschließend evaluiert und bewertet werden:

Forschungsfrage 1:

Kann durch die Kombination von ARCore Cloud Anchor und der Geospatial API eine bessere und zuverlässigere Indoor-Lokalisierung ermöglicht werden, als mit relativem VPS-Tracking nach Verlust von Street View Daten möglich ist?

Forschungsfrage 2:

Können automatisch Cloud Anchor im Hintergrund geladen werden, um damit das System regelmäßig zu kalibrieren und um dem ARCore Drift entgegenzuwirken, aber auch um eine Lokalisierung mit Start im Gebäude zu ermöglichen?

6.2 Methodik

Während der Entwicklung und Nutzung der implementierten Anwendung wurden regelmäßig neue Arbeitshypothesen über das Verhalten und die Genauigkeit des Systems und dessen zugrundeliegenden Technologien aufgestellt, für die im Rahmen der Untersuchungen jeweils Experimente entworfen und durchgeführt wurden. Die Beobachtungen dieser Experimente und dessen verwendete Testaufbauten werden im folgenden anhand der jeweiligen Hypothese in chronologischer Reihenfolge vorgestellt und evaluiert, wobei sich aus den Beobachtungen häufig weitere zu untersuchende Hypothesen ergaben.

Für die Visualisierung der einzelnen Testdurchläufe wurde Google MyMaps [13] ausgewählt, da damit die Möglichkeit besteht, diverse Ebenen an Punkten, Linien und anderen Markierungen auf eine Google Maps Karte zu zeichnen. Ebenfalls kann auch beispielsweise zwischen der Anzeige der normalen und der Satellitenansicht gewechselt sowie verfügbare Indoor-Karten verwendet werden.

Wie bereits in Kapitel 5.7 beschrieben, werden während der verschiedenen Prozesse der entwickelten Anwendung unterschiedliche Daten für die Untersuchung im Dateiformat *KML* exportiert, das nativ von Google MyMaps zur Visualisierung importiert werden kann. Dadurch konnten auch im Nachhinein neue Untersuchungen und Vergleiche durchgeführt und die Daten von vorheriger Experimenten verwendet werden. Da die exportierten Daten und Visualisierungen anhand von den globalen Koordinaten gespeichert werden, können diese in MyMaps übereinander gelagert angezeigt werden, wodurch beispielsweise die Positionen eines erstellten Gebäudeplanes mit den jeweiligen Ortungsdaten einer Lokalisierung verglichen werden können. Bei der Durchführung der Experimente wurde darauf geachtet, dass diese vor allem in Gebäuden mit verfügbaren Indoor-Karten durchgeführt wurden, da so die Visualisierung mit Google MyMaps deutlich besser dargestellt werden kann.

Für die Untersuchung der implementierten Lokalisierung soll als Vergleich die Ortung mit dem Visual Positioning System (VPS) der Geospatial API und der mobilen GPS-Ortung stattfinden, wobei hier unter GPS die Positionsbestimmung des in Kapitel 2.1.1 beschriebenen A-GPS verstanden wird, da dies für die Ortung innerhalb von Smartphones üblicherweise verwendet wird. Die im Rahmen dieser Arbeit implementierte Lokalisierungsmethode wird im Folgenden auch als *Combined Positioning Service* (CPS) beschrieben.

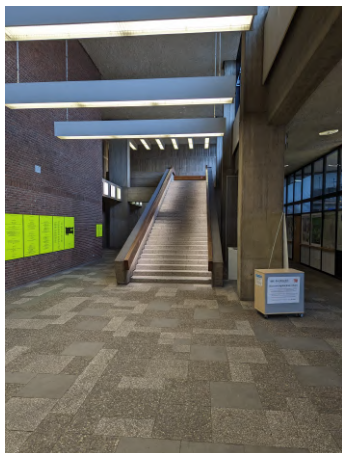
6.3 Testdurchführung

6.3.1 Erstellen eines Planes

Zunächst wird der Prozess des Erstellens eines Gebäudeplanes evaluiert, wofür zu Beginn die folgende Hypothese aufgestellt wird:

Hypothese 1: *Durch das relative Tracking von ARCore und dem regelmäßigen Platzieren von Cloud Anchorn kann eine Strecke genau gemappt werden und anschließend durch eine anfangs geositionierte Ausgangsposition zu einem Gebäudeplan aus globalen Koordinaten umgewandelt werden.*

Um diese Hypothese zu überprüfen, soll zunächst nur die Anwendung verwendet werden, um einen Gebäudeplan zu erstellen. Dabei wurde als Teststandort das Hauptgebäude der TU Berlin ausgewählt, weil dort sehr detaillierte Google Maps Indoor-Karten existieren und des Weiteren relativ nah am Eingang Street View Daten zur initialen Ortung mit der Geospatial API verfügbar sind. Die Fotos in Abbildung 13 geben einen Eindruck über die Gegebenheiten im Gebäude der TU.



(a) TU: Eingangsbereich



(b) TU: Flur im 1. Stock



(c) TU: Treppen im 1. Stock

Abbildung 13: Fotos im TU Gebäude im Bereich der beiden erstellten Gebäudepläne

In der Abbildung 14 ist der erstellte Gebäudeplan anhand der in Google MyMaps importierten geloggten Positionsdaten des Erstellprozesses abgebildet. Die einzelnen blauen Markierungen stellen die jeweiligen platzierten Cloud Anchor dar, während die schwarze Linie den zurückgelegten und damit getrackten Weg des Nutzers zeigt. Der Startpunkt dieses Gebäudeplanes ist die Markierung am oberen linken Rand der Abbildung, wo der Cloud Anchor direkt vor dem Seiteneingang in das Hauptgebäude platziert wurde. Es wurden 2 weitere Cloud Anchor im Erdgeschoss

platziert, bevor über ein Treppenhaus in das erste Obergeschoss gewechselt wurde, was in der Abbildung als Öse am linken Rand zu sehen ist. Im ersten Stock wurde direkt ein erneuter Cloud Anchor platziert und anschließend im Uhrzeigersinn der Flur im Kreis abgelaufen. Dabei wurden regelmäßig weitere Cloud Anchor vor verschiedenen Räumen platziert, wobei in diesem Testdurchlauf insgesamt relativ wenige erstellt wurden. Zur Einschätzung der Größenordnungen beläuft sich die Länge der beiden horizontal verlaufenden Flurstücke auf etwa je 65 Meter, während insgesamt eine Strecke von etwa 240 Metern abgelaufen wurde.

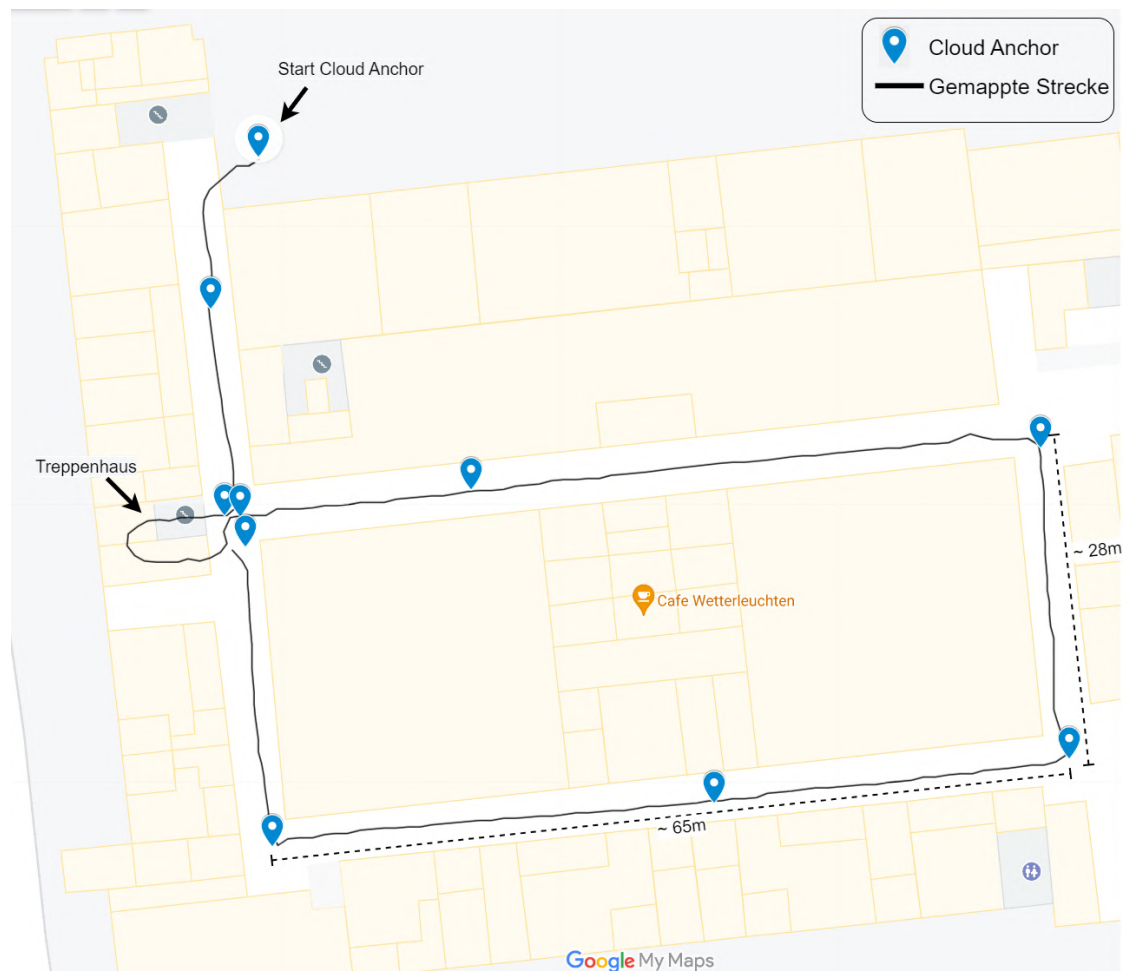


Abbildung 14: Erstellter Gebäudeplan in der TU Berlin mit Start am Seiteneingang und vollständiger Runde im ersten Stock

https://www.google.com/maps/d/u/0/edit?mid=1Q98C2IO_eGknaVSgZKT6NV-Pg8hbQG8

Trotz der insgesamt nur im Durchschnitt alle 25 Meter platzierten Cloud Anchor und der insgesamt ziemlich großen zurückgelegten Distanz konnte ein extrem gu-

ter Plan der abgelaufenen Strecke erstellt werden. Es ist zu sehen, dass sich der gemappte Laufweg immer korrekt in der Mitte des eingezeichneten Flures befindet und dass auch der am Ende der erstellten Strecke platzierte Cloud Anchor an der richtigen Stelle auf der Karte dargestellt werden konnte. Damit konnte die aufgestellte Hypothese 1 als zutreffend bestätigt werden.

Bei der Erstellung von weiteren Gebäudeplänen konnte beobachtet werden, dass Teile der Pläne in manchen Fällen um einen kleinen Faktor rotiert erstellt wurden. Dies lässt sich anhand von potenziellen Berechnungsfehlern und kleineren Ungenauigkeiten des relativen SLAM-Trackings erklären und soll im Weiteren genauer untersucht werden, weshalb die folgende nächste Hypothese aufgestellt wird:

Hypothese 1a: *Da das Erstellen eines Gebäudeplanes vor allem von der kontinuierlichen Genauigkeit des relativen ARCore SLAM-Trackings abhängig ist, können kleine Berechnungsfehler während des Erstellprozesses dafür sorgen, dass ein gesamter Gebäudeplan verdreht wird und damit alle folgenden Positionen eine gewisse Ungenauigkeit aufweisen.*

In Abbildung 15 ist ein erstellter Gebäudeplan zu sehen, dessen abgebildete Strecke sich nicht direkt entlang des eigentlich abgelaufenen Weges befindet, sondern um einige Grad rotiert dargestellt wird. In diesem Testdurchlauf wurde der Plan am Haupteingang des Hauptgebäudes der TU Berlin erstellt, welches in der Abbildung in der oberen rechten Ecke zu sehen ist. Nachdem Cloud Anchor vor dem Haupteingang und im Eingangsbereich erstellt worden sind, wurde eine Treppe ins erste Obergeschoss genommen, gefolgt von einer weiteren Treppe in die zweite Etage, von wo aus die als Rechteck dargestellte Strecke als Rundgang im Uhrzeigersinn entlang eines Flures gemappt wurde. Dabei wurden regelmäßig weitere Cloud Anchor erstellt und so eine insgesamt sehr mit Abbildung 14 vergleichbare Strecke von etwa 250 Metern Gesamtlänge und ca. 65 Metern entlang des langen Flurstückes abgelaufen. Zwar befindet sich die Strecke ein Stockwerk höher und es wurde ein anderer Ausgangspunkt verwendet, allerdings müssten beide dargestellten Strecken entlang des rechteckigen Flures nahezu gleich sein, da beide Flurstücke gleich aussehen und verlaufen.

Es ist zu sehen, dass sich die Darstellung zu Beginn des Gebäudeplanes noch entlang der tatsächlich abgelaufenen Strecke befindet, allerdings nach dem Wechseln der Etagen diese bereits nicht mehr auf dem eingezeichneten Weg des Flures dargestellt wird. Daraus lässt sich vermuten, dass bei der relativen Ortung durch ungleichmäßige Bewegungen während des Treppensteigens sowie auch aufgrund von vermutlich reduzierten Anzahlen an verfolgbaren Feature-Points im Bild der Kamera Ungenauigkeiten durch Fehlberechnungen aufgetreten sind. Aufgrund der Art der

Implementierung und der aufeinander aufbauenden relativen Positionsverfolgung führen bereits einzelne kleine Fehler dazu, dass der gesamte darauf folgende Weg und damit auch der restliche Gebäudeplan beeinträchtigt und wie in diesem Fall um etwa einige Grad fälschlich verschoben erstellt wird.

