

Masterarbeit

Studiengang Angewandte Informatik
Fachbereich 4 Wirtschaftswissenschaften 2
Hochschule für Technik und Wirtschaft Berlin

Verbesserte Objekterkennung in Augmented Reality
Anwendungen unter Verwendung von Sensorsystemen

1. Prüfer: Prof. Dr. Jürgen Sieck
2. Prüfer: Prof. Dr. Christian Herta

Vorgelegt von Christian Bunk, B.Sc.
zur Erlangung des akademischen Grades **Master of Science (M.Sc.)**
24. August 2012

Danksagung

Ich möchte mich bei Prof. Dr. Jürgen Sieck für die Betreuung dieser Arbeit und den hilfreichen Hinweisen bedanken.

Besonderen Dank geht an meine Freundin, Juana Conrady. Sie hat mich immer wieder motiviert und mir durch ihren ganz eigenen Blick auf diese Arbeit sehr geholfen.

Auch danke ich meinen Eltern und meinen Großeltern, die mich während meines Studiums sehr unterstützt haben.

Vielen dank auch an Sharif Abdel-Halim, Marvin Girbig, Alexander Miller, Daniel Remus und Christian Sandvoß für das kritische Lesen meiner Arbeit.

Inhaltsverzeichnis

Abbildungsverzeichnis	iv
1 Einleitung	1
1.1 Motivation und Problemstellung	1
1.2 Zielsetzung	2
1.3 Struktur der Arbeit	2
2 Grundlagen	4
2.1 Augmented Reality	4
2.1.1 Markerbasierte Erkennung	6
2.1.2 Markerlose Erkennung: Scale-Invariant Feature Transform (SIFT)	9
2.2 Lokalisierung	15
2.2.1 Verfahren	16
2.2.2 Eigenschaften	21
2.2.3 State of the Art	22
2.3 Schätzung und Vorhersage von Zuständen	28
2.3.1 Wahrscheinlichkeit	28
2.3.2 Markov-Kette	31
2.3.3 Bayes-Filter	31
2.3.4 Partikel Filter	33
3 Analyse	37
3.1 Vergleich der Technologien für eine Lokalisierung	37
3.2 Einsatzmöglichkeiten der Sensorik	38
3.3 Anforderungen	45
4 Konzept für die Lokalisierung	47
4.1 Bestimmung des aktuellen Gebäudes	47
4.2 Lokalisierung innerhalb eines Gebäude	48
4.2.1 Aufnahme der Basisdaten in der Offline Phase	48

4.2.2	Bestimmung einer Position in der Online Phase	50
4.2.3	Raumbasiert Lokalisierung	52
4.2.4	Dynamischer Zustandsschätzer: Partikel Filter	53
5	Visuelle Erkennung von Objekten	59
5.1	Einfluss auf die visuelle Erkennung	59
5.2	Objekterkennung: Nächste Nachbar Suche	61
5.3	Match-Scores	64
6	Systementwurf	65
6.1	Datenschicht	65
6.2	Lokalisierung	66
6.3	Visuelle Erkennung	67
7	Implementierung	70
7.1	Datenbank	71
7.1.1	Initialisierung & Konfiguration	71
7.1.2	Backup der Daten	71
7.2	Lokalisierung	72
7.2.1	Gebäude Check-In	72
7.2.2	Vertikale Positionsänderung	72
7.2.3	Bewegungsmessung	73
7.2.4	Aufnahme der Sensordaten und Erstellung von Fingerprints	74
7.2.5	Partikel-Filter	76
7.3	Visuelle Erkennung	79
7.3.1	Digitalisierung der Objekte	79
7.3.2	Objekterkennung: Nächste Nachbar Suche	81
8	Auswertung	87
8.1	Lokalisierung	87
8.2	Visuelle Erkennung	92
8.2.1	Vergleich der Matching Verfahren: Bruteforce vs. Stabile Matches	94
8.2.2	Qualität der Objekterkennung	96
8.2.3	Erkennung zwischen verschiedenen Objekten	99
9	Zusammenfassung und Ausblick	102
A	Anhang	I

A.1 Speicherung der Simulationsdaten	I
A.2 ER Datenmodell	II
A.3 Konfigurationsdateien	III
A.4 Tabellen zur Auswertung der Lokalisierung und der visuellen Erkennung .	V
A.5 Inhalt der CD	VII
Literatur	VIII
Internetquellen	XIV
Abkürzungsverzeichnis	XVII
Stichwortverzeichnis	XIX

Abbildungsverzeichnis

1	ARToolKit Marker.	7
2	QR-Code	8
3	Frame Marker des QCAR Frameworks.	8
4	The Invisible Train.	9
5	Lokale Merkmale als Punkte (rechts) und Kanten (links).	10
6	Zweidimensionale Gaußverteilung.	12
7	Erstellung eines DoG-Skalenraums durch Differenzbildung von zwei Bildern, die in der Gauß-Pyramide aufeinander folgen.	13
8	Bestimmung der Extrema durch Vergleich der 26 Nachbarn eines Pixels im DoG-Skalenraum.	13
9	String Marker und Usnap Plakat.	14
10	AR Marker vom Nintendo 3DS.	15
11	Time of Arrival.	17
12	Szenenanalyse eines Bildes.	20
13	Bild einer Funkzelle.	20
14	Einteilung der Welt in Längen- und Breitengrade.	24
15	Normalverteilung	30
16	Hidden Markov Modell.	32
17	Resampling	36
18	Veränderung des Luftdrucks innerhalb einer Minute und im Verlauf eines Tages.	40
19	Verhältnis des Luftdrucks und der Höhe mit der Barometrischen Höhenformel.	41
20	Zeigt die Auf- und Abwärtsbewegung beim langsamen gehen.	42
21	Ein Kompass gibt die aktuelle Richtung mit $\alpha = 40^\circ$ an. Dies entspricht der Richtung Nord.	44

22	Anordnung der Fingerprints (rote Punkte) auf einer Karte des TGS, Gebäude 2 in der dritten Etage.	50
23	Arithmetisches Mittel aller Fingerprints in jedem Raum.	53
24	Die Anzahl der Schlüsselpunkte ist abhängig von der Struktur des Objektes. Bildquelle: Pergamonmuseum Berlin - Museum für Islamische Kunst.	60
25	Aufnahme eines Objektes.	60
26	Symmetrie Test.	63
27	Komponenten der Datenmodells.	66
28	Komponenten der Lokalisierung.	66
29	Auf einer Karte wird eine Position mit einer Touch-Geste bestimmt (links). Dann kann eine Messung für verschiedene Richtungen an dieser Position gemacht werden.	67
30	Aufnahme von Bildern zu einem Objekt und Berechnung der Merkmale. .	68
31	Komponenten der Objekterkennung.	69
32	Zusammenspiel aller Komponenten.	69
33	Zustandsdiagramm. Auswahl des aktuellen Gebäudes.	72
34	Koordinatensysteme in Android.	73
35	Klassendiagramm RecordFingerprintActivity.	75
36	Klassendiagramm Service und Beans.	75
37	Sequenzdiagramm der Aufnahme der Fingerprints.	76
38	Klassendiagramme des Partikel-Filters.	77
39	Dieses Sequenzdiagramm zeigt die Verwendung eines Partikel-Filters und eines Services.	77
40	Screenshot: Gleichverteilung aller Partikel (grüne Punkte) während der Initialisierungsphase.	78
41	Screenshot: Partikel Wolke entspricht dem Ort der wahrscheinlichsten Position.	78
42	Berechnung der Schlüsselpunkte auf einem Bild.	81
43	Nächste Nachbar Suche der Schlüsselpunkte im Original- und Referenzbild.	82
44	Stabile Matches	84
45	Mittlere Genauigkeit (Summe aller Partikel) mit dem Partikel Filter in 500 Iterationen.	88

46	Grafische Darstellung der mittleren Genauigkeit in den einzelnen Räumen. Die Farben geben die mittlere Genauigkeit in diesem Raum an.	89
47	Gegenüberstellung von WLAN und Bluetooth bzgl. der Genauigkeit und der Präzision.	90
48	Zeigt den Energieverbrauch der verschiedenen Sensoren.	91
49	Einfache Objekte mit monotonen Farbverlauf.	92
50	Durchsichtige Objekte.	92
51	Gleiche Objekttypen mit unterschiedlich vielen Schlüsselpunkten.	93
52	Objekte mit vielen Farben und vielen starken Kontrasten.	93
53	Testbild.	94
54	Zeigt die Qualität der Objekterkennung für Versuch eins und zwei. Je größer die Anzahl der Schlüsselpunkte ist, desto höher ist der Anteil kor- rekt erkannter Objekte.	99
55	Korrekte Zuordnung der Schlüsselpunkte.	100
56	Beispiel für Ausreißer.	101
57	ER Modell 1	II
58	ER Modell 2	II

1 Einleitung

Diese Arbeit befasst sich mit der visuellen Erkennung von realen Objekten. Die visuelle Erkennung von Objekten ist überall da notwendig, wo Objekte nicht durch ID basierte Verfahren z.B. Barcodes erkannt werden können oder sollen. Ein Besucher eines Museum kann Informationen zu verschiedenen Ausstellungsobjekten erhalten. Diese Informationen werden in Form von Texttafeln, welche neben den Objekten angebracht sind, dem Besucher zur Verfügung gestellt. Diese Art der Informationsvermittlung ist jedoch recht nüchtern, was nicht unbedingt alle Zielgruppen anspricht. Mit dem Einsatz von audio und visueller Technik kann die Darstellung von Informationen abwechslungsreicher gestaltet werden. Die Schwierigkeit ist jedoch, dass neue Technik nur schwer in ein bestehendes Konzept eines Museums integriert werden kann ohne von der Ausstellung abzulenken. Bereits existierende Systeme wie Audio-Guides sind ein erster Schritt zu einer interaktiven Museumsführung. Augmented Reality kann dem Besucher neue Formen des Lernens und Erlebens in einem Museum ermöglichen, indem die Technik mit der Ausstellung kombiniert wird. Die Grenzen der realen Ausstellung und der digitalen Informationen fließen ineinander über.

1.1 Motivation und Problemstellung

In den meisten Augmented Reality Anwendungen werden sogenannte Marker erkannt. Auf diesen Markern können dann z.B. digitale Objekte, Bilder oder textbasierte Informationen dargestellt werden. Der Vorteil der Marker ist, dass sie technisch gut erkennbar sind. Jedoch ist die Anbringung von Markern an Ausstellungsobjekten in einem Museum meist nicht erwünscht. Hier muss also nach Lösungen gesucht werden, wie Objekte erkannt werden können ohne diese mit Markern zu versehen. Dazu sollen geeignete Algorithmen und Verfahren erarbeitet werden. Ausserdem stellt sich die Frage inwieweit die Struktur und Form der Objekte die Erkennung beeinflusst. Oft nehmen Verfahren zur Bildererkennung, je nach Menge der Bilder, viel Zeit in Anspruch.

Die meisten mobilen Geräten sind mittlerweile für eine Augmented Reality Anwendung

geeignet. Zusätzlich verfügen die Geräte über verschiedene Schnittstellen und Sensoren. Die Frage ist, ob man diese Sensoren verwenden kann um eine Erkennung der Objekte zu ermöglichen. Da eine Lokalisierung eines jeden Objektes sehr aufwendig wäre, sollen die Objekte indirekt lokalisiert werden. Bestimmte Objekte, wie z.B. die Ausstellungsobjekte in einem Museum werden nur sehr selten bewegt. Durch eine Zuordnung der Objekte zu einem Raum, kann durch eine Lokalisierung des Benutzers eine indirekte Lokalisierung der Objekte erfolgen. Mit dem Wissen über den Aufenthaltsort, können dem Benutzer zusätzliche Informationen über den aktuellen Raum, eine Karte mit Wegbeschreibungen oder weitere Informationen zu bestimmten Objekten präsentiert werden. Die Lokalisierung kann dabei helfen den Zeitaufwand für eine Bilderkennung zu reduzieren, indem die mögliche Bildmenge für einen Vergleich eingeschränkt werden kann. Da GPS nicht innerhalb von Gebäuden funktioniert, müssen Lösungen für eine alternative Ortung erarbeitet werden.

1.2 Zielsetzung

Zuerst soll eine Lokalisierung einer Person innerhalb eines Gebäudes realisiert werden. Für die Lokalisierung sollen Technologien genutzt werden, die mit den meisten mobilen Geräten kompatibel sind um eine möglichst breite Nutzbarkeit zu ermöglichen. Ausserdem soll untersucht werden, wie integrierte Sensoren eines mobilen Gerätes für eine Lokalisierung genutzt werden können. Anschließend soll eine visuelle Erkennung von Objekten umgesetzt werden, indem ein aktuelles Bild eines Objektes mit einer gespeicherten Menge von Bildern zu den Objekten verglichen wird. Dazu sollen Lösungen erarbeitet werden, mit denen Objekte aufgenommen und erkannt werden können. Das Ziel dieser Arbeit ist es unter Verwendung verschiedener Sensoren, ein mobiles Gerät innerhalb eines Gebäudes zu lokalisieren, um anschließend Objekte visuell erkennen zu können. Diese Arbeit soll den Grundstein für eine Augmented Reality Anwendung legen. Vorrangig geht es um die Erarbeitung von Konzepten und Lösungsmöglichkeiten für eine Objekterkennung. Eine visuelle Darstellung von digitalen 3D Objekten, wie es in AR Anwendungen üblich ist, soll nicht das Ziel dieser Arbeit sein.

1.3 Struktur der Arbeit

Im nächsten Kapitel werden die Grundlagen dieser Arbeit beschrieben. Die Grundlagen beschreiben, welche Methoden der visuellen Erkennung in Augmented Reality Anwendungen genutzt werden können. Desweiteren werden verschiedene Methoden der Loka-

lisierung vorgestellt. Im dritten Kapitel wird analysiert, welche Sensoren für eine Lokalisierung nutzbar sind. Kapitel vier und fünf beschreiben die Konzepte der visuellen Erkennung und der Lokalisierung. Kapitel sechs erklärt den Aufbau des Systems und Kapitel sieben die Implementierung der einzelnen Komponenten. In Kapitel acht wird die Auswertung der Lokalisierung und der visuellen Erkennung vorgenommen.

In der Arbeit werden einige UML Klassendiagramme gezeigt, in denen bewusst auf eine umfangreiche Beschreibung der Parameter verzichtet worden ist, da die Klassendiagramme sonst sehr schnell unübersichtlich werden. Für einen detaillierten Einblick in die Klassen kann der Quellcode auf der CD eingesehen werden. Die Seitennummerierung im Anhang ist mit großen römischen Buchstaben formatiert. Literaturangaben auf Bücher und wissenschaftliche Veröffentlichungen werden in eckigen Klammern mit Großbuchstaben der Autoren und der Jahreszahl angegeben. Verweise auf online Quellen werden in eckigen Klammern mit einer fortlaufenden Nummerierung angegeben.

2 Grundlagen

In diesem Kapitel werden die Grundlagen für diese Arbeit besprochen. Zuerst wird ein kurzer Einblick in das Gebiet der Augmented Reality gegeben. Anschließend werden verschiedene Verfahren der visuellen Erkennung besprochen, wie sie in Augmented Reality Anwendungen vorkommen. Desweiteren gibt es einen Überblick, welche technischen Möglichkeiten für eine Lokalisierung von Objekten oder Personen existieren. Abschließend wird in diesem Kapitel auf Grundlagen der Wahrscheinlichkeitsrechnung eingegangen, welche für die Lokalisierung wichtig sein werden.

2.1 Augmented Reality

Augmented Reality (AR) beschreibt die erweiterte Wahrnehmung der Wirklichkeit. Mit AR können Bilder aus der Wirklichkeit durch Einblendung computergenerierter Informationen ergänzt bzw. überdeckt werden [WBW11]. Virtuelle Objekte, 2D oder 3D, verschmelzen mit der realen Welt. Die einfachste Form von AR ist das Einblenden eines 2D Bildes über ein digitales Video-Bild. Aufwendigere AR Anwendungen ermöglichen das einblenden von 3D Objekten im Kontext der realen Umgebung. Da virtuelle Objekte ohne weitere Hilfsmittel nicht sichtbar sind, benötigt AR ein Display auf dem die virtuellen Objekte dargestellt werden können [CF08].

Es existieren zwei Konzepte um AR zu realisieren, mittels einer magischen Linse oder eines magischen Spiegels. Beim Konzept des magischen Spiegels wird ein Monitor oder eine Projektionsfläche wie ein Spiegel verwendet. Der Benutzer steht vor dem Monitor und wird von einer Kamera erfasst. Auf dem Monitor sieht sich der Benutzer wie in einem Spiegel selbst. Zusätzlich werden auf oder um den Benutzer virtuelle Objekte eingeblendet. Das andere Konzept der magischen Linse ermöglicht es einem Benutzer mittels eines Displays auf die reale Welt zu schauen, welche durch virtuelle Objekte erweitert wird. Ein Display kann z.B. ein Computer-Monitor, ein Smartphone oder ein head-mounted display (HMD) sein. Bei Monitoren und Smartphones, wird die reale Umgebung durch eine Kamera erfasst. Die virtuellen Objekte werden von einem Szenengenerator eingefügt

und abhängig von der Ausrichtung des verwendeten Gerätes rotiert und skaliert [Bun10]. Bei HMD schaut der Benutzer durch ein Display, welches das Licht von außen teilweise durchlässt. Gleichzeitig werden Projektionen auf das Display dargestellt. Diese doppelte Eigenschaft des Displays funktioniert dadurch, dass nur bestimmte Wellenlängen von außen durchgelassen werden. Dadurch nimmt der Benutzer die reale Umgebung etwas dunkler wahr, wenn er durch das Display schaut. Ein sogenannter Head-Tracker erkennt die Position und die Lage des Kopfes. Mit diesen Informationen können die virtuellen Objekte rotiert und skaliert werden. Eine andere Variante von HMD ist das Virtual Retinal Display (dt. virtuelle Netzhautanzeige). Ähnlich wie ein Beamer mit Licht ein Bild auf eine Wand projiziert, wird beim Virtual Retinal Display das Bild mit einem energiearmen Laser in den Farben Rot, Grün und Blau, direkt auf die Netzhaut gezeichnet [Fur]. Diese Technik ermöglicht, bei nur geringem Energieverbrauch, eine hohe Helligkeit, starke Kontraste und Tiefenschärfe.

Augmented Reality ist nicht auf die Erweiterung durch visuelle Informationen beschränkt [Azu97]. Sinne wie das Hören, Fühlen und Riechen können ebenfalls angesprochen werden. Neben der Augmented Reality gibt es noch die Augmented Virtuality. Während in der Augmented Reality die reale Welt um virtuelle Objekte erweitert wird, wird in der Augmented Virtuality die virtuelle Welt um reale Objekte erweitert. Azuma et al. [ABB⁺01] haben für Augmented Reality folgende Kriterien aufgestellt:

- Kombination realer und virtueller Objekte,
- Interaktivität und Echtzeit,
- Registrierung im 3D Raum.