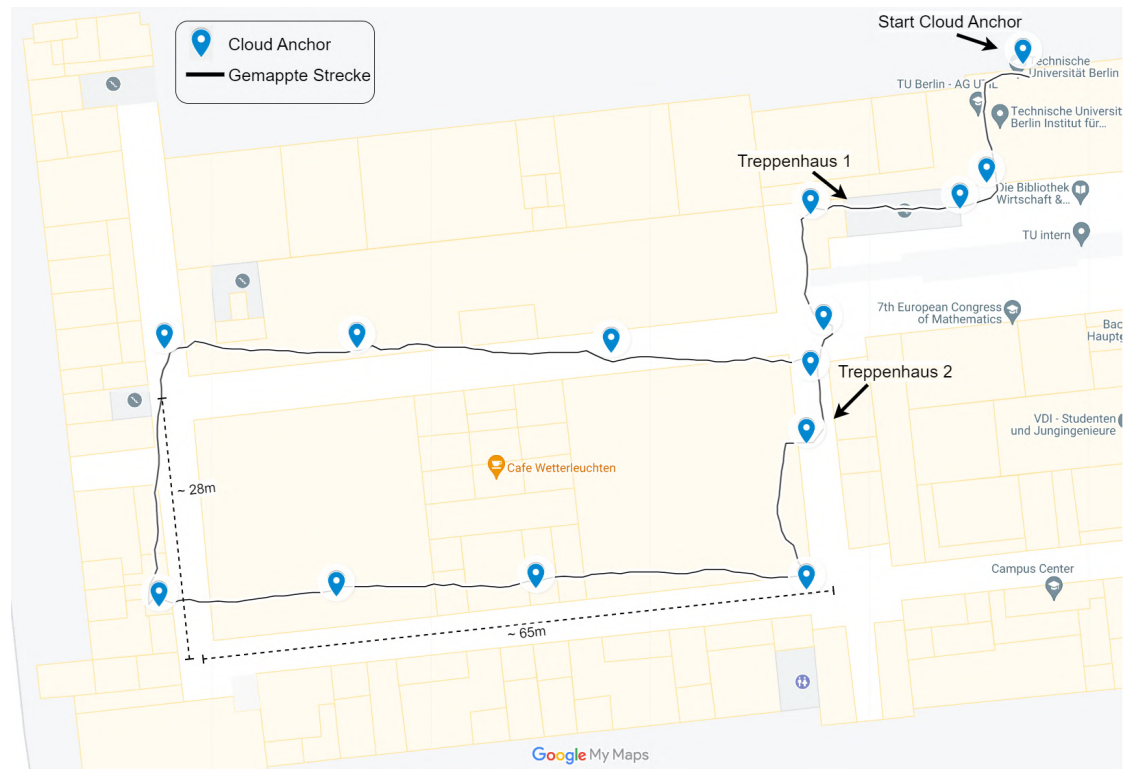


Abbildung 15: Erstellter Gebäudeplan in der TU Berlin mit Start am Haupteingang und vollständiger Runde im zweiten Stock bei aufgetretener Drehung
https://www.google.com/maps/d/u/0/edit?mid=1Q98C2IO_eGknaVSgZKT6NV-Pg8hbQG8

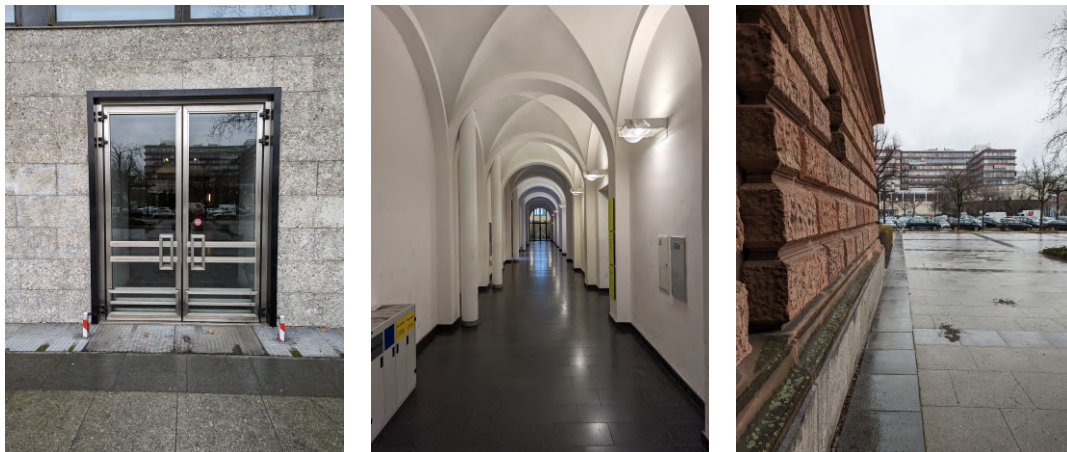
Die Ursachen für diese potenziellen Berechnungsfehler des relativen Trackings soll im Rahmen der Hypothese 4 näher untersucht und daraus Maßnahmen für möglichst genaue Ergebnisse während des Mapping-Prozesses hergeleitet werden.

Im Folgenden soll allerdings zunächst die generelle Genauigkeit des relativen 6DoF-Trackings von ARCore analysiert werden, indem diese mit dem Visual Positioning System der Geospatial API verglichen wird:

Hypothese 2: Die ARCore Geospatial API liefert auch nach dem initialen Verlust von Street View durch visuelles Umgebungstracking weiterhin die aktuelle Position des Nutzers, die ebenfalls von relativem SLAM-Drift beeinflusst sein sollte. Durch die

Verwendung von Ankern sollte dem SLAM-Drift der relativen ARCore Ortung entgegenwirken können, weshalb sich die Genauigkeit dieser Ortung stabiler als die der Geospatial API verhalten sollte.

Um diese Hypothese zu überprüfen, soll das relative ARCore SLAM-Tracking in Relation zu verwendeten Ankern mit dem Visual Positioning System (VPS) der Geospatial API verglichen werden. Um dies optimal untersuchen zu können, wurde das in Kapitel 5.6 beschriebene Fragment zum Vergleichen der beiden Ortungsmethoden implementiert.



(a) Seiteneingang (A in 17) (b) Flur nach 1. Abbiegung (c) Strecke nach Ausgang

Abbildung 16: Fotos der verwendeten Umgebung für die Tracking Tests in der TU

Zunächst soll die relative ARCore Ortung anhand eines einzelnen, anfangs platzierten Ankers mit der kontinuierlichen Ortung der Geospatial API nach dem Verlust von einem initialen Street View Fix verglichen werden. Als Ausgangspunkt für das relative ARCore-Tracking wird dieselbe mit VPS-Ortung bestimmte Position und Orientierung verwendet. Die Strecke für diesen Testlauf wurde so ausgewählt, dass sich der Startpunkt zunächst draußen mit verfügbarem Street View befindet, die Route dann durch ein Gebäude mit mehreren Abzweigungen verläuft und anschließend wieder draußen für eine erneute Positionsbestimmung anhand der Street View Daten endet. Dabei wurde darauf geachtet, dass die gleiche Strecke mit einer hohen Präzision häufig abgelaufen werden kann, aber mit einer insgesamt Distanz von etwa 160 Metern nicht zu kurz ist. Die Gegebenheiten der gewählten Testumgebung können in den Fotos der Abbildung 16 betrachtet werden. Da es keine Referenzwerte für die jeweiligen richtigen Positionen gibt, können nur durch Wiederholung der Durchläufe und anhand des Vergleichens der bestimmten Anfangs- und Abschlusspositionen Rückschlüsse auf die jeweiligen Genauigkeiten und Anfälligkeiten für Drift

The map displays the trajectories of two tracking methods: SLAM-Tracking (green lines) and VPS-Tracking (blue lines). The trajectories are shown around a building with a yellow facade and a grey roof. The map includes labels for 'Startpunkt' (Start point), 'Endpunkt' (End point), 'Ausgang' (Exit), and 'Eingang (A)' (Entrance A). Dimensions of approximately 60m and 40m are indicated. A legend in the top right corner identifies the tracking methods.

https://www.google.com/maps/d/u/0/edit?mid=1AbjsXlwikFiKsWKDFyVX01p_3Kbn1bo

44

on zu dem initialen Anker deutlich näher beisammen liegen und insgesamt weniger Ausreißer haben. Im Gegensatz dazu sind die mit VPS getrackten Strecken deutlich unterschiedlicher und weisen größere Ungleichheiten beziehungsweise Drift in der Ortung auf, sodass am Endpunkt bei erneuter Street View Verfügbarkeit diese Positionen sogar in Richtung der SLAM-Strecke korrigiert werden. Dabei ist zu erkennen, dass am rechten Eingang (A) in das Gebäude einige der Ungenauigkeiten entstehen, die für die leicht fälschlichen Rotation der in Blau gezeichneten VPS-Ortungen vermutlich die Ursache sind. Dies lässt darauf schließen, dass durch die Verwendung eines initialen Ankers in ARCore eine stabilere relative Ortung stattfinden kann, die weniger von Fehlberechnungen und Drift betroffen ist, als es mit dem rein relativen VPS der Fall ist.

Da in der implementierten Anwendung bei der Erstellung der Gebäudepläne nicht nur ein einzelner initialer Anker verwendet wird, sondern regelmäßig weitere entlang der Strecke platziert werden, soll im Folgenden anhand einer ähnlichen Untersuchung geprüft werden, wie sich das relative Tracking verhält, wenn entlang der Strecke mehrere Anker platziert werden. Die folgende Hypothese ergibt sich für diese Untersuchung:

Hypothese 3: *Durch regelmäßiges Platzieren von ARCore Ankern sollte die Genauigkeit des relativen Trackings weiter verbessert werden können, da eine maximale Entfernung von 8 Metern zum letzten Anker in der ARCore Dokumentation für optimale Ortungsergebnisse empfohlen wird.*

Für die Untersuchung dieser Hypothese wurde der gleiche Testaufbau erneut verwendet, allerdings wurden diesmal neben dem initialen Anker während des Ablaufs der Strecke in einem festgelegten Abstand weitere Anker relativ zum jeweils letzten hinzugefügt, die von ARCore zur relativen Ortung verwendet wurden. Dabei wurden wieder je 5 Durchläufe pro Testszenario durchgeführt, wobei zunächst ein Anker an der jeweils aktuellen Position alle 5 Meter platziert wurde, während im zweiten Szenario ein Anker alle 20 Meter platziert wurde. Die jeweils so bestimmten Positionen wurden erneut mit den Daten der VPS-Ortung verglichen und auf einer Karte visualisiert.

In den beiden Abbildungen 18a und 18b sind die Ergebnisse dieser beiden Testdurchläufe zu sehen, wobei im Vergleich zum vorherigen Test mit der Abbildung 17 keine deutliche Veränderung der Genauigkeit des relativen ARCore-Trackings mit der Strecke in Grün festgestellt werden konnte. So verlaufen die jeweils bestimmten Strecken nahezu gleich, beziehungsweise befinden sich die Unterschiede in einer Größenordnung, die nicht auch durch leichte Abwandlungen beim Ablaufen der

Strecke begründbar wären. Im Gegensatz dazu ist deutlich erkennbar, dass die Abweichungen zwischen den einzelnen blau gezeichneten VPS-Strecken vor allem in Abbildung 18a deutlich größer sind und damit ein stärkerer Drift verzeichnet wurde. Diese Beobachtung lässt sich nur durch die deutlich höhere Auslastung des Smartphones erklären, da für jeden einzelnen platzierten und getrackten Anker ein hoher Leistungsaufwand benötigt wird. So scheint es, dass das relative VPS-Tracking deutlich mehr von hoher Auslastung des Gerätes betroffen ist, als das relative ARCore SLAM-Tracking.