Mit der Anforderung der Interaktivität und Echtzeit sind mit Medien wie Videos und Filme keine Augmented Reality Anwendungen möglich. Zwar kombinieren Filme reale und virtuelle Objekte, sie sind aber nicht interaktiv. Die Registrierung im 3D Raum bedeutet, dass ein virtuelles Objekt an die reale Welt an einem bestimmten Punkt gebunden ist. Entfernt man sich von diesem Punkt wird das virtuelle Objekt kleiner bzw. größer, wenn man sich ihm nähert. Wenn man sich um den Punkt bewegt, an dem sich das virtuelle Objekt befindet, wird auch das virtuelle Objekt entsprechend gedreht. Dadurch entsteht der Eindruck, dass ein virtuelles Objekt fest in der realen Welt verankert ist. Streng genommen schließt diese Anforderung einfache Einblendungen von Informationen aus. Jedoch ist die Anforderung Registrierung im 3D Raum nicht in allen Definitionen ein Kriterium für Augmented Reality [1]. Oft wird lediglich von einer Erweiterung der

Realität durch Zusatzinformationen gesprochen. Welcher Art diese Informationen sein können ist häufig nicht spezifiziert.

2.1.1 Markerbasierte Erkennung

AR Anwendungen benötigen einen eindeutigen Punkt, an dem virtuelle Bilder oder Objekte in einem digitalen Bild eingefügt werden können. Die meisten Anwendungen verwenden dazu AR Marker. Ein Marker beschreibt ein eindeutiges Muster, das von der AR Software erkannt werden kann [CF08]. Der Marker wird verwendet, um die Entfernung und den Blickwinkel zur Kamera zu bestimmen und somit das virtuelle Objekt korrekt darstellen zu können. Owen et al. [OXM02] haben Kriterien für Marker definiert, wie diese am besten erkannt werden können. Im Folgenden werden die Anforderungen an die Form, die Farbe sowie die Erkennung der Marker beschrieben.

Form

Da Marker von optischen Sensoren, z.B. einer Kamera, erkannt werden müssen, ist es wichtig, dass sie aus allen Richtungen gut erkennbar sind. Daher sollte die Form eines Markers gleichmäßig sein z.B. quadratisch. Damit die Position und die Ausrichtung eines Markers relativ zur Kamera bestimmt werden kann, müssen die Koordinaten von mindestens vier Punkten bekannt sein. Sind die vier Punkte des Markers bekannt, also auch die Entfernung der Punkte zueinander, kann aus dem Kamerabild die Position und Ausrichtung des Markers berechnet werden [Bun10, S. 13]. Dies ist wichtig um virtuelle Objekte korrekt auf den Markern abbilden zu können.

Farbe

Da Marker gut zu erkennen sein sollen, muss bei der Verwendung von Farben, die farbliche Auflösung der Kamera berücksichtigt werden. Je nach Lichtverhältnissen wird die Erkennung von ähnlichen Farben schwieriger. Außerdem erhöht die Verwendung von farblichen Bildern den Speicherbedarf und die Zeit, die für die Analyse des Bildes benötigt wird. Der Kontrast zwischen dem Marker und der Umgebung sollte möglichst groß sein. Schwarz-weiß Marker bieten einen guten Kontrast zur Umgebung. Der Hintergrund des Markers sollte Weiß und von einem schwarzen Quadrat eingeschlossen sein (siehe Abbildung 1) [Bun10, S. 13].

Erkennung

Die Erkennung eines Markers kann entweder über ein Symbol oder ein Bild erfolgen, wie in Abbildung 1 zu sehen ist [Bun10, S. 14]. Bei einem Marker wie in Abbildung 1 wird

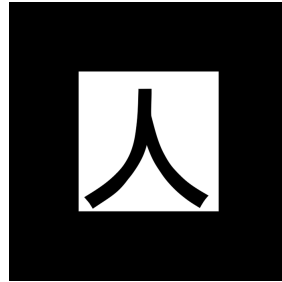


Abbildung 1: ARToolKit Marker [2].

nach einem Symbol gesucht, das von einem schwarzen Rand eingeschlossen ist. Dann wird das innere Bild in die Perspektive der Kamera transformiert und anschließend die Ausrichtung des Markers bestimmt. Die Ausrichtung wird über die Lage des Symbols erkannt. Die Erkennung eines Markers mit einem Referenzbild erfolgt mit Hilfe eines Korrelationskoeffizienten. Dabei wird ein Raster sowohl über den Marker als auch über das Referenzbild gelegt. Anschließend werden alle Rasterpunkte beider Bilder miteinander verglichen und die Grauwerte ausgelesen. Die Differenz der beiden Grauwerte am selben Rasterpunkt wird gespeichert. Für absolut identische Bilder ergibt der Korrelationskoeffizient Eins. Zwei maximal verschiedene Bilder (Schwarz und Weiß) haben einen Korrelationskoeffizienten von Null. Dieser Prozess wird mit allen vorhandenen Referenzbildern wiederholt. Das Bild mit dem höchsten Korrelationskoeffizienten sollte dann dem gesuchten Marker entsprechen. Wichtig ist das die vorhandenen Referenzbilder zueinander einen möglichst niedrigen Korrelationskoeffizienten aufweisen, um falsche Zuordnungen zu vermeiden.

Neben dem Marker aus Abbildung 1 gibt es noch andere Formen von Markern. Eine sehr weit verbreitete Art von Markern sind QR-Codes¹. Ein QR-Code ist ein zweidimensionaler Code, welcher aus einer quadratischen Matrix mit weißen und schwarzen Punkten besteht (siehe Abbildung 2). QR-Codes wurden von der japanischen Firma Denso Wave 1994 entwickelt, können aber lizenz- und kostenfrei verwendet werden [3, 4]. QR-Codes

¹QR steht für „Quick Response“ dt. „schnelle Antwort“.

wurden hauptsächlich für die Kodierung von Informationen entwickelt. In einem QR-Code können maximal bis zu 4296 Zeichen kodiert werden. Das Framework QCAR von



Abbildung 2: QR-Code [3]

Qualcomm [5] verwendet sogenannte Frame Marker (siehe Abbildung 3) [Bun11, S. 10]. Diese bestehen aus einem schwarzen Rand. An jeder der vier Innenseiten des Frame Markers sind neun Rechtecke angebracht, wodurch die Frame Marker erkannt werden. Jedes Rechteck entspricht einer Kodierung von einem Bit. Somit stehen 9 Bit pro Seite zur Verfügung. Diese 9 Bit Kodierung erlaubt bis zu 512 Frame Marker zu definieren. Die Kodierung ist an jeder Seite unterschiedlich, um auch die Ausrichtung (oben, unten, links und rechts) des Frame Markers bestimmen zu können. Das ist wichtig, um z.B. 3D-Objekte korrekt auf diesem Marker ausrichten zu können. Der innere Bereich des Frame Markers ist für die Erkennung unbedeutend und kann mit beliebigem Inhalt gefüllt werden. Ausgenommen sind die Farben Schwarz und Weiß, da sonst die Kodierung nicht mehr erkennbar ist. Im Gegensatz zu QR-Codes haben Frame Marker den Vorteil, dass die Kodierung kaum auffällt.

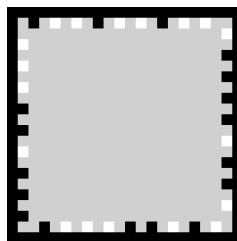


Abbildung 3: Frame Marker des QCAR Frameworks [5].

Ein Anwendung, die Marker verwendet, ist The Invisible Train (dt. der unsichtbare Zug). Dies ist ein multiplayer Spiel, bei dem ein virtueller Zug auf einer realen Holz-

schiene dargestellt wird. Über ein mobiles Gerät, welches mit einer Kamera ausgestattet ist, kann der Zug betrachtet werden. Verschiedene Marker sind über das Spielfeld verteilt [6]. Die Marker dienen als Orientierungspunkte, um den Zug korrekt über die Schienen zu leiten (siehe Abbildung 4). Die Spieler können den Zug beschleunigen oder die virtuellen Weichen stellen.



Abbildung 4: The Invisible Train [6].

Ein Marker muss nicht unbedingt ein Bild sein. Marker können auch elektronisch z.B. eine LED sein. Soll auf die Verwendung von Markern verzichtet werden (markerloses Tracking), muss versucht werden Merkmale direkt aus einem beliebigen Bild zu berechnen, welche dann für die Erkennung verwendet werden können. Ein solches Verfahren, das Merkmale in Bildern berechnen kann, ist das Scale-Invariant Feature Transform (SIFT) Verfahren, das im nächsten Abschnitt erläutert wird.

2.1.2 Markerlose Erkennung: Scale-Invariant Feature Transform (SIFT)

Während beim markerbasiertem Verfahren nach bestimmten Formen und Mustern gesucht wird, werden bei einem markerlosen Verfahren besondere Merkmale auf dem Bild berechnet, welche dann für eine Erkennung genutzt werden. Eine markerlose Erkennung bedeutet nicht, dass keine Marker verwendet werden, sondern dass natürliche Merkmale aus einem Bild berechnet werden. Das Bild kann beliebigen Inhalt darstellen und muss kein konkretes Muster wie bei einem gewöhnlichen Marker repräsentieren. Ein Verfahren, das lokale Merkmale aus einem Bild extrahiert, ist das SIFT Verfahren. Dieses wurde von Lowe in [Low99, Low03] vorgestellt. Die Merkmale eines Bildes dienen zum Vergleich mit Merkmalen anderer Bilder. Sie sind unveränderlich (invariant) gegenüber Veränderungen in der Skalierung und Drehung und teilweise invariant gegenüber der Beleuchtung und

dem Blickwinkel der Kamera. Die Invarianz gegenüber Skalierung ist nur durch die Auflösung der Kamera beschränkt. Mit dem SIFT Verfahren können ausschließlich schwarz-weiß Bilder verarbeitet werden. Da eine ausführliche Beschreibung des Verfahrens den Rahmen dieser Arbeit sprengen würde, soll im folgenden lediglich ein allgemeiner Einblick in die Funktionsweise von SIFT gegeben werden. Weitere Informationen über das SIFT Verfahren finden sich in den bereits genannten Arbeiten von Lowe oder Prestle [Pre07, Seite 75 ff].

Lokale vs. globale Merkmale

Lokale Merkmale (engl. *local features*) sind Bildmuster, die sich aus ihrer direkten Umgebung abheben [TM08]. Sie betrachten bestimmte Bildeigenschaften wie Farben und Farbverteilungen, Feinheiten von Strukturen in Texturen, besondere Punkte wie Kanten oder Ecken, aber auch die Beleuchtung bestimmter Bildbereiche [Pre07, S. 50f]. Wie in Abbildung 5 zu sehen ist, können lokale Merkmale aus Punkten, aus Kanten oder Bildbereiche bestehen. Lokale Merkmale können überall dort gefunden werden, wo sich die Intensität bestimmter Eigenschaften z.B. Farben, Beleuchtung oder Strukturen in einem Bild ändert.



Abbildung 5: Lokale Merkmale als Punkte (rechts) und Kanten (links)[TM08].

Während lokale Merkmale Teile eines Bildes beschreiben, beschreiben globale Merkmale (engl. *global features*) ein Bild insgesamt. Ein Beispiel für ein globales Merkmal ist z.B. ein Histogramm eines Bildes. Ein Histogramm stellt die Häufigkeitsverteilung von bestimmten Merkmalen dar. Es besteht aus mehreren Bins (dt. Kasten). In einem Bin wird die Information, wie oft ein Merkmal vorkommt, abgelegt. Ein Bild besteht aus mehre-

ren Pixeln². Jedes Pixel hat einen eigenen Farbwert. Ein Bin eines Histogramms kann z.B. für einen konkreten Farbwert oder auch für einen Farbbereich stehen. Jedes Pixel wird entsprechend seiner Farbe einem Bin zugeordnet. Nun kann aus dem Histogramm die Anzahl der verschiedenen Farben eines Bildes abgelesen werden. Theoretisch kann man sagen, dass zwei Bilder mit exakt dem selben Histogramm identisch sind, da sie die selben Pixel und Farben darstellen. Jedoch wird dabei die Anordnung der Pixel in einem Bild komplett ignoriert. Ein Problem mit Histogrammen als globales Merkmal ist, wenn das Objekt in dem Bild kleiner bzw. größer dargestellt wird, die Pixelverteilung sich komplett ändert. Somit ist ein Histogramm als lokales Merkmal zu einem Bild nicht geeignet. Lokale Punktmerkmale werden im Folgenden als Schlüsselpunkte bezeichnet.

Berechnung der Schlüsselpunkte

Zuerst müssen mögliche Schlüsselpunkt-Kandidaten lokalisiert werden. Diese werden anhand bestimmter Merkmale identifiziert, wie z.B. das Vergleichen der Intensität eines Pixels mit seinen unmittelbaren Nachbarn. Diese Merkmale müssen in verschiedenen Auflösungen des Bildes stabil bleiben. Dazu wird zuerst ein Skalenraum erstellt. Der Skalenraum $g(x, \xi)$ ist eine Datenstruktur, welche aus einer Folge eines Bild $g(x)$ mit unterschiedlichen Auflösungen besteht. [Jäh05, S. 152]. Die aktuelle Auflösung des Bildes wird mit ξ bezeichnet. Der Skalenraum ermöglicht eine feine Skaleneinteilung durch einen kontinuierlichen Skalenparameter. Ein Skalenraum kann z.B. durch Faltung des Bildes $I(x, y)$ mit einem Gauß-Filter $G(x, y, \sigma)$, mit variabler Standardabweichung $\sigma \in \mathbb{R}, \sigma > 0$ erstellt werden [Low03, S. 5].

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (2.1)$$

Dieser Skalenraum wird dann als Gauß'scher Skalenraum $L(x, y, \sigma)$ bezeichnet. Der Gauß-Filter ist eine zweidimensionale Gaußfunktion³, die wie folgt definiert:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (2.2)$$

Der Gauß-Filter ist ein Glättungs-Filter, wodurch das Rauschen in einem Bild vermindert werden kann [7], bei dem gröbere Strukturen jedoch erhalten bleiben. Bei der Glättung fließen zu jedem Bildpunkt die Werte der benachbarten Bildpunkte mit in die Berechnung ein. Die benachbarten Bildpunkte werden mit der zweidimensionalen Gaußglocke

²Pixel: Abkürzung des englischen Begriffes *picture element*.

³Die Gaußfunktion wird im Abschnitt 2.3.1 ab Seite 29 näher thematisiert.

(siehe Abbildung 6) gewichtet. Je höher die Standardabweichung σ ist, desto flacher wird die Gaußglocke und desto mehr benachbarte Bildpunkte gehen mit zunehmenden Gewicht in die Glättung ein [Pre07].

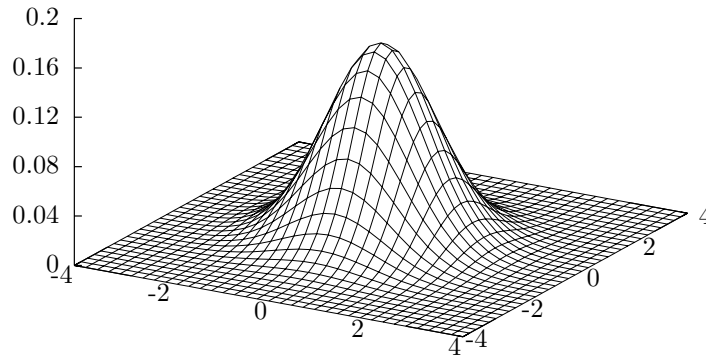


Abbildung 6: Zweidimensionale Gaußverteilung, $\sigma = 1$.

Um stabile Schlüsselpunkte zu bestimmen wird das Bild in verschiedenen Skalierungen übereinander gelegt und jede Ebene mit dem Gauß-Filter um einen konstanten Faktor k gefaltet. Die Differenz zwischen zwei benachbarten Ebenen wird mit der Difference of Gaussian (DoG) Funktion bestimmt, wodurch ein DoG-Skalenraum entsteht. Der DoG-Skalenraum wird durch die Differenz zweier Bilder, die in der Gauß-Pyramide aufeinander folgen, gebildet (siehe Abbildung 7) [Low03, S. 5]. Die Extremstellen aus dem DoG-Skalenraum sind die Schlüsselpunkt-Kandidaten. Die Extrema werden durch Vergleich der Intensität eines Pixels mit seinen 26 Nachbarn (8 benachbarte Punkte auf gleicher Ebene, je 9 benachbarte Punkte auf der Ebene darüber und darunter) im DoG-Skalenraum bestimmt, wie in Abbildung 8 verdeutlicht wird. Ein Extremwert liegt vor, wenn die Intensität des Pixels größer oder kleiner als die Intensität der Pixel all seiner Nachbarn ist. Wenn dies der Fall ist, wird der Punkt mit seinen x- und y-Koordinaten sowie der Standardabweichung σ als Schlüsselpunkt-Kandidat zur weiteren Verarbeitung gespeichert [Low03, S. 7f]. Anschließend werden die ermittelten Schlüsselpunkt-Kandidaten auf ihre Stabilität untersucht. Schlüsselpunkt-Kandidaten mit schwachen Extremwerten werden verworfen. Alle übrigen Kandidaten sind stabile Schlüsselpunkte.

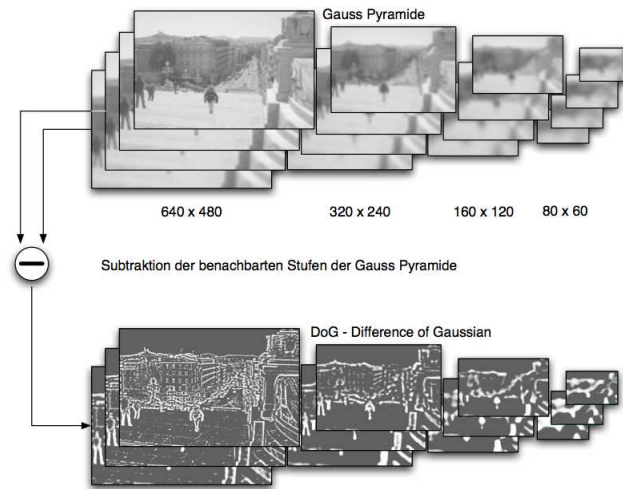


Abbildung 7: Erstellung eines DoG-Skalenraums durch Differenzbildung von zwei Bildern, die in der Gauß-Pyramide aufeinander folgen [Hey05, S. 12].

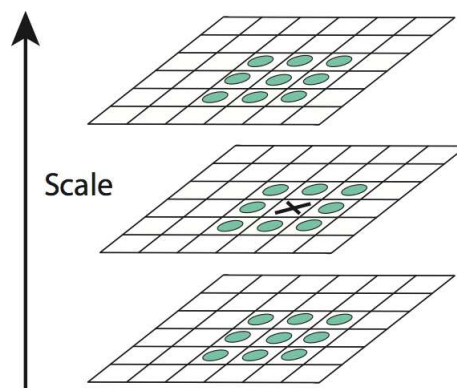


Abbildung 8: Bestimmung der Extrema durch Vergleich der 26 Nachbarn eines Pixels im DoG-Skalenraum [Low03].