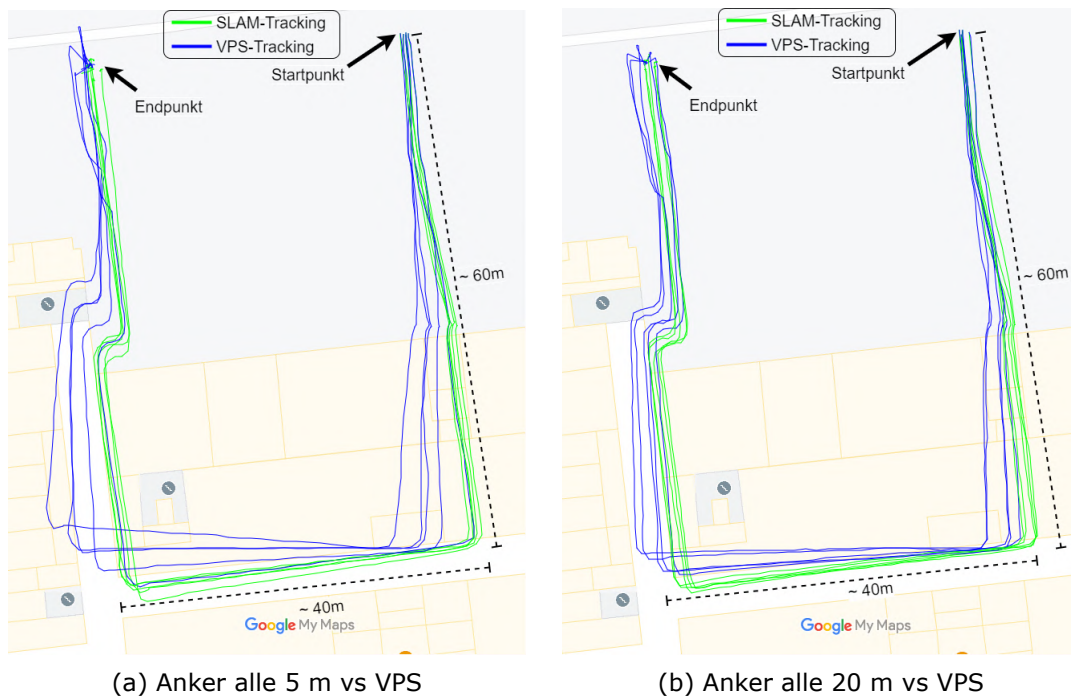


Abbildung 18: Vergleich VPS-Tracking und relatives SLAM-Tracking bei regelmäßiger Platzierung von Ankern alle 5m und alle 20m

<https://www.google.com/maps/d/u/0/edit?mid=1sfA0tA02PCc2VLI6tWH60YnqQ8QqBYg>

Die aufgestellte Hypothese 3 konnte mit diesem Experiment nicht bestätigt werden und obwohl die ARCore Dokumentation eine maximale Distanz zu Ankern von 8 Metern für optimales Tracking empfiehlt [58], konnten trotz einer Strecke von insgesamt etwa 160 Metern keine signifikanten Veränderungen festgestellt werden.

In allen diesen drei Testdurchläufen konnte allerdings beobachtet werden, dass fehlerhafte Positionsbestimmungen und Drift der Ortung scheinbar an einzelnen Orten ihren Ursprung haben, so wie in diesem Szenario bei dem Öffnen der Eingangstür und dem damit verbundenen unregelmäßig bewegten Smartphone und potenziellen Spiegelungen an der Glastür. Daraus lässt sich die folgende Hypothese aufstellen:

Hypothese 4: Die Ungenauigkeiten des relativen Trackings und der beobachtete Drift scheinen von wenigen einzelnen Momenten an Fehlberechnung auszugehen, an denen das Smartphone vermutlich ruckartig oder ungleichmäßig viel bewegt wurde.

Um diese Hypothese zu überprüfen und um die häufigsten Ursachen für den relativen Drift festzustellen, sollen die Testläufe der Hypothese 2 und 3 mit anderen Gegebenheiten wiederholt werden. In den bisherigen Untersuchungen wurde die Kamera des Smartphones so gerade und stabil schräg voraus wie möglich gehalten, während vor allem nur beim Öffnen der Türen entlang der Strecke ungleichmäßigere Bewegungen unvermeidbar waren.

Im Rahmen dieser Hypothese soll zunächst das Verhalten bei horizontalen Drehungen der Kamera untersucht werden, wofür bei dem Ablaufen derselben Strecke wie in den vorherigen Testläufen an den beiden unteren Abbiegungen jeweils eine extra Drehung von 360 Grad im Stand durchgeführt wurde. Dabei wurden diese Drehungen nicht besonders schnell durchgeführt, sondern in einem mit dem restlichen Testlauf vergleichbarem gleichmäßigen Tempo. Im Vergleich dazu soll in einem weiteren Versuch an denselben beiden Stellen das Smartphone sehr schnell um die 90 Grad gedreht werden, um zu prüfen, ob dies deutliche negative Effekte hat.

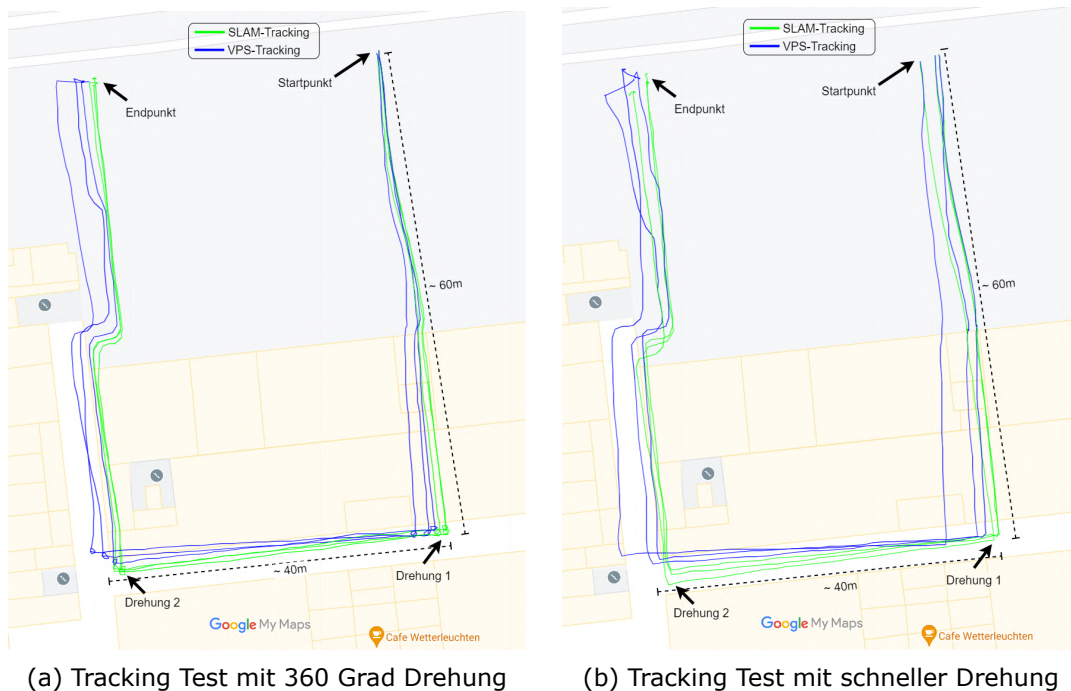


Abbildung 19: Tracking Test mit unterschiedlichen Drehungen

<https://www.google.com/maps/d/u/0/edit?mid=1ARW9kwUtCI28vN1Jgf8nVf4dJN4Zn5o>

In Abbildung 19 ist das Ergebnis dieser beiden Versuche zu sehen, wobei in den beiden unteren Ecken die jeweiligen Drehungen des Smartphones zu erkennen sind. Wie auch bei den vorherigen Tests zeigen die in Blau gezeichneten Linien die Ortung durch VPS, während in Grün die Ergebnisse des relativen ARCore-Trackings zu sehen sind, wobei aufgrund der Ergebnisse des letzten Versuches lediglich mit einem einzelnen initialen Anker getestet wurde. In der Abbildung 19a ist der Durchlauf mit der 360 Grad Drehung dargestellt und in Abbildung 19b die extra schnelle Drehung. Insgesamt sind in beiden Versuchen diese beiden Orte allerdings nicht als Ursache für deutlich verstärkten relativen Drift auszumachen, da erneut bereits vorher größere Rotationen entstehen. Vor allem im zweiten Versuch ist zwar eine größere Abweichung während eines einzelnen Durchlaufes zu erkennen, bei näherer Betrachtung wird aber deutlich, dass diese von der initialen Ungenauigkeit der VPS-Lokalisierung ausgeht. Im Folgenden sollen deshalb anhand von weiteren Experimenten die Auswirkung von extremeren Bewegungen untersucht werden.

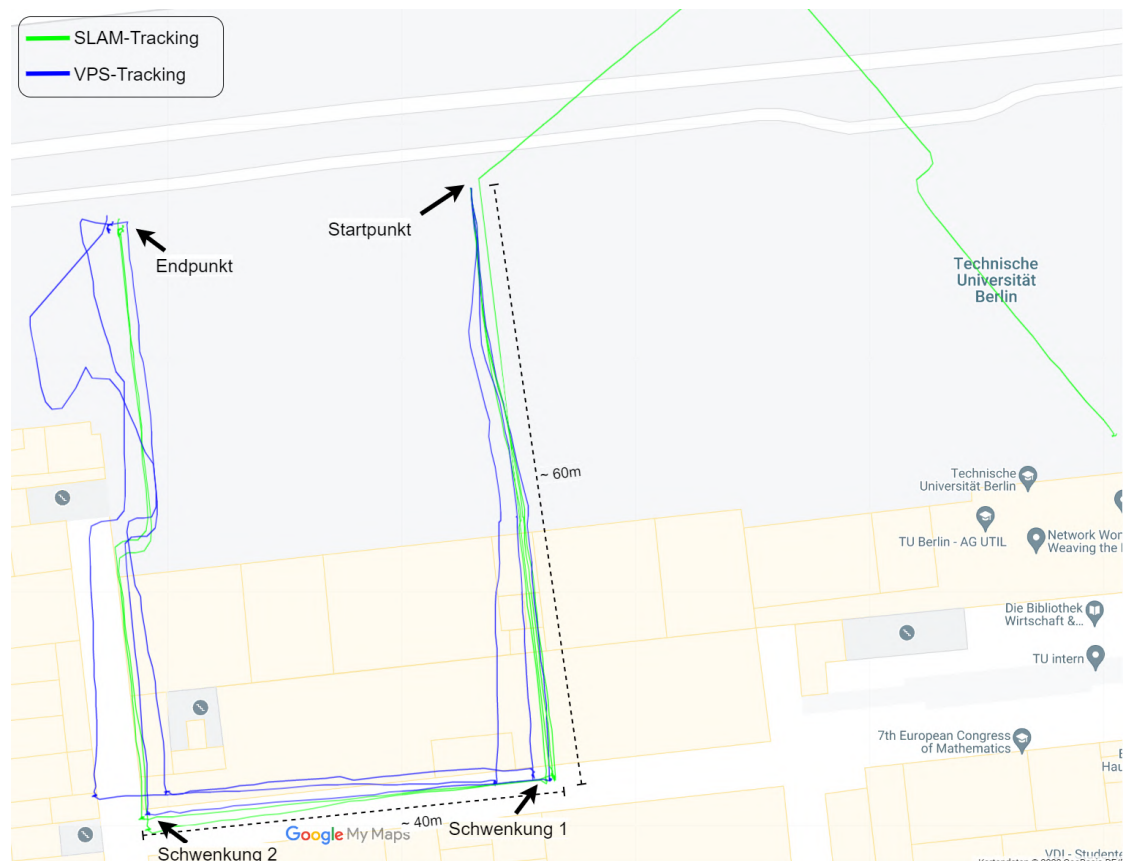


Abbildung 20: Tracking Test mit Umgebungsscan an den Ecken

<https://www.google.com/maps/d/u/0/edit?mid=1sVcs3KcII27dCBIQfWHERrEIjqdpmo>

In der Abbildung 20 ist ein weiterer ähnlicher Versuch abgebildet, bei dem an den jeweiligen beiden unteren Ecken die Smartphone-Kamera für einige Sekunden zum Scannen der Umgebung zu beiden Seiten sowie nach oben und unten geschwenkt wurde. Es ist zu erkennen, dass diese Bewegung durchaus größere Einflüsse auf die Ortung genommen hat, sodass zum einen bei der VPS-Ortung größere Abweichungen zu sehen sind, aber vor allem, dass bei einem der 3 Durchläufe die Rotation des relativen SLAM-Trackings vollständig falsch bestimmt wurde. Bei diesem einen Durchlauf ist genau ab der ersten Ecke mit der ausgeführten Bewegung die bestimmte Strecke um 90 Grad falsch rotiert, woraufhin alle weiteren Positionen relativ zu dieser falschen Rotation bestimmt wurden und die Strecke zurück über den Startpunkt verläuft.

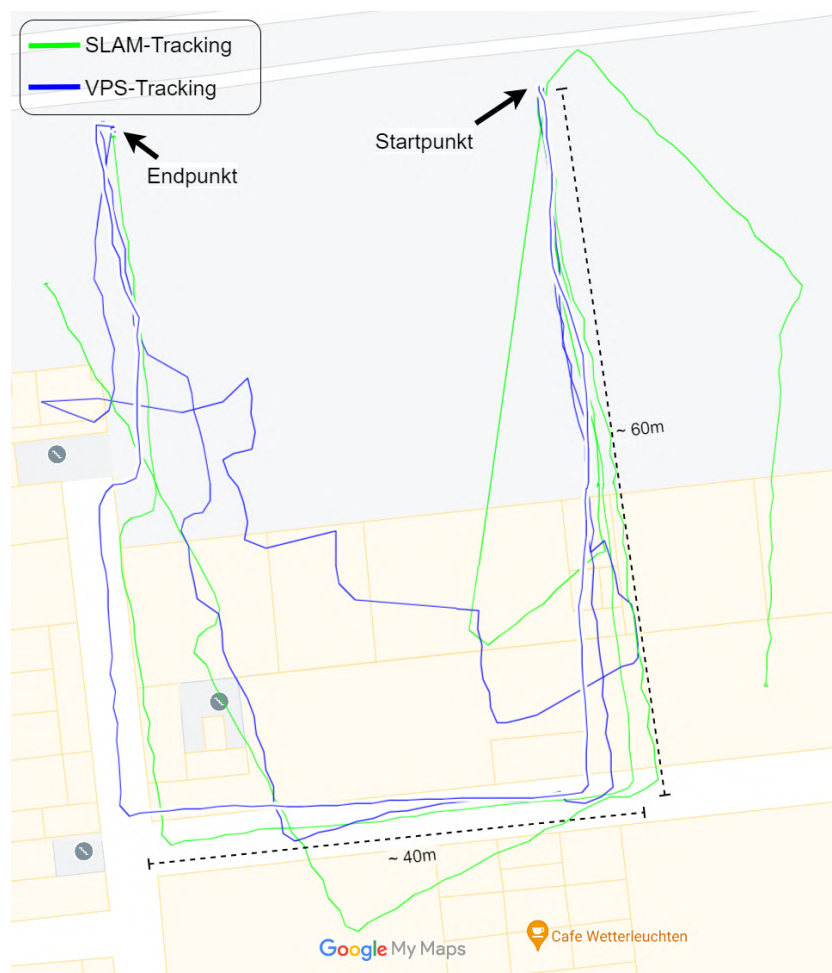


Abbildung 21: Tracking Test mit dauerhaftem extremen Bewegen und Schwanken des Smartphones

<https://www.google.com/maps/d/u/0/edit?mid=1sVcs3KcII27dCBIQfWHERrEIjqdpiMo>

Diese Beobachtung bestätigt die bisherige Vermutung, dass Berechnungsfehler durch ungewöhnliche und ausgeprägte einzelne Bewegungen verursacht werden können und die gesamte folgende Ortung dadurch aufgrund der relativen Bestimmung beeinträchtigt werden.

Während aller bisherigen Versuchen wurde darauf geachtet, dass das Smartphone in der Regel sehr stabil und gerade nach vorne gehalten wurde, damit das relative SLAM-Tracking bestmöglich funktionieren kann. Um zu überprüfen, ob dies einen positiven Effekt auf die Ergebnisse hatte, wurde ein Testdurchlauf der gleichen Strecke durchgeführt, bei dem das Smartphone permanent extrem stark bewegt und hin und her geschwenkt wurde. Wie in Abbildung 21 deutlich wird, sinkt dadurch die Genauigkeit und Zuverlässigkeit des relativen Trackings beider Ortungsmethoden sehr stark und teilweise werden Bewegungen erkannt, die so gar nicht stattgefunden haben. Daraus lässt sich schließen, dass die Fehlerwahrscheinlichkeit deutlich steigt, je intensiver und schneller die Smartphone-Kamera bewegt wird und dass für optimale Ergebnisse bei der relativen Ortung das Smartphone bestmöglich gerade und stabil voraus gehalten werden sollte.

6.3.2 Lokalisierung

Nachdem der Mapping-Prozess und das relative ARCore SLAM-Tracking hinreichend evaluiert und untersucht wurde, wird im Folgenden der Prozess des Lokalisierens mit der entwickelten Anwendung anhand verschiedener Gebäudepläne betrachtet. Dazu wird zunächst die folgende Ausgangshypothese aufgestellt und untersucht:

Hypothese 5: *Durch die relative Ortung zu erfolgreich aufgelösten Cloud Anchors eines erstellten Gebäudeplanes kann die globale Position des Smartphones genau bestimmt werden.*

Um diese Hypothese zu untersuchen und zu überprüfen, soll anhand eines erstellten Gebäudeplanes lokalisiert werden und dabei die mit dem System bestimmten Positionen (CPS) zusammen mit den je aktuellen GPS- und VPS-Positionen zum Vergleich gespeichert werden. Dabei soll die Lokalisierung zunächst an einem Ort mit erfolgreicher Positionsbestimmung der Geospatial API starten, damit die implementierte Ortung mit der relativen VPS-Ortung nach Street View Verlust verglichen werden kann. Außerdem ist ein Start außerhalb eines Gebäudes für den Vergleich mit der GPS-basierten Smartphone Ortung ebenfalls sinnvoll, da diese innerhalb von Gebäuden nur sehr beschränkt funktioniert.

Als Versuchsaufbau wurde eine Strecke am Ostbahnhof Berlin gewählt, wo über eine insgesamt Strecke von etwa 200 Metern von der einen Seite des Gebäudes zur anderen Seite gelaufen werden kann, wobei an den Eingängen auf beiden Seiten jeweils gute Street View Abdeckung für VPS-Ortung verfügbar ist. So konnte am Ende eines Testlaufes die tatsächliche Position mit einem erneuten VPS-Fix bestmöglich überprüft werden. In Abbildung 22 sind einige Fotos der verwendeten Umgebung am Ostbahnhof zu sehen, um einen Eindruck über die Gegebenheiten zu bekommen.



(a) Ostbahnhof Hintereingang

(b) Ostbahnhof Tunnel

(c) Ostbahnhof Haupteingang

Abbildung 22: Fotos am Ostbahnhof entlang des lokalisierten Gebäudeplanes

In der Abbildung 23 sind die Ergebnisse dieses Testlaufes zu sehen, wobei 4 Strecken in verschiedenen Farben abgebildet sind. Zunächst sind in Schwarz die Positionen des korrespondierenden Gebäudeplanes zu sehen, dessen einzelne Cloud Anchor als blaue Markierungen dargestellt sind und dessen Genauigkeit auch die Limitierung der implementierten Ortung darstellt. Die rote Linie zeigt die jeweils mit GPS-Ortung bestimmten Positionen, in Blau werden die jeweiligen Positionen der Geospatial API mit VPS-Ortung dargestellt und in Grün sind die jeweiligen mit der implementierten hybriden Ortungskomponente CPS (Combined Positioning Service) bestimmten Positionen zu sehen.

Bei diesem Versuch wurde am Hintereingang am nordöstlichen Ende des Bahnhofes gestartet. Hier ist zu erkennen, dass sich die Positionen der VPS- und CPS-Ortung an der nahezu selben Stelle befinden. Im Laufe des Versuches und mit steigender Distanz zum Startpunkt ist deutlich zu erkennen, dass die blaue Linie der VPS-Ortung konstant vom gelaufenen Weg abdriftet, während die jeweilige implementierte CPS-Ortung nahezu genau auf der Strecke des erstellten Planes bleibt, die bestmög-

lich abgelaufen wurde. Die Positionierung mit der GPS-Ortung des Android Systems funktioniert in diesem Versuch erstaunlich genau und befindet sich an vielen Stellen sogar näher am tatsächlich abgelaufenen Weg als die VPS-Ortung.

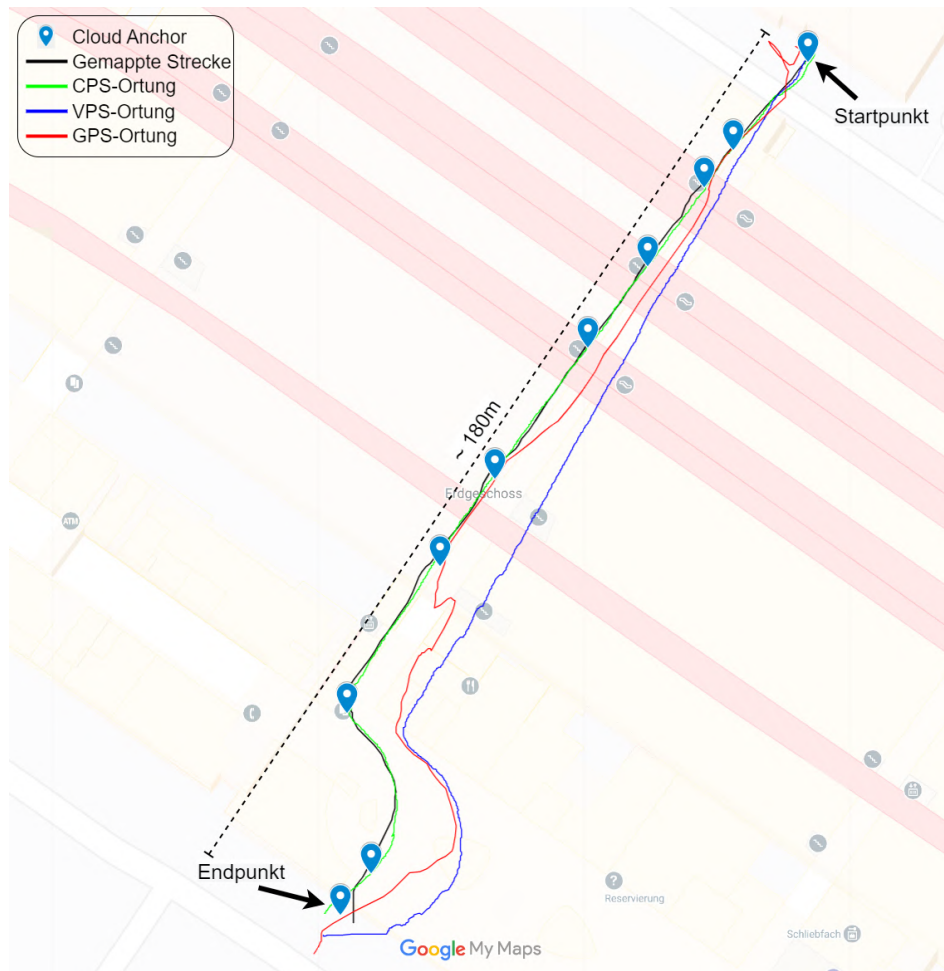


Abbildung 23: Lokalisierung am Ostbahnhof mit Vergleich zu VPS und GPS
https://www.google.com/maps/d/u/0/edit?mid=1L0-y_Sc7z5LijAzsr_K0-2djHPt0ftE

Bei der erneuten VPS-Ortung mit Geospatial-Fix vor dem Haupteingang am Endpunkt der Strecke zur Überprüfung wird deutlich, dass sich die in Blau gezeichnete Position der VPS-Ortung im Moment der Street View Verfügbarkeit stark an die CPS-Position korrigiert. Daraus lässt sich schließen, dass die implementierte CPS-Ortung die Position genauer bestimmen konnte und weniger von Drift betroffen war, womit die Hypothese 5 als bestätigt betrachtet werden kann und sich die folgende neue Hypothese ergibt:

Hypothese 6: *Innerhalb eines Gebäudeplanes kann durch regelmäßiges Auflösen von Cloud Anchor entlang des Weges dem Drift des relativen Trackings entgegen gewirkt werden.*

Um diese Hypothese zu untersuchen, wird bei der Lokalisierung nach einem Gebäudeplan ein weiterer Datensatz erhoben, der die Korrekturen beim Auflösen eines jeweiligen Cloud Anchor aufzeigen soll. Da bei der Lokalisierung die erwarteten Positionen der weiteren Cloud Anchor als Vorhersage dargestellt werden und dessen Positionen im relativen Koordinatensystem bekannt sind, kann zum Zeitpunkt eines erfolgreichen Auflöserns bestimmt werden, wie weit die tatsächliche Position des Ankers von der Vorhersage entfernt ist und damit wie weit die Ungenauigkeit der Ortung korrigiert wurde. Dabei wird die Größe der korrigierten Distanz mit der jeweiligen Entfernung zum letzten Cloud Anchor verglichen, um Rückschlüsse auf die Korrelation mit der Distanz ziehen zu können.

Für diesen Test wurde dieselbe Strecke als Rundgang viermal abgelaufen und dabei lokalisiert, wobei im Anschluss die geloggten Daten zusammengefügt wurden.

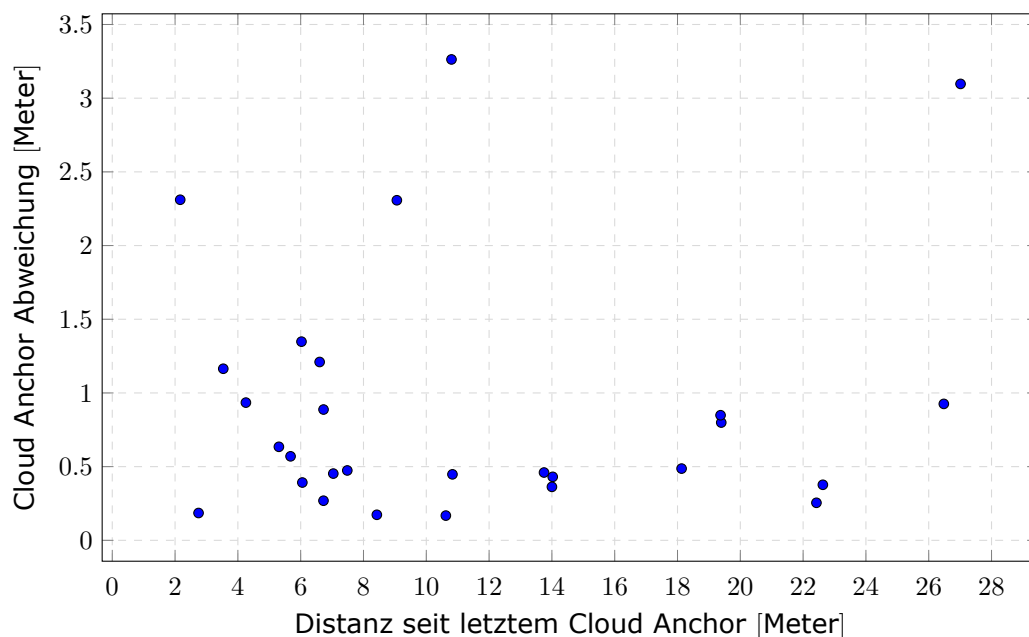


Abbildung 24: Fehlerkorrekturen beim Auflösen der Cloud Anchor

In Abbildung 24 sind die Ergebnisse aus dieser Untersuchung grafisch dargestellt, wobei auf der X-Achse die jeweilige Entfernung seit dem letzten Cloud Anchor und auf der Y-Achse die Größe der Korrektur jeweils in Metern zu sehen sind. In der Ta-

belle 1 sind die wichtigsten Kennzahlen dieser Daten abgebildet, aber auch die Fehlerkorrekturen anhand der jeweiligen 3 Achsen aufgeschlüsselt, wobei die Z-Achse der Höhe entspricht und X und Y zwei beliebige orthogonal zueinander verlaufene Richtungen auf der Ebene sind. Für die Bestimmung des Mittelwertes und des Medians wurden für die 3 Achsen die jeweiligen absoluten Werte verwendet, damit die Größe der durchschnittlichen Abweichungen besser betrachtet werden können.