Jedem stabilen Schlüsselpunkt wird eine Orientierung zugewiesen, wodurch die Schlüsselpunkte invariant gegenüber einer Drehung werden.

Deskriptoren

Ein Deskriptor (engl. descriptor; abgeleitet von „to describe“ dt. „beschreiben“) ist ein Vektor von Merkmalen, welcher die Umgebung vom Schlüsselpunkt beschreibt. Mit Hilfe

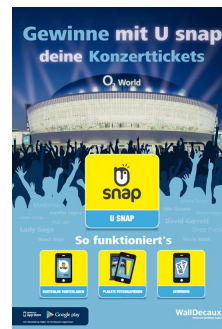
von Deskriptoren können zwei Schlüsselpunkte auf ihre Ähnlichkeit untersucht werden. Wie genau mit Hilfe von Merkmalen Objekte in Bildern erkannt werden können, wird in Abschnitt 5.2 ab Seite 61 beschrieben.

State of the Art

Im folgenden sollen drei Anwendungen gezeigt werden, welche ein markerloses Verfahren einsetzen.



(a) String Marker [8]



(b) Usnap Plakat [9]

Abbildung 9: String Marker und Usnap Plakat.

Die britische Firma String Labs Ltd. [8] hat ein Verfahren entwickelt, das markerbasierte mit markerloser Erkennung kombiniert. Außen müssen die Marker über einen schwarzen Rand verfügen, während im Innern ein beliebiges Bild dargestellt werden kann (siehe Abbildung 9(a)) [Bun11, S. 9]. Somit muss eine Berechnung der Merkmale nur dann durchgeführt werden, wenn ein Marker mit schwarzem Rand erkannt worden ist. Dies ist durchaus sinnvoll, da eine Erkennung von Kanten weniger aufwändig ist als die Berechnung von Merkmalen.

Nintendo hat seiner Spielekonsole 3DS eigene Marker beigelegt (Abbildung 10). Zusammen mit der Spielekonsole können mit den Markern interaktive Spiele gespielt werden. Da die Marker von Nintendo ein Bild zeigen, werden hier wahrscheinlich natürliche Merkmale aus einem Bild extrahiert.

Mit der Anwendung Usnap⁴ können bestimmte Plakate erkannt werden (siehe Abbildung 9(b)). Da die Plakate ein beliebiges Bild zeigen, ist dies ein markerloses Verfahren.

⁴Usnap steht im Android Market und in Apples App Store zur Verfügung.



Abbildung 10: AR Marker vom Nintendo 3DS.

So können die Benutzer direkt mit der Werbung interagieren.

2.2 Lokalisierung

In diesem Kapitel sollen die Grundlagen einer Lokalisierung besprochen werden. Das Wort Lokalisierung leitet sich aus dem lateinischen Wort „lokus“ (dt. Ort) ab und bedeutet allgemein die Feststellung einer Position [10] bzw. den „Ort zu bestimmen, an dem sich etwas befindet, bzw. einen bestimmten Punkt oder Bereich einzugrenzen“ [Bro86]. Synonym wird statt Lokalisierung oft von Ortung, Ortsbestimmung oder Positionsbestimmung gesprochen. Wobei mit Ortsbestimmung nach [dtv99] eine „Festlegung der geographischen Länge und Breite eines Ortes auf der Erdoberfläche mit Hilfe von Gestirnen oder durch Funkortung“ gemeint ist. Der Begriff „Ortung“ wird auch als „Ortsbestimmung einer Person oder eines Gegenstandes“ sowie „Positionsbestimmung eines Land-, See-, Luft- oder Raumfahrzeugs“ oder „die Ermittlung von Ort, Bewegungszustand (Geschwindigkeit, Beschleunigung) und Bahn eines Fahrzeugs“ beschrieben [Bro86]. Für eine Ortung sind besondere Messungen erforderlich. Werden alle Messungen für die Ortung von dem zu ortenden Objekt selbst vorgenommen, so spricht man von Eigen-Ortung (engl. positioning). Im Gegensatz dazu werden bei der Fremd-Ortung (engl. tracking) die Messungen außerhalb des zu ortenden Objektes gemacht [Rot05].

Die Lokalisierung von Personen oder Objekten ist heutzutage von sehr großer Bedeutung. In vielen Bereichen wie z.B. der Ortung von Waren in einem Lager, über Positionsbestimmung von Autos und Flugzeugen, bis hin zur Navigation in Städten und zur

Unterhaltung in Spielen werden Methoden der Lokalisierung eingesetzt.

Verfahren zur Lokalisierung können in drei Kategorien eingeteilt werden [KHKE05]: Positionierungs- und Ortungssysteme sowie Systeme, welche die Orientierung des Gerätes erkennen. Ein Positionierungssystem liefert eine absolute Position. Um eine Position absolut zu beschreiben, stehen verschiedene Koordinatensysteme, wie geographische, geozentrische oder kartesische Koordinatensystem zur Verfügung [S. 274][Rot05]. Im Gegensatz zu Positionierungssystemen liefern Ortungssysteme keine absolute Position, sondern bestimmen die Umgebung, bzw. einen symbolischen Ort, an dem sich ein gesuchtes Objekt befindet. Ein symbolischer Ort kann z.B. ein Gebäude oder eine Raumnummer sein. Jedoch wird nicht immer genau zwischen einer absoluten Position und einer symbolischen Position unterschieden, sodass sie häufig fälschlicherweise synonym verwendet werden. Wird keine absolute Position benötigt, kann eine relative Position bezogen auf einen bestimmten Punkt angegeben werden (300m vom Bahnhof entfernt). In einem Gebäude macht es oft mehr Sinn eine Etage durch eine symbolische Position zu beschreiben (vierte Etage), als eine absolute Position anzugeben (Ursprung Eingangstür, davon 5m nördlich und 18m nach oben). Darüber hinaus ist wichtig zu wissen, dass ein System zur Positionierung nicht die exakte Position liefern muss. Aufgrund ungenauer Messwerte kann die berechnete Position von der realen Position abweichen.

2.2.1 Verfahren

In diesem Abschnitt sollen die verschiedenen Verfahren erläutert werden, mit denen eine Position oder eine Entfernung zu einem Punkt berechnet werden kann.

Lateration

Bei der Lateration wird die Position eines Objekts durch Messungen der Entfernung zu mehreren Referenzpunkten bestimmt [HB01]. Zur Bestimmung der Position in einem zweidimensionalen Koordinatensystem werden die Entfernungen zu mindestens drei nicht kollinearen⁵ Punkten benötigt. Zur Messung der Entfernung zu einem Objekt können zeitbasierte Verfahren verwendet werden. Dabei wird die Entfernung aus der Zeit, die ein Signal von einem Punkt zum anderen benötigt, berechnet. Es wird zwischen Verfahren unterschieden, welche die Ankunftszeit messen (Time of Arrival (ToA)) und Verfahren, bei denen die Zeitdifferenz der Signale gemessen wird (Time Difference of Arrival (TDoA)) [GL02] [Rot05, S. 277].

⁵Kollineare Punkte sind Punkte, die auf einer Geraden liegen.

- Das ToA Verfahren (auch als Time of Flight bezeichnet) misst die Zeit, die ein Signal mit einer bekannten Geschwindigkeit von einer Quelle zu einem Empfänger benötigt [Wid10] [11]. Verschiedene Arten von Signalen breiten sich mit unterschiedlicher Geschwindigkeit aus. Ein akustisches Signal (Schallwelle) hat in der Luft bei einer Raumtemperatur von 21°C eine Geschwindigkeit von ca. 344 Meter pro Sekunde. Dagegen breitet sich im Vakuum ein elektromagnetisches (EM) Signal mit Lichtgeschwindigkeit aus ($c = 299792458 \frac{m}{s}$). Die zurückgelegte Distanz d kann für EM Signale mittels der Lichtgeschwindigkeit c folgendermaßen berechnet werden:

$$d = c * (t_i - t_0) \quad (2.3)$$

t_i ist der Zeitpunkt der Ausstrahlung des Signals an der Quelle i und t_0 der Zeitpunkt des Empfangs [SSTa08, S. 41]. Durch lösen der nichtlinearen Gleichungssysteme zu drei Basisstationen kann eine Position bestimmt werden. Die berechnete Position ist der Schnittpunkt der Kreisflächen (siehe Abbildung 11). Die Mittelpunkte der Kreisflächen bilden die Basisstationen (Berlin, Hamburg, Erfurt), welche die Signale ausgesendet haben. Der Radius der Kreisflächen ist die berechnete Entfernung zu den Punkten. Bei Empfang eines einzigen Signals, liegt die mögliche Position irgendwo auf dem Umfang des Kreises. Durch den Schnitt zweier Kreisflächen ergeben sich schon zwei mögliche Positionen. Erst der Schnittpunkt zu drei Kreisflächen ergibt die gesuchte Position.

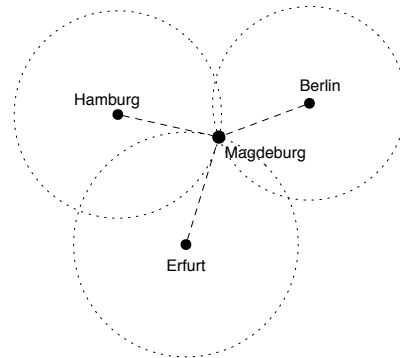


Abbildung 11: Berechnung des aktuellen Standortes durch Kenntnis der Entfernung zu drei Referenzpunkten.

Aufgrund der sehr hohen Ausbreitungsgeschwindigkeit von EM Signalen werden

für kurze Distanzen sehr genaue Uhren benötigt, um aus den Signallaufzeiten die Entfernung zum Sender berechnen zu können. Soll z.B. die Entfernung auf einen Meter genau bestimmt werden, müssen die Uhren bis zu 3.3 Nanosekunden genau gehen. Darüber hinaus muss dem Empfänger die genaue Zeit bekannt sein, zu der das Signal ausgesendet wurde. Eine andere Möglichkeit wäre das Signal als Echo zurückzusenden. Da das Signal den doppelten Weg zurückgelegt hat, ist die Entfernung aus der halben Laufzeit zu berechnen.

In der Regel breiten sich Signale gleichmäßig im Raum aus. Durch bestimmte Einflüsse z.B. durch Reflexion an Objekten kann die gleichmäßige Ausbreitung von Signalen verhindert werden. Dies kann zu falschen Entfernungsberechnungen führen. Durch Aggregation verschiedener Messungen sowie durch Kenntnisse der Umgebung ist es möglich falsche Messwerte heraus zu filtern, wie es z.B. beim Active Bat Verfahren (siehe Seite 22) gemacht wird. Time of Arrival wird z.B. im GPS System (siehe Seite 23) verwendet, um die Entfernung vom Empfänger zu den Satelliten zu bestimmen [PCB00]. Flugzeuge erzeugen ein elektromagnetisches Signal, welches vom Boden reflektiert wird um die aktuelle Höhe zu ermitteln. Roboter verwenden Ultraschall-Signale um die Entfernung zu einem Hindernis zu berechnen.

- Das TDoA Verfahren misst die Zeitunterschiede, die ein Signal benötigt um zu verschiedenen Empfängern zu gelangen [Wid10] [12]. Alle Punkte, deren Differenz der Abstände zu zwei Punkten (Brennpunkte) gleich ist, bilden eine Hyperbel [13]. Der Schnitt zweier Hyperbeln ist die gesuchte Position. Auch bei diesem Verfahren werden sehr genaue Uhren benötigt.

Winkelbasiert

Bei winkelbasierten Verfahren wie Angle of Arrival (AoA) wird die Entfernung zu einem Punkt in einem zweidimensionalen Koordinatensystem durch Messung der Winkel an zwei Referenzpunkten bestimmt [HB01, GL02] [Rot05, S. 277]. Zusätzlich muss noch die Entfernung der beiden Referenzpunkte zueinander bekannt sein. Um den Winkel eines eintreffenden Signals zu bestimmen werden sehr genaue Antennen benötigt.

Signalstärke

Aus der empfangenen Signalstärke, auch Received Signal Strength Indicator (RSSI) genannt, kann auf die Entfernung zum Sender geschlossen werden. Voraussetzung ist,

dass die Sendeleistung eines Kommunikationsteilnehmers bekannt ist [Rot05, S. 277] [Zap06, Wid10]. Je weiter ein Signal von seiner Quelle entfernt ist, desto mehr wird es abgeschwächt (gedämpft). Die Dämpfung im freien, luftleeren Raum (Freiraumdämpfung) ist abhängig von der Frequenz f und dem Abstand zum Ursprung r [BHK⁺09, S. 197] [14]. Es gilt für die Freiraumdämpfung F , mit c für die Lichtgeschwindigkeit:

$$F = \left(\frac{4\pi \cdot f \cdot r}{c} \right)^2 \quad (2.4)$$

Innerhalb der Erdatmosphäre werden die Signale zusätzlich durch Luft (und den enthaltenen Gase wie z.B. Sauerstoff, Stickstoff und Wasserdampf) gedämpft. Außerdem werden die Signale durch andere Objekte (Menschen, Wände, Berge oder Täler) absorbiert oder reflektiert. Durch Reflexion an verschiedenen Objekten kann es zu einer Mehrwegausbreitung kommen, bei der die Signale den Empfänger zu unterschiedlichen Zeiten erreichen [GK09]. Die Bestimmung der Entfernung aus der Signalstärke kann entweder auf empirisch gewonnenen Daten oder auf mathematischen Modellen beruhen [BP00]. Die Signalstärke kann als Leistungspegel L (Level) angegeben werden. Die Einheit für den Leistungspegel ist Dezibel (dB). Mit dem Vorzeichen „+“ werden Verstärkungen und mit dem Vorzeichen „-“ werden Abschwächungen gekennzeichnet [GK09, S. 75]. Eine Übertragung von 0 dB bedeutet eine 1:1 Übertragung [15].

Szenenanalyse

Bei der Szenenanalyse werden bestimmte Merkmale (engl. features) aus einer Szene verwendet um eine Position zu bestimmen [HB01]. So kann z.B. die Form bzw. die Silhouette eines Objektes Aufschluss über den Ort des Betrachters geben (siehe Abbildung 12). Ein Vorteil der Szenenanalyse ist, dass die Position durch Beobachtungen der Merkmale bestimmt werden kann, ohne dass geometrische Winkel oder Entfernungen gemessen werden müssen. Veränderungen der Umgebung können dazu führen, dass Merkmale neu erstellt werden müssen. Eine Szene kann aus Bildern oder aus anderen messbaren Eigenschaften wie z.B. elektromagnetische Wellen bestehen. Das RADAR Projekt (siehe Seite 27) zeichnet die Signalstärken mehrerer Basisstationen auf und weist ihnen eine Position im Raum zu. Eine Positionsbestimmung erfolgt durch Vergleich der aktuellen mit den gespeicherten Signalstärken.



Abbildung 12: Szenenanalyse eines Bildes [HB01]. Die Silhouette z.B. der Berge, kann ein Hinweis auf den Aufenthaltsort des Betrachters geben.

Zellbasierte Ortung

Ein Mobiltelefon ist i.d.R mit der am nächstgelegenen Funkzelle (Basisstation) verbunden (siehe Abbildung 13). Wenn man weiß mit welcher Funkzelle ein Mobiltele-

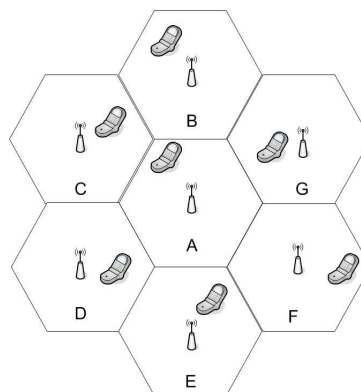


Abbildung 13: Mehrere Funkzellen. Die Position eines mobilen Gerätes wird durch die Nähe zur aktuellen Funkzelle bestimmt [16]

fon verbunden ist, kann man daraus das Wissen über den Ort des Mobiltelefons ableiten [HB01, GL02, Zap06] [Rot05, S. 277]. Dieses Verfahren wird auch Cell of Origin (CoO) genannt und funktioniert sowohl in GSM- und UMTS-Netzen als auch in WLAN-Infrastrukturen. Die Genauigkeit dieses Verfahrens ist jedoch an die Größe der Funkzelle gebunden [SSTa08, S. 43]. Je größer eine Funkzelle ist, desto geringer ist damit die Genauigkeit. Die Reichweite einer Zelle kann in Ballungsräumen, wo viele Basisstationen sehr dicht aufgestellt sind, bis zu 200 Meter betragen. Während in ländlichen Regionen,

wo nur sehr wenige Basisstationen aufgestellt sind, die Zelle eine Reichweite über 10 Kilometer haben kann.

2.2.2 Eigenschaften

Hightower et al. [HB01] haben verschiedene Eigenschaften für Lokalisierungssysteme zusammengefasst, welche im folgenden Abschnitt erläutert werden:

- **Physische oder symbolische Position**

Ein Lokalisierungssystem kann entweder eine physische oder eine symbolische Position angeben. Eine physische Position ist eine exakte Position in einem bestimmten Koordinatensystem. Eine symbolische Position ist eine abstrakte Beschreibung eines Ortes z.B. sein bestimmter Raum in einem Gebäude oder die Haltestelle eines Zuges. Systeme, die eine physische Position liefern, können natürlich auch eine symbolische Position angeben.

- **Absolute und relative Bezugsangaben**

Ein Lokalisierungssystem kann entweder absolute oder relative Bezugsangaben liefern. Systeme mit absoluter Bezugsangabe teilen sich ein gemeinsames Koordinatensystem und liefern eine Position absolut zum Koordinatenursprung. Relative Systeme liefern Positionsangaben relativ zur aktuellen Position. In relativen Systemen wird keine absolute Position geliefert, sondern es wird eine Aussage gemacht, wie weit das Objekt noch entfernt ist.

- **Genauigkeit und Präzision**

Die Genauigkeit gibt an, wie exakt eine berechnete Position mit der wirklichen Position übereinstimmt. Die Präzision beschreibt, wie oft diese Genauigkeit erreicht wird, d.h. wie zuverlässig die Lokalisierung ist.

- **Skalierung**

Die Skalierung bezieht sich auf verschiedene Eigenschaften eines Lokalisierungssystems. Auf die Reichweite in der Objekte lokalisiert werden können⁶. Die Anzahl der Objekte, die gleichzeitig lokalisiert werden können. Die Zeit, die für eine Lokalisierung benötigt wird. So kann z.B. GPS (siehe Seite 23) gleichzeitig von unbegrenzt vielen Empfänger genutzt werden. Andere Systeme wie z.B. Barcode-Lesegeräte können nur einen Barcode zum selben Zeitpunkt erfassen.