	Distanz	X-Achse	Y-Achse	Z-Achse
Minimale Abweichung	0,17 m	-1,13 m	-1,05 m	-3,02 m
Maximale Abweichung	3,26 m	0,49 m	0,83 m	1,24 m
Spannweite	3,09 m	1,62 m	1,88 m	4,26 m
Mittelwert	0,90 m	0,38 m	0,35 m	0,62 m
Median	0,53 m	0,26 m	0,17 m	0,34 m
Standardabweichung	0,85 m	0,42 m	0,48 m	0,98 m
Varianz	0,72 m	0,17 m	0,23 m	0,96 m

Tabelle 1: Fehlerkorrekturen beim Auflösen der Cloud Anchor mit Betrachtung der einzelnen Achsen

Anhand der Abbildung 24 wird deutlich, dass in den meisten Fällen die Position eines Cloud Anchor um etwa 50 Zentimeter korrigiert wird, in einigen Fällen aber auch um etwas mehr als einen Meter, wobei auch ein paar Ausreißer von bis zu 3,26 Metern an insgesamtem Fehler vorhanden sind. Diese Ausreißer sogar dafür, dass der Mittelwert der Abweichung bei etwa 90 Zentimetern liegt, während der Median mit etwa 53 Zentimetern deutlich geringer ist. In der Tabelle 1 ist eindeutig zu sehen, dass die größten Ausreißer entlang der Z-Achse entstanden sind, also vor allem nur die Höhe betreffen. Entlang dieser Z-Achse ist schon die mittlere Abweichung etwas höher als die der beiden horizontalen Achsen, allerdings ist an der mit 0,98 Metern etwa doppelt so hohen Standardabweichung und an der mit 0,96 Metern mehr als 4-mal so großen Varianz, die deutliche höhere Streuung der Werte und die damit einhergehende Fehlerursache der Cloud Anchor Auflösungen zu erkennen.

Dabei ist keine deutliche Korrelation zwischen der Größe der Fehler und dem Abstand zwischen den Cloud Anchoren zu erkennen und die Ausreißer treten sowohl bei sehr kleinen als auch bei sehr großen Entfernungen auf. Daraus kann geschlossen werden, dass im Rahmen der untersuchten Größenordnungen der relative Drift der Ortung eher gering ist und sich die Korrekturen vor allem auf die Ungenauigkeiten beim Auflösen der Cloud Anchor selbst beziehen, die in seltenen Fällen scheinbar auch mehrere Meter falsch aufgelöst werden können. Dabei scheint die horizontale Position der Cloud Anchor deutlich genauer aufgelöst werden können als die Höhe, wo die durchschnittlichen Abweichungen etwas größer sind, aber vor allem eine hohe Spannweite und Streuung beobachtet wurde.

Ein regelmäßiges Laden von Cloud Anchoren während der Lokalisierung ist also nicht nur als Korrektur eines potenziellen relativen Drifts hilfreich, sondern sorgt vor allem dafür, dass eventuelle Ungenauigkeiten bei der Auflösung eines einzelnen Cloud Anchors ausgeglichen und korrigiert werden können.

Anhand der folgenden Hypothese soll der automatische Prozess des Ladens von Cloud Anchoren während der Lokalisierung betrachtet werden:

Hypothese 6a: *Während der Lokalisierung können entlang der Strecke automatisch Cloud Anchor erfolgreich aufgelöst werden, ohne dass der Nutzer aktiv eingreifen muss.*

Zu Beginn wurde die Anwendung so implementiert, dass während der aktiven Lokalisierung alle weiteren Cloud Anchor in der Nähe versucht wurden, aufgelöst zu werden. In einigen Testdurchläufen ist dadurch die Problematik beobachtet worden, dass die Cloud Anchor API einen falschen Anker erfolgreich auflöst, obwohl sich dieser eigentlich an einem anderen Ort befindet. Da dies vor allem in Gebäuden vorgekommen ist, in denen sich die visuellen Eigenschaften der Strecke sehr ähneln und wiederholen, lässt sich daraus schließen, dass die Cloud Anchor API teilweise unterschiedliche Orte nicht unterscheiden kann und fälschlich auflöst. Es ist außerdem aufgefallen, dass häufig nur ein einzelner Cloud Anchor des Planes regelmäßig und an verschiedenen Orten fälschlich aufgelöst wurde, woraus zu schließen ist, dass es bei der Erstellung zu ungünstigen Gegebenheiten kommen kann, sodass einzelne Cloud Anchor weniger Übereinstimmung für ein erfolgreiches Auflösen benötigen.

Zur Lösung dieses Problems wurde die Anwendung so angepasst, dass nur die dem Nutzer physikalisch nächsten Cloud Anchor versucht, geladen zu werden, während die weiter entfernten zunächst ignoriert werden. Dabei wurde die Anzahl der maximal gleichzeitig aufzulösenden Cloud Anchor als Parameter in das User Interface eingebaut, damit dies beim Auftreten des Problems untersucht werden konnte.

So wurde bei einem Testdurchlauf der Lokalisierung an der TU Berlin entlang eines ähnlich aussehenden Flures bei zunächst bis zu 5 gleichzeitig aufzulösenden Cloud Anchoren festgestellt, dass regelmäßig ein falscher Cloud Anchor gefunden wurde, woraufhin die Position nicht mehr korrekt bestimmt werden konnte, sondern zu der eigentlichen Position des fälschlich aufgelösten Ankers gesprungen ist. Zwar konnte die Anwendung im Anschluss auch die korrekten Anker weiterhin auflösen, allerdings kann die Positionsbestimmung so nicht mehr gewährleistet werden. In Abbildung 25a ist dieses Verhalten zu sehen, wo die eigentlich gelaufene Strecke mit der VPS-Ortung in Blau gut getrackt werden konnte, während die CPS-Ortung in

Grün einige sehr falsche und größere Sprünge zeigt. Die grünen Markierungen entlang der Strecke entsprechen den CPS-Positionen zum Zeitpunkt eines erfolgreichen Auflöserns der Cloud Anchor.

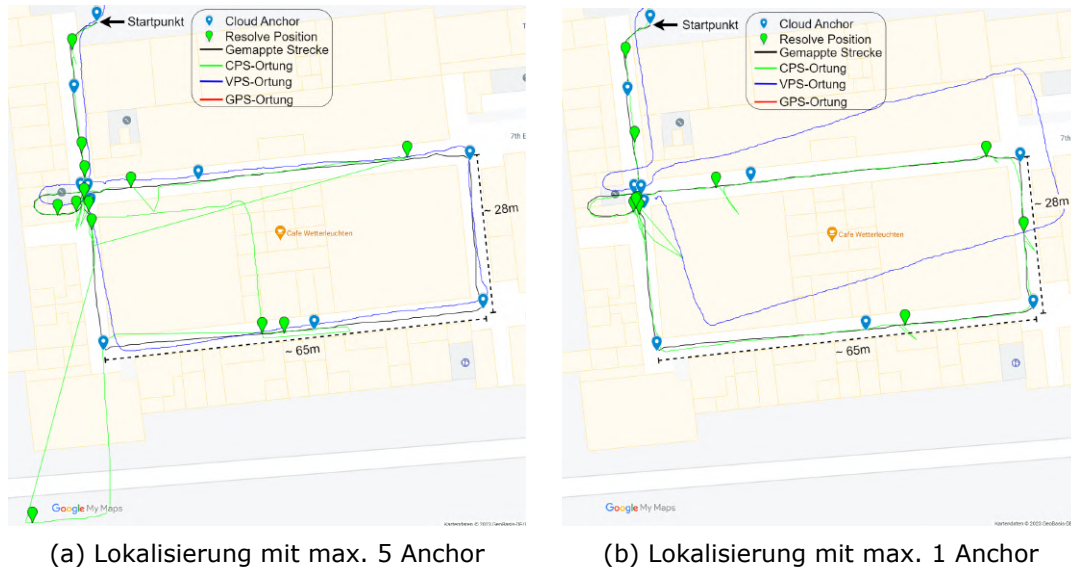


Abbildung 25: Vergleich der Lokalisierung mit verschiedenen Auslöse-Einstellungen

<https://www.google.com/maps/d/u/0/edit?mid=15sJhaV8-YwyWiP9VVGAt5b6D71rj3RE>

Für die gleiche Strecke wurde im Anschluss die Anzahl der gleichzeitig aufzulösernden Cloud Anchor nach dem ersten auf lediglich den aktuell nächsten reduziert, wodurch diese Problematik der fälschlich aufgelösten Anker vollständig beseitigt werden konnte. In der Abbildung 25b ist dieser zweite Durchlauf der Lokalisierung auf gleicher Strecke abgebildet, wobei deutlich zu sehen ist, dass die bestimmten Positionen entlang des tatsächlich abgelaufenen Weges erfolgreich berechnet und keine Cloud Anchor fälschlich aufgelöst wurden. Dabei ist in diesem Durchlauf ein deutlicher Fehler der VPS-Ortung zu sehen, bei der sowohl die initiale Position um 1 bis 2 Meter falsch bestimmt wurde, aber vor allem eine große fälschliche Rotation aufgetreten ist, die allerdings erst nach dem Wechseln der Etage sichtbar wurde. Da dieser Durchlauf bei sehr vergleichbaren Gegebenheiten durchgeführt wurde, deutet diese Beobachtung erneut auf die Unberechenbarkeit und den potenziell großen Einfluss kleiner Fehler bei der visuellen Bewegungsverfolgung. Die kleineren sichtbaren Ausreißer von der Strecke der CPS-Ortung direkt nach dem Auflösen eines Ankers sind durch ein zu frühes Darstellen der neuen Position geschuldet, die bei der Rekalibrierung zunächst eine kurze Zeit benötigt, um alle virtuellen Modelle neu zu platzieren.

Es ist zu erwähnen, dass das in diesen Versuchen beschriebene Verhalten anhand von vielen anderen Gebäudeplänen so nicht aufgetreten ist und dass in den meisten Fällen das automatische Auflösen von 5 oder mehr Cloud Anchors kein Problem dargestellt hat. Daraus lässt sich schließen, dass zum einen die Cloud Anchor Zuverlässigkeit sehr von den Umgebungen abhängig ist, dass aber auch bei der Erstellung scheinbar ungünstige, aber unvorhersehbare Gegebenheiten für diese Probleme sorgen können.

In den bisherigen Untersuchungen konnte beobachtet werden, dass einzelne Cloud Anchor bereits aufgelöst werden können, obwohl sie sich noch nicht direkt in Sicht befinden. Anhand der folgenden Hypothese soll dies überprüft werden:

Hypothese 7: *Cloud Anchor verwenden die gesamten Umgebungsdaten seit dem letzten Anker, weshalb auch zwischen zwei Ankerpunkten bereits der nächste erfolgreich geladen werden kann und die Kamera nicht direkt auf die genaue Position eines Cloud Anchor gehalten werden muss.*

Um diese Hypothese zu untersuchen, wurde der Datenexport des Lokalisierungsprozesses so erweitert, dass die aktuelle Position zum Zeitpunkt einer erfolgreichen Cloud Anchor Auflösung ebenfalls gespeichert wird. Diese Positionen wurden anschließend gemeinsam mit der bestimmten Strecke und dem zugehörigen Gebäudeplan erneut in Google MyMaps importiert, um hier visuell dargestellt werden zu können. Für diesen Versuch wurden die Strecken der GPS- und VPS-Ortung aus der Grafik entfernt, da diese nicht relevant sind und nur der Übersichtlichkeit schaden.

In der Abbildung 26 ist der beschriebene Versuch zu sehen, wobei als grüne Markierungen die jeweiligen Positionen zum Zeitpunkt des erfolgreichen Auflösens zu sehen sind. Der Testdurchlauf hat an der oberen rechten Ecke begonnen und ist im Uhrzeigersinn entlang der Strecke durchgeführt worden. Die erste grüne Markierung oben rechts scheint sich zwar direkt an der Position eines blauen Cloud Anchors zu befinden, allerdings wurde an diesem Punkt bereits der nächste Anker (2) erfolgreich aufgelöst, welcher sich etwa 12 Meter weiter entfernt befindet. So ist auch für alle weiteren Cloud Anchor entlang der Strecke zu sehen, dass diese bereits einige Meter vor der jeweiligen Position erfolgreich aufgelöst wurden und teilweise sich zu dem Zeitpunkt auch noch außerhalb der Sicht befinden.

Anhand dieses Versuches konnte die Hypothese bestätigt werden, dass bei der Erstellung eines Cloud Anchors die gesamte Strecke seit dem vorherigen als zugehörige Daten gespeichert werden, anhand dessen diese entlang des gesamten Weges auch erfolgreich aufgelöst werden können.