⁶Die Reichweite kann weltweit, innerhalb eines Stadtgebietes, in einem Gebäude oder innerhalb eines Raumes sein

- **Kosten**

Die Kosten für ein System können durch verschiedene Faktoren beeinflusst werden. Auf der einen Seite entstehen Kosten für die Installation, den laufenden Betrieb sowie Wartungsarbeiten. Andere Kosten entstehen durch benötigte Hardware (Sender, Empfänger, Verwaltungssysteme) oder durch das benötigte Personal.

- **Limitierungen**

Manche Systeme funktionieren nur in speziellen Umgebungen. So sind GPS Empfänger innerhalb von Gebäuden nicht fähig das Satellitensignal zu empfangen. Andere Systeme können nur Tags lesen, wenn keine anderen Tags in der Nähe sind.

2.2.3 State of the Art

In diesem Abschnitt werden verschiedene Systeme vorgestellt, die eine Lokalisierung umgesetzt.

Active Badge

Das Active Badge System verwendet Infrarot Beacons (dt. Leuchtfeuer) um Personen innerhalb von Gebäuden zu lokalisieren [WFG92] [Rot05, S. 295 ff.]. Ein Benutzer trägt einen kleinen Infrarot Sender, welcher in regelmäßigem Intervall diffuse Infrarotsignale aussendet. In den Infrarotsignalen ist eine eindeutige Benutzeridentifikation kodiert. Infrarotsensoren, die sich im Gebäude befinden, empfangen die Signale der Beacons und leiten diese an einen Rechner weiter. Da Infrarotsignale an Wänden reflektiert werden, breiten sich die Signale nicht in andere Räume aus. Durch die Reflexion an Wänden werden auch Signale empfangen, die nicht direkt auf einen Sensor treffen. Active Badge liefert eine symbolische Position, nämlich den konkreten Raum, in dem sich der Benutzer befindet [HB01]. Die Verwendung von Infrarot kann durch fluoreszierende Stoffe oder Sonnenlicht, welche selber Infrarot aussenden, beeinträchtigt werden. Diffuses Infrarotlicht hat nur eine Reichweite von wenige Metern und kann daher nur in kleinen bis mittelgroßen Räumen genutzt werden.

Active Bat

Das Active Bat (dt. Fledermaus) System verwendet Ultraschall zur Lokalisierung [WJH97, HHS⁺99] [Rot05, S. 299 f.]. Die Benutzer tragen sogenannte Bats. Nach Aufforderung eines Servers, senden die Bats ein kurzes Ultraschallsignal aus. Damit sich keine Kollisionen ergeben, wählt der Server immer genau ein bestimmtes Bat aus. Die Aufforderung

des Servers an das Bat werden per Funk übertragen. An der Decke eines Raums befinden sich Ultraschallempfänger, welche in einem Raster von 1,2 Meter angebracht sind. Wird ein Signal aufgefangen, wird dieses an den Location Server weitergegeben. Anhand der Laufzeiten, die das Signal vom Aussenden bis zu den Empfängern benötigt, kann vom Server eine Position des Bats berechnet werden. Active Bat benötigt eine große Infrastruktur von festen Empfängern, welche in einem Raster an der Zimmerdecke angebracht werden. Durch die Verwendung vieler aktiver Empfänger steigen die Kosten. Mit dem Active Bat System wurde eine Genauigkeit bis zu 9 Zentimeter erreicht [HB01].

Satellitenortung

Die Ortung durch Satelliten⁷ wurde ursprünglich zu militärischen Zwecken entwickelt [Bro86, DH04, Man09], [17]. Die Satellitenortung bietet den großen Vorteil der Navigation vor allem in der See- und Luftfahrt, da hier die Fahrzeuge mittels Autopilot automatisch gesteuert werden. Aber auch für Landfahrzeuge wird die Satellitenortung immer wichtiger. Satellitenortungssysteme wie das amerikanische Global Positioning System (GPS) bieten eine hohe Genauigkeit und eine weltweite Verfügbarkeit. Die Satelliten senden ihre genau Position auf ihrer Umlaufbahn zusammen mit der Weltzeit aus. Zur Berechnung der eigenen Position müssen die Entfernungen zu mindestens 3 Satelliten bekannt sein. Zur Berechnung der Entfernung zum Satellit wird die Laufzeit des Funksignals verwendet. Im Gegensatz zu den Satelliten untereinander, sind die Uhren der Empfänger nicht mit den Satelliten synchron. Jedoch kann der Empfänger ohne synchronisierte Uhren nicht bestimmen, wie lange das Signal vom Satelliten bis zum Empfänger unterwegs war. Daher werden die Uhren der Empfänger mit der Uhrzeit der Satelliten synchronisiert. Die Uhrzeit, wann das Signal vom Satelliten gesendet wurde, wird im Signal kodiert. Damit in allen Satelliten eine synchrone Uhrzeit vorhanden ist, sind alle Satelliten mit einer sehr genauen Cäsium Atomuhr ausgestattet, welche täglich mit den noch genaueren Atomuhren der US Air Force GPS Bodenstation synchronisiert werden. 24 Satelliten werden für eine globale Abdeckung benötigt. Zusätzlich sind noch weitere Satelliten zur Ausfallsicherheit im Orbit.

Eine GPS Position kann durch Länge (engl. longitude), Breite (engl. latitude) und die Höhe (engl. altitude) bestimmt werden. Für die Entfernungsmessung soll im folgenden jedoch nur die Länge und Breite verwendet werden. Zur Bestimmung einer Position ist die Erde in Längen und Breitengrade eingeteilt. Die Breitengrade haben alle einen kon-

⁷Ein Satellit ist ein „Begleiter eines Planeten“, „auf ellipt. oder kreisähn. Umlaufbahnen von Planeten und Monden gebrachter i.d.R. unbemannter Raumflugkörper“ [Bro86].



Abbildung 14: Einteilung der Welt in Längen- und Breitengrade [18].

stanten Abstand von 111.3km. Die Längengrade verringern ihren Abstand, ausgehend vom Äquator mit 111.3km, bis zu den Polen mit 0km. Eine Position wird dabei durch die Winkel der Längen und Breitengrade angegeben. Für die HTW am Standort Schöneweide wäre das $Breite = 52.45751^\circ$ und $Laenge = 13.52682^\circ$. Für kleine Areale, wie z.B. ein Stadtgebiet, kann die Entfernung mit dem Satz des Pythagoras berechnet werden. Geht man weiter davon aus, dass sich die Berechnung nur auf Mitteleuropa bezieht, kann ein durchschnittlicher Abstand von 71.5km zwischen den Längengraden angenommen werden [18]. Die Entfernung (d) zwischen zwei Punkten berechnet sich dann wie folgt:

$$\Delta x = 71.5 * (lon_1 - lon_2) \quad (2.5)$$

$$\Delta y = 111.3 * (lat_1 - lat_2) \quad (2.6)$$

$$d = \sqrt{\Delta x^2 + \Delta y^2} \quad (2.7)$$

Sollen größere Entfernungen bestimmt werden, muss beachtet werden, dass die Längen- und Breitenkreise kein rechtwinkliges Gitternetz bilden. Die Berechnung muss dann auf Basis einer Kugeloberfläche erfolgen. Für weitere Informationen siehe [18].

RFID

Radio Frequency Identification (RFID) [Fin08, S. 6 ff., S. 23 ff.] ist ein Übertragungsverfahren, bei dem Daten zwischen einem Transponder (im englischen auch *tag* genannt) und einem Lesegerät ausgetauscht werden. Auf dem internen Speicher des Transponders können zusätzliche Informationen gespeichert werden. RFID-Transponder können in drei Klassen eingeteilt werden: passiv, semi-passiv und aktiv.

- Passive Transponder besitzen keine eigene Stromversorgung. Die benötigte Energie für den Mikrochip wird durch die Antenne aus dem elektromagnetischen Feld des Lesegerätes entnommen. Passive Transponder können kein eigenes Signal aussenden. Zur Kommunikation wird das Feld des Lesegerätes (z.B. durch Lastmodulation oder modulierte Rückstreuung) beeinflusst. Da passive Transponder ohne Energiequelle ausgerüstet sind, sind die Anschaffungskosten sehr gering. Passive RFID tags ersetzen in vielen Bereichen die Barcode Technologie.
- Semi-Passive Transponder besitzen eine eigene Energiequelle. Dies kann z.B. eine Batterie oder eine Solarzelle sein. Die Energie wird jedoch nur für die Versorgung des Mikrochips benötigt. Semi-passive Transponder können, wie passive Transponder, kein eigenes Signal aussenden, sondern müssen zur Kommunikation das Signal des Lesegerätes modulieren.
- Aktive Transponder verfügen über eine eigene Energiequelle, welche zur Versorgung des Mikrochips und zur Erzeugung eines eigenen Signals dient. Da das vom Lesegerät empfangene elektromagnetische Feld nicht mehr zur Energieversorgung des Transponders benötigt wird, kann ein deutlich schwächeres Feld verwendet werden. Dies kann zu einer höheren Kommunikationsreichweite beitragen. Die Anschaffungskosten für aktive Transponder sind deutlich höher als die für passive Transponder.

RFID kann auf verschiedenen Sendefrequenzen arbeiten, welche den Bereichen LF (low frequency: 30 kHz - 300 kHz), HF (high frequency: 3 MHz - 30 MHz) sowie UHF (ultra high frequency: 300 MHz - 3 GHz) zugeordnet werden. Je nach Frequenz lassen sich Sendereichweiten von 1cm (close-coupling), bis 1m (remote-coupling) und über 1m (long-range) realisieren. Dabei ist die Sendereichweite von verschiedenen Faktoren abhängig: Von den Umweltbedingungen und von der Energie, die zum einem dem Lesegerät und zum anderen dem tag zur Verfügung steht.