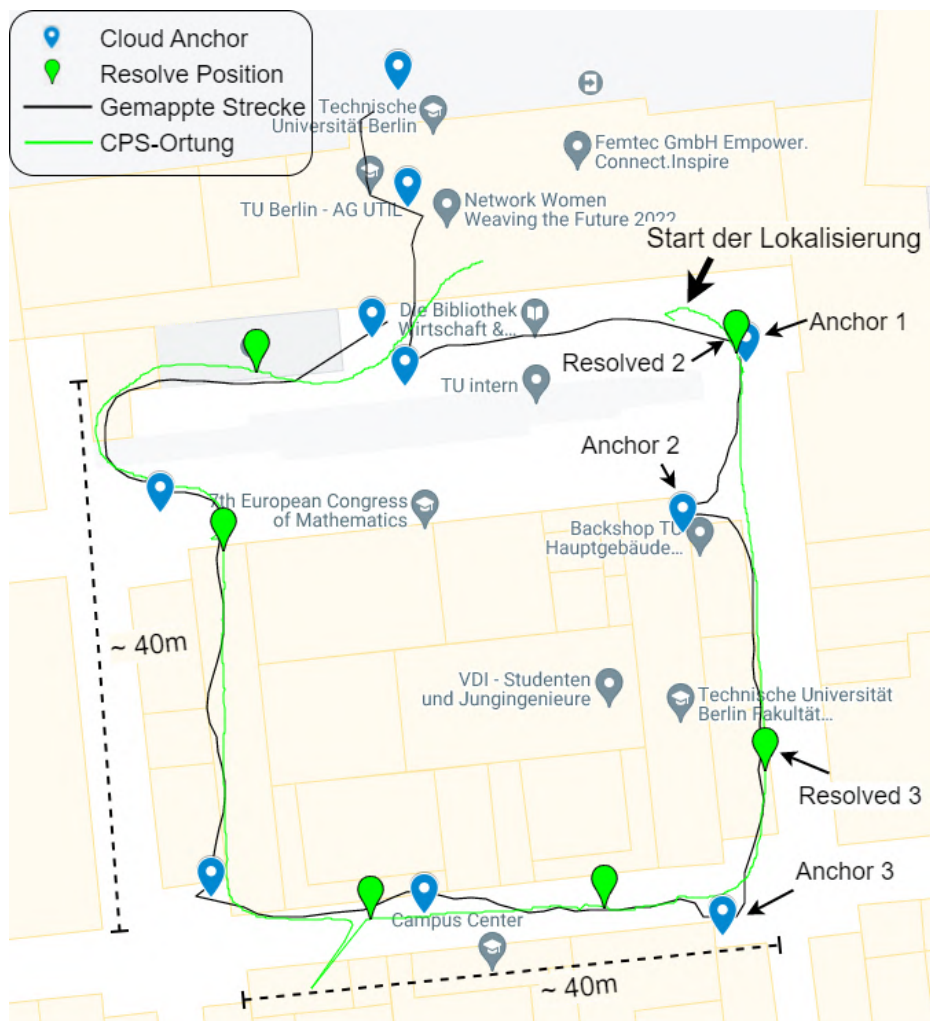


Abbildung 26: Positionen zum Zeitpunkt der Cloud Anchor Auflösungen
https://www.google.com/maps/d/u/0/edit?mid=1Vk0qu4TTf5_7n9_4syfSFz2INZMsPL8

In den bisherigen Untersuchungen wurde die Anwendung in der Regel im Vergleich mit dem VPS-Tracking der Geospatial API betrachtet, wobei oft eine Position mit Street View als Startpunkt der Lokalisierung verwendet wurde. Eine der wichtigsten Anforderungen an die implementierte Anwendung ist allerdings die Möglichkeit, innerhalb eines gemappten Gebäudes die Lokalisierung zu beginnen und sich fortlaufend global orten zu können. Im Folgenden wird dies anhand einer weiteren Hypothese überprüft:

Hypothese 8: *Durch die regelmäßige Platzierung von Cloud Anchoren entlang des Gebäudeplanes kann ein Nutzer an einem beliebigen Ort entlang der gemappten*

Strecke anfangen und sich mit dem System initial verbinden. So kann auch innerhalb von Gebäuden eine genaue globale Lokalisierung stattfinden, während mit GPS- und VPS-Ortung keine annähernd so genauen Positionsbestimmungen zu erwarten sind.

Um diese Hypothese zu überprüfen, wurde die Lokalisierung anhand eines Gebäudeplanes direkt innerhalb des Gebäudes begonnen, wobei die Ergebnisse der verschiedenen Ortungskomponenten erneut verglichen werden sollen. Da der Plan aus insgesamt weniger als 40 Cloud Anchor besteht, konnte das automatische gleichzeitige Auflösen verwendet werden, wodurch ein erster Cloud Anchor innerhalb weniger Sekunden gefunden werden konnte.

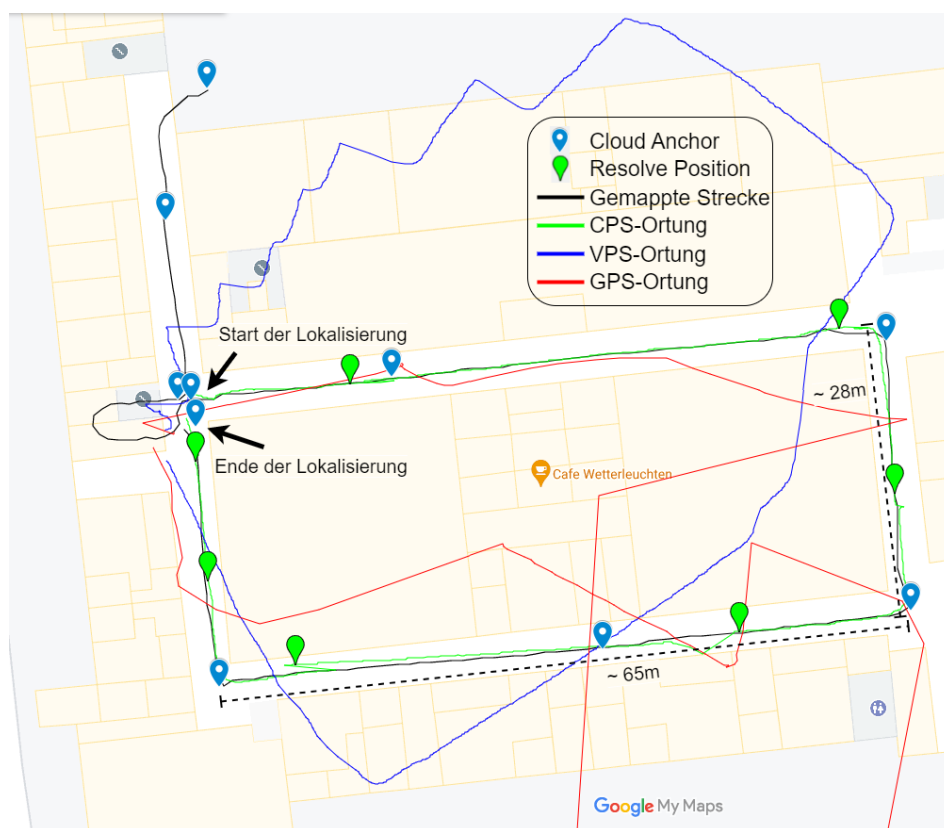


Abbildung 27: Lokalisierung mit Startpunkt im Gebäude

<https://www.google.com/maps/d/u/0/edit?mid=1dDfFvi0iY0BgTUffVL21V7CFCAGsHwk>

In Abbildung 27 sind die Ergebnisse dieses Versuches zu sehen, wobei der zugehörige Gebäudeplan anhand der schwarzen Linie und dessen Cloud Anchor als blaue Markierungen dargestellt werden. Wie auch in den vorherigen Untersuchungen ist die bestimmte Position der VPS-Ortung in Blau und die der mobilen GPS-Ortung in Rot gezeichnet, während die CPS-Ortung in Grün dargestellt ist und die Positionen

zum Zeitpunkt des Auflösens eines Ankers als grüne Markierungen zu sehen sind.

Die Lokalisierung wurde an der oberen linken Ecke des rechteckigen Flures gestartet und von dort aus wurde die Strecke im Uhrzeigersinn als eine vollständige Runde abgelaufen. Dabei ist zu erkennen, dass die Ortung mit der implementierten Methode sehr genau und zuverlässig entlang des Weges eingezeichnet ist, während sowohl die VPS- als auch die GPS-Ortung sehr deutlich schlechter sind und beide für eine genaue Positionierung in diesem Szenario ungeeignet wären. In diesem Versuch ist die Ortung der GPS-Komponente sogar tendenziell näher am gelaufenen Weg, als die der Geospatial API, allerdings gibt es dort einen sehr großen Ausreißer, der für eine einzelne bestimmte Position etwa 250 Meter nach Süden springt, was bei der GPS-Ortung häufiger beobachtet werden konnte.

Insgesamt zeigt diese Untersuchung sehr stark die Vorteile der implementierten Anwendung im Vergleich zu den alternativen globalen Ortungsmethoden für die Indoor-Nutzung auf und macht deutlich, dass die Lokalisierung auch innerhalb von Gebäuden mit derselben Genauigkeit stattfinden kann, wie sie in den vorherigen Versuchen beobachtet wurde.

6.4 Ergebnisse

Im Rahmen der durchgeführten Untersuchungen konnten sehr gute Ergebnisse für die implementierte Anwendung sowie der zugrundeliegenden Technologien beobachtet werden. So konnte zunächst bestätigt werden, dass initiale globale Positionierung mit der Geospatial API in der Regel sehr schnell und zuverlässig stattfindet, was als Referenz für die relativen Gebäudepläne sehr wichtig ist. Die exakte Genauigkeit der bestimmten Koordinaten konnte aufgrund einer fehlenden wahren Position als Vergleich nicht bestimmt werden, allerdings wurde in den Versuchen deutlich, dass die gleiche Position zuverlässig in vielen Durchläufen bestimmt wird und dabei die Genauigkeit augenscheinlich im Vergleich zu Satellitenkarten als Referenz in der Regel im Sub-Meter Bereich lag. Nur in sehr seltenen Fällen wurden Abweichungen im Meterbereich festgestellt, wobei in diesen Fällen dies auch die angezeigte Präzision der Geospatial API widerspiegelt, die normalerweise in einem Bereich zwischen 20 und 60 cm an horizontaler Genauigkeit gependelt ist.

Das anschließende relative Tracking des Visual Positioning System der Geospatial API nach Verlust von Street View Verfügbarkeit wurde mit dem relativen ARCore-Tracking anhand eines initial platzierten Ankers verglichen, wobei die Genauigkeit des relativen ARCore-Trackings bereits hier insgesamt etwas zuverlässiger war und ein geringerer Drift beobachtet wurde. Vor allem in der Kombination mit dem Auflö-

sen von regelmäßig platzierten Cloud Anchoren konnte so mit dem implementierten System eine bessere und zuverlässigere Indoor-Ortung ermöglicht werden, womit die Forschungsfrage 1 als bestätigt betrachtet werden kann.

Die regelmäßig entlang eines Gebäudeplanes platzierten Cloud Anchor konnten während der Lokalisierung automatisch von der Anwendung aufgelöst werden, um Drift oder andere Ungenauigkeiten zu korrigieren. Dabei wurde vor allem mit der Hypothese 6 deutlich, dass bei der Auflösung der Cloud Anchor nicht immer die exakte Position bestimmt werden kann, sondern in seltenen Fällen diese bis zu 3 Meter falsch platziert werden. Deshalb kann das regelmäßige Auflösen neben dem SLAM-Drift auch diese potenziell auftretenden Ausreißer an Ungenauigkeit der Positionierung korrigieren, da diese in den meisten Fällen deutlich genauer bestimmt werden.

Die einzelnen Cloud Anchor können an jeder Stelle innerhalb eines Gebäudeplanes aufgelöst werden und so einen Einstieg in die Lokalisierung ermöglichen, wobei ab einer Größe von über 40 Cloud Anchoren aufgrund der API-Limitationen der Nutzer eine ungefähre Ausgangsposition oder die Etage angeben muss. Die Forschungsfrage 2 konnte anhand der durchgeführten Versuche bestätigt werden, allerdings können ungeeignete Gegebenheiten der Umgebung dazu führen, dass fälschlich ein falscher Cloud Anchor aufgelöst wird. Innerhalb der durchgeführten Untersuchungen konnte dies durch die Einschränkung des aufzulösenden Radius vollständig behoben werden, allerdings ist ein fälschliches Auflösen der Cloud Anchor im Falle von sehr ungünstigen Gegebenheiten nicht auszuschließen.

Insgesamt lässt sich sagen, dass die implementierte Anwendung eine sehr genaue Ortung mit Bestimmung der globalen Koordinaten auch innerhalb von Gebäuden ermöglicht, wobei die Genauigkeit der jeweiligen Positionsbestimmung in der Regel im Bereich der Geospatial API Genauigkeit liegt und dabei eine deutlich zuverlässigere und stabilere Ortung bietet, als diese nach Verlust von Street View Daten. Insgesamt kann so auch mit Start innerhalb von Gebäuden eine Ortungsgenauigkeit erzielt werden, die extrem nah an der Geospatial API bei Street View Verfügbarkeit liegt. Die maximale Genauigkeit der Lokalisierung ist vor allem von der Erstellung eines Gebäudeplanes abhängig, wobei eine ruhige und stabile Kameraführung wichtig ist. Mit einem guten Gebäudeplan können in der Regel mit einer Genauigkeit im Bereich um etwa einen Meter erfolgreich die globalen Koordinaten an jedem gemappten Punkt innerhalb eines Gebäudes zuverlässig bestimmt werden.

6.5 Weitere Beobachtungen

6.5.1 Akkuverbrauch

Durch die Verwendung von Augmented Reality in Kombination mit diversen aufwendigen Berechnungen und des hohen Datenverkehrs der Cloud Anchor und Geospatial APIs wurde bei der entwickelten Anwendung ein sehr hoher Stromverbrauch beobachtet. Im Rahmen einer längeren Versuchsreihe von etwa 2 Stunden an fast dauerhafter Nutzung von verschiedenen Teilen der Anwendung wurde so ein Akkuverbrauch von etwa 80 % der gesamten Gerätekapazität festgestellt. Da es sich bei dem verwendeten Testgerät um ein Google Pixel 6 mit einer Akkukapazität von 4600 mAh handelt, kann so ein Akkuverbrauch der Anwendung von etwa 1840 mAh pro Stunde bestimmt werden, wenn der parallele eigene Verbrauch des Betriebssystems nicht gesondert betrachtet wird.

6.5.2 Datennutzung

Eine der größten Einschränkungen der implementierten Anwendung, ist die ständige Abhängigkeit zu verfügbaren mobilen Daten oder einer WLAN-Verbindung. Durch die Verwendung der ARCore Geospatial API und der Cloud Anchor API besteht die Notwendigkeit einer aktiven Internetverbindung sowohl für das Erstellen eines Gebäudeplanes als auch für das spätere Lokalisieren. Vor allem die Cloud Anchor API sorgt für einen sehr hohen Datenverbrauch, da diese ständig die Umgebungsdaten über das Internet an die API-Server sendet, um diese mit den Daten der aufzulösenden Cloud Anchor abzugleichen, aber auch um Neue zu erstellen. Bei einer langen Versuchsreihe von etwa 2 Stunden konnte ein Datenverbrauch von ungefähr 500 MB beobachtet werden, allerdings wurde eine Mischung verschiedener Versuche durchgeführt. Bei dauerhafter Nutzung des Erstellens oder Lokalisierens wäre ein deutlich höherer Verbrauch zu erwarten.

7 Fazit

Im Rahmen dieser Arbeit konnte eine Augmented Reality Android-Anwendung zur genauen globalen Ortung innerhalb von Gebäuden entwickelt werden. Aus einer Kombination an relativem SLAM-Tracking und ARCore Cloud Anchorn als globale Referenz können digitale Gebäudepläne erstellt werden, die mithilfe der Geospatial API mit genauen geografischen Koordinaten versehen werden. Anhand dieser so erstellten Gebäudepläne kann sich ein Nutzer anschließend auch innerhalb von Gebäuden genau lokalisieren, indem einer der Cloud Anchor durch einfaches Bewegen der Kamera im gemappten Bereich aufgelöst wird. Anschließend wird dem Nutzer die genaue Position und Orientierung auf einer 2D-Karte angezeigt und die gemappte Strecke und die weiteren Cloud Anchor in Augmented Reality dargestellt und regelmäßig zur Kalibrierung aufgelöst.

Es wurden ausgiebige Untersuchungen zu Verhalten und Genauigkeit des entwickelten Systems und der zugrundeliegenden Technologien sowohl bei der Erstellung der Gebäudepläne als auch bei der Lokalisierung vorgenommen. Insgesamt konnte eine sehr hohe Präzision bei der Erstellung von Gebäudeplänen beobachtet werden, die sich in den meisten Fällen im Sub-Meter Bereich befand. So konnten die globalen Koordinaten eines Nutzers innerhalb von Gebäuden sehr genau und zuverlässig bestimmt werden, wobei dies mit einer Genauigkeit sehr nah an der Ortung des Visual Positioning System anhand von Street View Daten möglich war.

Dabei ist die größte beobachtete Einschränkung des entwickelten Systems der potenzielle Drift des relativen SLAM-Trackings bei der Erstellung eines Gebäudeplanes, der vor allem bei abrupten oder zu schnellen Bewegungen des Smartphones auftritt und die Genauigkeit aller folgenden Positionen stark beeinträchtigen kann. Durch das gerade und stabil Halten der Smartphone-Kamera konnte dieser Drift in den meisten Fällen auf ein Minimum reduziert werden. Da aber ein einziger Fehler bei der Positionsverfolgung den gesamten folgenden Plan und damit die maximale Genauigkeit der späteren Lokalisierung beeinträchtigt, sollte jeder erstellte Gebäudeplan durch den Vergleich mit der Indoor-Karte auf das Auftreten von groben Ungenauigkeiten untersucht werden.

Beim Auflösen der Cloud Anchor konnten potenzielle Ungenauigkeiten vor allem bei der Höhe beobachtet werden, die allerdings durch das regelmäßige automatische Auflösen während der Lokalisierung ausgeglichen werden konnten, wodurch ebenfalls ein möglicher relativer Drift korrigiert werden konnte.

Im Vergleich zu herkömmlichen funkbasierten Methoden konnten mit Augmented Reality und der Nutzung visueller Ortungsmethoden sehr viel genauere Ergebnisse bei

der Indoor-Lokalisierung erzielt werden. Dabei ist die Ortungsgenauigkeit vergleichbar mit anderen AR-Lösungen aus der Literatur, ohne dass etwa digitale Baupläne oder andere Voraussetzungen wie externe Hardware vorhanden sein müssen. Des Weiteren können die genauen geografischen Koordinaten innerhalb von Gebäuden bestimmt werden, was in der erreichten Genauigkeit so bisher nicht möglich gewesen ist. Dadurch wird die Verwendung von Indoor-Karten zum Verständnis der bestimmten Position sowie auch zur Wegfindung zu eingezeichneten Orten ermöglicht.

Die implementierte Ortungsmethode eignet sich deshalb für diverse Anwendungsfälle, wo ein Nutzer aktiv seine genaue Position und Orientierung in größeren Gebäuden bestimmen möchte. So sind etwa Bahnhöfe, Flughäfen, Universitäten und Einkaufszentren typische Beispiele für geeignete Anwendungsgebiete, da diese aufgrund der Größe häufig unübersichtlich sind, weshalb die Lokalisierung sowie die Möglichkeit zur genauen Navigation zu ausgewählten Orten sehr hilfreich sein kann. Ein weiterer sehr geeigneter Anwendungsfall wäre ein digitaler Museumsführer, der einen Nutzer entlang einer Route im Museum führen und dabei in Augmented Reality diverse virtuelle Inhalte an die Ausstellungsstücke projizieren kann. Neben diesen Verwendungen in Indoor-Umgebungen kann die Nutzung dieser Ortungsmethode sich auch bei hybriden Anwendungsfällen vor allem in urbanen Gebieten anbieten, wo die Ortung mit GPS nicht ausreichend genau funktioniert. Ein Beispiel wäre hier die Unterstützung von Lieferboten bei der Findung von Eingängen oder Ablageorten.

Lediglich für Anwendungsfälle, wo der Nutzer sein Smartphone in der Tasche hat oder es anderweitig benutzen muss, ist die implementierte Methode aufgrund des benötigten aktiven Lokalisierungsprozesses nicht geeignet, allerdings sollten in der Regel für diese Fälle die Genauigkeiten von Funk- oder GPS-basierten Methoden ausreichend sein.

Durch die Nutzung der ARCore Cloud Anchor und Geospatial APIs ist ein nicht unbeträchtlicher Aspekt des Datenschutzes bei der Verwendung der implementierten Anwendung zu erwähnen, da dauerhaft die Umgebungsdaten im Sichtfeld der Kamera erhoben und an die Google Server zur Verarbeitung gesendet werden. Aufgrund der proprietären und nicht einsehbaren Architektur dieser Services ist die Nutzung und Dauer der Speicherung dieser Daten nicht genauer einsehbar.

8 Ausblick

Da die im Rahmen dieser Arbeit implementierte Anwendung primär für die Untersuchung der entwickelten Ortungsmethode entworfen wurde und eher ein Proof of Concept darstellt, sind einige mögliche Ergänzungen und aufbauende Projekte vorstellbar. Zwar ist die Verwendung von der Anwendung zur Lokalisierung und Navi-

gation in großen Indoor-Umgebungen bereits vollständig möglich, allerdings würde mit einer Nutzerstudie mit Fokus auf dem User-Interface die einfache Benutzbarkeit für neue Nutzer verbessert werden können. Um die Erstellung von großen Gebäudeplänen zu vereinfachen, sollte die Anwendung ebenfalls um eine Möglichkeit der späteren Erweiterung eines Planes ergänzt werden, sodass diese nicht in einem einzelnen Durchlauf erstellt werden müssen. Auch die Möglichkeit von privaten Gebäudeplänen ist durchaus für einige Anwendungsfälle geeignet, damit auch sensible Informationen verwendet werden können, die nicht für die Allgemeinheit bestimmt sind.

Die Entwicklung einer Variante der Anwendung für Augmented Reality Smart-Glasses bietet sich ebenfalls sehr an, um die Nutzung zu ermöglichen, wenn die freie Verfügbarkeit der Hände benötigt wird.

Das nutzerbasierte Erstellen von Gebäudeplänen ist zwar für eine allgemeine Anwendung sehr geeignet, damit eine große Abdeckung an verfügbaren Plänen erreicht werden kann, allerdings bietet sich die Trennung des Erstellprozesses für konkrete Anwendungsfälle von einzelnen Geschäften oder Orten durchaus an. Beispielsweise bei der Nutzung als eine digitale Museumsführer-App, sollte die Anwendung für den Endnutzer lediglich aus der Lokalisierung, Navigation sowie der Übersicht an relevanten Informationen zum aktuellen Standort bestehen, während der Gebäudeplan von den Mitarbeitern erstellt und gepflegt werden sollte, damit für eine bestmögliche Genauigkeit und Vollständigkeit gesorgt werden kann.

Bei der initialen Lokalisierung von sehr großen Gebäudeplänen mit über 40 Cloud Anchors ist die größte Schwierigkeit das Finden eines initialen Cloud Anchors. Zwar sind einige verschiedene Optionen dafür implementiert, diese brauchen allerdings alle einen manuellen Schritt des Benutzers. Zur Lösung dieser Einschränkung könnte das in Kapitel 2.2.1 beschriebene WiFi-Fingerprinting für die grobe initiale Positionsbestimmung verwendet werden, indem bei der Erstellung eines Gebäudeplanes an jedem Cloud Anchor zusätzlich ein WiFi-Fingerprint gespeichert wird. Damit könnte bei der späteren Lokalisierung die ungefähre Position sofort bestimmt werden, anhand dessen die aufzulösenden Cloud Anchor automatisch ausgewählt werden können und wodurch der manuelle Prozess des Auswählens entfällt.

Neben den beschriebenen Erweiterungen der Anwendung sind außerdem einige weitere wissenschaftliche Untersuchungen denkbar. So wäre eine Untersuchung der Microsoft Azure Spatial Anchor oder anderer Cloud Anchor Alternativen als Vergleich mit der aktuellen Lösung hilfreich, um die beste Technologie zur Ortung zu finden. Außerdem könnte durch intensive Untersuchungen der Geospatial API Präzision die maximale Genauigkeit der globalen Referenz der Gebäudepläne besser eingeordnet werden.

9 Literaturverzeichnis

- [1] E. L. Bilawal Sidhu, *Make the world your canvas with the ARCore Geospatial API*. Adresse: <https://developers.googleblog.com/2022/05/Make-the-world-your-canvas-ARCore-Geospatial-API.html> (besucht am 16.02.2023).
- [2] Google Developers, *Cloud Anchors allow different users to share AR experiences*. Adresse: <https://developers.google.com/ar/develop/cloud-anchors> (besucht am 19.02.2023).
- [3] Werner Mansfeld, *Satellitenortung und Navigation - Grundlagen und Anwendung globaler Satellitennavigationssysteme*, Deutsch. Vieweg+Teubner, 2004.
- [4] National Oceanic and Atmospheric Administration, *The Global Positioning System*. Adresse: https://oceanservice.noaa.gov/education/tutorial_geodesy/geo09_gps.html (besucht am 20.02.2023).
- [5] National Coordination Office for Space-Based Positioning, *GPS Accuracy*. Adresse: <https://www.gps.gov/systems/gps/performance/accuracy/> (besucht am 19.02.2023).
- [6] N. Zhu, J. Marais, D. Bétaille und M. Berbineau, „GNSS Position Integrity in Urban Environments: A Review of Literature,” *IEEE Transactions on Intelligent Transportation Systems*, Jg. 19, Nr. 9, S. 2762–2778, 2018. DOI: 10.1109/TITS.2017.2766768.
- [7] M. Singhal und A. Shukla, „Implementation of location based services in android using GPS and web services,” *International Journal of Computer Science Issues (IJCSI)*, Jg. 9, Nr. 1, S. 237, 2012.
- [8] S. Battersby, M. Finn, E. Usery und K. Yamamoto, „Implications of Web Mercator and Its Use in Online Mapping,” *Cartographica: The International Journal for Geographic Information and Geovisualization*, Jg. 49, S. 85–101, Juni 2014. DOI: 10.3138/carto.49.2.2313.
- [9] *Google Maps Gebäudeplan*. Adresse: <https://maps.google.de/help/maps/floorplans/> (besucht am 22.01.2023).
- [10] OpenStreetMap Foundation, *OpenStreetMap*. Adresse: <https://www.openstreetmap.org/> (besucht am 20.02.2023).
- [11] *OpenStationMap*. Adresse: <https://www.openstationmap.org> (besucht am 22.01.2023).
- [12] F. de Metz, *Indoor=*. Adresse: <https://indoorequal.org/> (besucht am 22.01.2023).
- [13] Google MyMaps, *Create and share custom maps with Google My Maps*. Adresse: https://www.google.com/intl/en_us/maps/about/mymaps/ (besucht am 20.02.2023).

- [14] Google MyMaps, *Import map features from a file*. Adresse: <https://support.google.com/mymaps/answer/3024836?hl=en> (besucht am 20.02.2023).
- [15] P. Davidson und R. Piché, „A Survey of Selected Indoor Positioning Methods for Smartphones,” *IEEE Communications Surveys & Tutorials*, Jg. 19, Nr. 2, S. 1347–1370, 2017. DOI: 10.1109/COMST.2016.2637663.
- [16] B. Viel und M. Asplund, „Why is fingerprint-based indoor localization still so hard?” In *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*, 2014, S. 443–448. DOI: 10.1109/PerComW.2014.6815247.
- [17] A. Yassin, Y. Nasser, M. Awad u. a., „Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications,” *IEEE Communications Surveys & Tutorials*, Jg. 19, Nr. 2, S. 1327–1346, 2017. DOI: 10.1109/COMST.2016.2632427.
- [18] G. M. Mendoza-Silva, J. Torres-Sospedra und J. Huerta, „A Meta-Review of Indoor Positioning Systems,” *Sensors*, Jg. 19, Nr. 20, 2019, ISSN: 1424-8220. DOI: 10.3390/s19204507.
- [19] F. Zafari, A. Gkelias und K. K. Leung, „A Survey of Indoor Localization Systems and Technologies,” *IEEE Communications Surveys & Tutorials*, Jg. 21, Nr. 3, S. 2568–2599, 2019. DOI: 10.1109/COMST.2019.2911558.
- [20] X. Zhao, Z. Xiao, A. Markham, N. Trigoni und Y. Ren, „Does BTLE measure up against WiFi? A comparison of indoor location performance,” in *European Wireless 2014; 20th European Wireless Conference*, 2014, S. 1–6.
- [21] Y. Shu, C. Bo, G. Shen, C. Zhao, L. Li und F. Zhao, „Magicol: Indoor Localization Using Pervasive Magnetic Field and Opportunistic WiFi Sensing,” *IEEE Journal on Selected Areas in Communications*, Jg. 33, Nr. 7, S. 1443–1457, 2015. DOI: 10.1109/JSAC.2015.2430274.
- [22] R. T. Azuma, „A Survey of Augmented Reality,” *Presence: Teleoperators and Virtual Environments*, Jg. 6, Nr. 4, S. 355–385, Aug. 1997. DOI: 10.1162/pres.1997.6.4.355.
- [23] Microsoft, *Microsoft HoloLens 2: For precise, efficient hands-free work*. Adresse: <https://www.microsoft.com/en-us/hololens> (besucht am 18.02.2023).
- [24] Apple Developers, *ARKit: Integrate iOS device camera and motion features to produce augmented reality experiences in your app or game*. Adresse: <https://developer.apple.com/documentation/arkit/> (besucht am 18.02.2023).
- [25] Google Developers, *With ARCore, build new augmented reality experiences that seamlessly blend the digital and physical worlds*. Adresse: <https://developers.google.com/ar> (besucht am 18.02.2023).

- [26] PTC, Inc., *Vuforia Engine*. Adresse: <https://developer.vuforia.com/> (besucht am 18.02.2023).
- [27] *Wikitude Augmented Reality SDK*. Adresse: <https://www.wikitude.com/products/wikitude-sdk/> (besucht am 18.02.2023).
- [28] J. Zhao, „A Review of Wearable IMU (Inertial-Measurement-Unit)-based Pose Estimation and Drift Reduction Technologies,” *Journal of Physics: Conference Series*, Jg. 1087, Nr. 4, S. 042 003, Sep. 2018. DOI: 10.1088/1742-6596/1087/4/042003.
- [29] C. Cadena, L. Carlone, H. Carrillo u. a., „Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age,” *IEEE Transactions on Robotics*, Jg. 32, Nr. 6, S. 1309–1332, 2016. DOI: 10.1109/TR0.2016.2624754.
- [30] Google Developers, *ARCore Fundamental concepts*. Adresse: <https://developers.google.com/ar/develop/fundamentals> (besucht am 18.02.2023).
- [31] W. Sakpere, M. Adeyeye Oshin und N. Mlitwa, „A State-of-the-Art Survey of Indoor Positioning and Navigation Systems and Technologies,” *South African Computer Journal*, Jg. 29, S. 145, Dez. 2017. DOI: 10.18489/sacj.v29i3.452.
- [32] Microsoft, *Azure Spatial Anchors overview*. Adresse: <https://learn.microsoft.com/en-us/azure/spatial-anchors/overview> (besucht am 19.02.2023).
- [33] PTC, Inc., *Vuforia Area Targets*. Adresse: <https://library.vuforia.com/environments/area-targets> (besucht am 18.03.2023).
- [34] Neogoma, *Stardust Visual Positioning SDK: Map, train, update, locate!* Adresse: <https://neogoma.com/> (besucht am 02.03.2023).
- [35] Google Developers, *Build global-scale, immersive, location-based AR experiences with the ARCore Geospatial API*. Adresse: <https://developers.google.com/ar/develop/geospatial> (besucht am 16.02.2023).
- [36] T. Reinhardt, *Using Global Localization to Improve Navigation*. Adresse: <https://ai.googleblog.com/2019/02/using-global-localization-to-improve.html> (besucht am 19.02.2023).
- [37] Apple Developers, *Tracking Geographic Locations in AR*. Adresse: https://developer.apple.com/documentation/arkit/content_anchors/tracking_geographic_locations_in_ar (besucht am 21.02.2023).
- [38] B.-C. Huang, J. Hsu, E. T.-H. Chu und H.-M. Wu, „ARBIN: Augmented Reality Based Indoor Navigation System,” *Sensors*, Jg. 20, Nr. 20, 2020, ISSN: 1424-8220. DOI: 10.3390/s20205890.

- [39] A. Ahmad, P. Claudio, A. Alizadeh Naeini und G. Sohn, „WI-FI RSS Fingerprinting For Indoor Localization Using Augmented Reality," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Jg. V-4-2020, S. 57–64, 2020. DOI: 10.5194/isprs-annals-V-4-2020-57-2020.
- [40] P. Herbers und M. König, „Indoor Localization for Augmented Reality Devices Using BIM, Point Clouds, and Template Matching," *Applied Sciences*, Jg. 9, Nr. 20, 2019, ISSN: 2076-3417. DOI: 10.3390/app9204260.
- [41] F. Baek, I. Ha und H. Kim, „Augmented reality system for facility management using image-based indoor localization," *Automation in construction*, Jg. 99, S. 18–26, 2019.
- [42] J. Chen, S. Li und W. Lu, „Align to locate: Registering photogrammetric point clouds to BIM for robust indoor localization," *Building and Environment*, Jg. 209, S. 108675, 2022.
- [43] B. Mahmood, S. Han und D.-E. Lee, „BIM-Based Registration and Localization of 3D Point Clouds of Indoor Scenes Using Geometric Features for Augmented Reality," *Remote Sensing*, Jg. 12, Nr. 14, 2020, ISSN: 2072-4292. DOI: 10.3390/rs12142302.
- [44] T. Varelas, A. Pentefountas, C. Georgiadis und D. Kehagias, „An AR Indoor Positioning System Based On Anchors," Bd. 6, Dez. 2020. DOI: 10.20319/mijst.2020.63.4357.
- [45] L. Platinsky, M. Szabados, F. Hlasek u. a., „Collaborative augmented reality on smartphones via life-long city-scale maps," in *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, IEEE, 2020, S. 533–541. DOI: 10.48550/ARXIV.2011.05370.
- [46] J. Mercier, N. Chabloz, G. Dozot, O. Ertz, E. Bocher und D. Rappo, „BiodivAR: A Cartographic Authoring Tool for the Visualization of Geolocated Media in Augmented Reality," *ISPRS International Journal of Geo-Information*, Jg. 12, Nr. 2, 2023, ISSN: 2220-9964. DOI: 10.3390/ijgi12020061.
- [47] Google Developers, *Cloud Anchors developer guide for Android*. Adresse: <https://developers.google.com/ar/develop/java/cloud-anchors/developer-guide> (besucht am 16.03.2023).
- [48] Firebase, *Cloud Firestore*. Adresse: <https://firebase.google.com/docs/firestore> (besucht am 14.02.2023).
- [49] Firebase, *Get started with Cloud Firestore Security Rules*. Adresse: <https://firebase.google.com/docs/firestore/security/get-started> (besucht am 14.02.2023).

- [50] Google Developers, *Overview of One Tap sign-in on Android*. Adresse: <https://developers.google.com/identity/one-tap/android/overview> (besucht am 14.02.2023).
- [51] Firebase, *Firebase Authentication*. Adresse: <https://firebase.google.com/docs/auth> (besucht am 14.02.2023).
- [52] Google Developers, *ARCore: Quickstart for Android*. Adresse: <https://developers.google.com/ar/develop/java/quickstart> (besucht am 18.03.2023).
- [53] Google Developers, *Build ARCore apps without OpenGL*. Adresse: <https://developers.google.com/sceneform> (besucht am 13.02.2023).
- [54] Thomas Gorisse et. al., *Sceneform Maintained SDK for Android*. Adresse: <https://github.com/SceneView/sceneform-android> (besucht am 13.02.2023).
- [55] Thomas Gorisse et. al., *SceneView Android: 3D and AR Android Composable and View with Google Filament and ARCore*. Adresse: <https://github.com/SceneView/scenview-android> (besucht am 13.02.2023).
- [56] Google, *Filament is a real-time physically-based renderer written in C++. It is mobile-first, but also multi-platform*. Adresse: <https://google.github.io/filament/> (besucht am 10.03.2023).
- [57] Google Developers, *Cloud Anchors developer guide: Resolve a previously hosted anchor*. Adresse: https://developers.google.com/ar/develop/java/cloud-anchors/developer-guide?hl=en#resolve_a_previously_hosted_anchor (besucht am 03.03.2023).
- [58] Google Developers, *Working with Anchors*. Adresse: <https://developers.google.com/ar/develop/anchors> (besucht am 27.02.2023).

10 Abbildungsverzeichnis

1	GPS-Ortung anhand von 4 sichtbaren Satelliten [4]	5
2	Multipathing Effekt bei GPS-Ortung [6]	6
3	Vergleich der Lokalisierung mit BLE und WiFi [20]	9
4	Übersicht der zu implementierenden Anwendung	22
5	UML-Diagramm: Aufbau Gebäudeplan	23
6	Übersicht der entworfenen System-Architektur	27
7	Fragment: Login	28
8	Fragment: Übersicht der verfügbaren Gebäudepläne	29
9	Fragment: Erstellen eines neuen Gebäudeplanes	31
10	Fragment: Lokalisierung mit einem Gebäudeplan	32
11	Navigation während aktiver Lokalisierung	34
12	Fragment: Genauigkeitstests	35
13	Fotos im TU Gebäude im Bereich der beiden erstellten Gebäudepläne .	39
14	Erstellter Gebäudeplan in der TU Berlin mit Start am Seiteneingang und vollständiger Runde im ersten Stock	40
15	Erstellter Gebäudeplan in der TU Berlin mit Start am Haupteingang und vollständiger Runde im zweiten Stock bei aufgetretener Drehung . . .	42
16	Fotos der verwendeten Umgebung für die Tracking Tests in der TU . . .	43
17	Vergleich VPS-Tracking und relatives SLAM-Tracking mit initialem Anker	44
18	Vergleich VPS-Tracking und relatives SLAM-Tracking bei regelmäßiger Platzierung von Ankern alle 5m und alle 20m	46
19	Tracking Test mit unterschiedlichen Drehungen	47
20	Tracking Test mit Umgebungsscan an den Ecken	48
21	Tracking Test mit dauerhaftem extremen Bewegen und Schwanken des Smartphones	49
22	Fotos am Ostbahnhof entlang des lokalisierten Gebäudeplanes	51
23	Lokalisierung am Ostbahnhof mit Vergleich zu VPS und GPS	52
24	Fehlerkorrekturen beim Auflösen der Cloud Anchor	53
25	Vergleich der Lokalisierung mit verschiedenen Auslösungs-Einstellungen	56
26	Positionen zum Zeitpunkt der Cloud Anchor Auflösungen	58
27	Lokalisierung mit Startpunkt im Gebäude	59

11 Tabellenverzeichnis

1	Fehlerkorrekturen beim Auflösen der Cloud Anchor mit Betrachtung der einzelnen Achsen	54
---	--	----

12 Abkürzungsverzeichnis

6DoF:	Six Degrees of Freedom
A-GPS:	Assisted GPS
API:	Application Programming Interface
AR:	Augmented Reality
BLE:	Bluetooth Low Energy
CPS:	Combined Positioning Service (entwickeltes System)
GNSS:	Global Navigation Satellite System
GPS:	Global Positioning System
IMU:	Inertial Measurement Unit
KML:	Keyhole Markup Language
RSS:	Received Signal Strength
UID:	Unique Identifier
SLAM:	Simultaneous Localization and Mapping
TU Berlin:	Technische Universität Berlin
VPS:	Visual Positioning System (Geospatial API)
XML:	Extensible Markup Language