LANDMARC

Ni et al. verfolgen mit dem Location Identification based on Dynamic Active RFID Calibration (LANDMARC) Verfahren eine Lokalisierung durch feste Referenz tags [NLLP03]. Die Position der Referenz tags ist bekannt. Das Verfahren benötigt die Signalstärken von jedem tag zu den Lesegeräten. Jedoch liefern RFID Systeme keine reinen Signalstärke-Werte. Das Lesegerät kann lediglich einen Power-Level (1 bis 8 in LANDMARC) registrieren. Da jedes Lesegerät mit einer 802.11b WLAN Schnittstelle ausgestattet ist,

können Informationen über die tags an einen Location Server gesendet werden. Mit Hilfe von Tracking tags können Personen oder Objekte lokalisiert werden. Dabei wird die Signalstärke der Tracking tags mit denen der Referenz tags verglichen. Zur Bestimmung der Position eines Tracking tags wird die Nähe zu den Referenz tags ermittelt. Das Referenz tag, welches dem Tracking tag am nächsten ist, wird als Position angenommen.

Zur Bestimmung der nächsten Nachbarn wird der euklidische Abstand eines jedes Tracking tags zu allen Referenz tags folgendermaßen ermittelt:

$$E_j = \sqrt{\sum_{i=1}^n (\theta_i - S_i)^2} \quad (2.8)$$

Dabei ist $\vec{S} = (S_1, S_2, \dots, S_n)$ ein Vektor mit Signalstärken des Tracking tags am Lesegerät i , mit $i \in [1, n]$ und $\vec{\theta} = (\theta_1, \theta_2, \dots, \theta_n)$ die Signalstärke eines Referenz tags am Lesegerät i . Zu jedem Tracking tag gibt es einen Vektor $\vec{E} = (E_1, E_2, \dots, E_m)$ der berechneten euklidischen Abstände zu jedem der m Referenz tags. Je näher ein Referenz tag zu einem Tracking tag ist, desto geringer sollte der euklidische Abstand sein. Ein Algorithmus kann nun durch Vergleich aller euklidischer Abstände an jedem Referenz tag die k -nächsten Nachbarn (engl. k -nearest neighbor) finden.

Da die genauen Positionen der Referenz tags bekannt sind, kann durch die k -nächsten Nachbarn eine Position (x, y) des tracking tags folgendermaßen bestimmt werden:

$$(x, y) = \sum_{i=1}^k (w_i)(x_i, y_i) \quad (2.9)$$

Dabei stellt w_i ein Gewichtungsfaktor zu dem i -ten nächsten Nachbarn dar. Die Autoren haben in Ihrem experimentellen Versuch 16 Referenz tags gleichmäßig im Abstand von einem Meter angeordnet. Sie fanden heraus, dass mit $k = 4$ für die Anzahl der nächsten Nachbarn die besten Ergebnisse erzielt wurden. Des Weiteren wurden Einflüsse der Umwelt untersucht, welche sich durch Personen im Raum ergaben. Hierbei wurde festgestellt, dass durch Personen im Raum die Lokalisierung kaum beeinflusst wird. Auch die Anzahl der Lesegeräte hat Einfluss auf die Lokalisierung. Je mehr Lesegeräte verwendet werden, desto mehr Daten können gesammelt werden. Bedingt durch den hohen Preis wurden jedoch nicht mehr als vier Lesegeräte verwendet. Bei der Anordnung der Referenz Tags wurde festgestellt, dass bestimmte Raumstrukturen eine Positionierung

verfälschen können. Dies kann jedoch durch zusätzliche Referenz Tags (mehr tags auf gleichem Raum) verhindert werden. Mit LANDMARC konnte eine mittlere Genauigkeit von 1 Meter erzielt werden.

VIRE

Virtual Reference Elimination (VIRE), vorgestellt von Zhao et al. in [ZLN07], verwendet wie LANDMARC Referenz tags zur Bestimmung einer Position. Die Tracking tags werden an dem Objekt angebracht, das lokalisiert werden soll. Durch Eliminierung unwahrscheinlicher Positionen wird versucht, eine höhere Genauigkeit zu erreichen als beim LANDMARC Verfahren. Es sollen jedoch keine zusätzlichen Referenz tags hinzugefügt werden. Denn diese würden nicht nur höhere Kosten verursachen, sondern auch die Signale zu anderen Referenz tags stören. VIRE verwendet neben einem realen 2D Raster von Referenz tags zusätzliche virtuelle tags. Aus den bekannten Positionen der realen tags, können die Positionen der virtuellen tags berechnet werden. Mehrere Lesegeräte bestimmen die Signalstärken der Referenz und Tracking tags. Anhand einer Proximity Map wird angezeigt, ob sich ein Tracking tag im Bereich eines Referenz tags aufhält. Dies lässt sich über den RSSI-Wert bestimmen. Nachdem für alle Lesegeräte eine Proximity Map erstellt wurde, wird aus mehreren Proximity Maps der Durchschnitt berechnet. Für die Bestimmung, wann ein Tracking tag in der Nähe zu einem Referenz tag ist, wird ein geeigneter Schwellenwert benötigt. Im schlechtesten Fall wurde eine Genauigkeit von 0,47m ermittelt.

RADAR

RADAR, entwickelt von Microsoft Research, ist ein System zur Lokalisierung von Benutzern innerhalb von Gebäuden [BP00]. Zur Lokalisierung werden die Signalstärken von mehreren Wireless Local Area Network (WLAN) Basisstationen ausgewertet. Die Entwickler haben dazu drei Basisstationen auf einer Büro Etage installiert. In der Offline Phase werden einmalig die benötigten Daten für die Anwendung gesammelt. Die Signalstärken zu den Basisstationen werden als Referenzwerte aufgenommen und zusammen mit einer Position gespeichert. Die Position (x, y) wird durch einen Click auf einer Karte, in der Offline Phase, angegeben. Die Kapselung von Sensorwerten an einer bestimmten Position wird auch als Fingerprint bezeichnet. Ein Fingerprint (dt. Fingerabdruck) ist somit ein eindeutiges Merkmal von Eigenschaften an einer konkreten Position. Die mobilen Clients senden Daten mit den Signalstärken jeder Basisstation an eine zentrale Instanz. Da es schon durch den Benutzer zu einer Signaldämpfung (sie-

he Abschnitt 2.2.1) kommt, werden für jede Position Messungen aus vier verschiedenen Richtungen vorgenommen. Die Online Phase ist dann die aktive Phase, in der aus einem aktuellen Messwert eine Position bestimmt wird. Dabei wird die aktuelle Messung mit den gespeicherten Fingerprints verglichen. Es wurde ein einfacher linearer Suchalgorithmus implementiert, welcher für die relativ geringe Menge von Datensätzen ausreichend war. Ein Nachteil dieses Verfahrens ist der relativ hohe Aufwand in der Offline Phase. Verändern sich Eigenschaften in der Umgebung müssen die Fingerprints neu aufgenommen werden.

2.3 Schätzung und Vorhersage von Zuständen

In diesem Abschnitt geht es um die Grundlagen einer Zustandsschätzung mit Hilfe eines Bayes-Filters. Der Bayes-Filter schätzt die Wahrscheinlichkeit über den zukünftigen Zustand eines Systems, basierend auf dem aktuellen Zustand. Für ein besseres Verständnis sollen zunächst grundlegende Begriffe wie Wahrscheinlichkeit, Zufallsvariable, Wahrscheinlichkeitsdichte und Normalverteilung beschrieben werden.

2.3.1 Wahrscheinlichkeit

Die Wahrscheinlichkeit P (engl. probability) ist definiert durch die Anzahl der günstigen Fälle g im Verhältnis zur Anzahl der möglichen Fälle m :

$$P = \frac{g}{m} \quad (2.10)$$

Die Wahrscheinlichkeit ist somit eine Zahl zwischen Null und Eins:

$$0 \leq P \leq 1 \quad (2.11)$$

„Ein unmögliches Ereignis hat die Wahrscheinlichkeit Null[...], ein sicheres Ereignis die Wahrscheinlichkeit Eins[...].“ [HS11, S. 138]. „Die Menge der möglichen Ergebnisse eines Zufallsexperiments bilden den so genannten Ereignisraum Ω .“ [HS11, S. 135].

Zufallsvariable

Eine Zufallsvariable X kann verschiedene Werte aus einem Zufallsexperiment annehmen [Bos11]. Sie kann als Funktion $X(\omega)$ betrachtet werden, welche den Zufallsergebnissen $\omega \in \Omega$ verschiedene Werte aus \mathbb{R} zuordnet [19]. $P(X = x)$ ist die Wahrscheinlichkeit, dass eine Zufallsvariable X den Wert x annimmt.

Wahrscheinlichkeitsdichte

Mit Hilfe der Wahrscheinlichkeitsverteilung lässt sich herausfinden, mit welcher Wahrscheinlichkeit eine Zufallsvariable einen Wert kleiner oder gleich x annimmt. Die Wahrscheinlichkeitsverteilung der Zufallsvariable X ist durch die folgende Verteilungsfunktion beschrieben [HS11, S. 178]:

$$F(x) = P(X \leq x) \quad (2.12)$$

Aus der ersten Ableitung der Verteilungsfunktion erhält man die Wahrscheinlichkeitsdichtefunktion (WDF oder kurz Dichtefunktion) $f(x)$ [Obs09, S. 8]. Die Integrale einer Dichtefunktion geben die Wahrscheinlichkeit an, dass die Zufallsvariable in einem Intervall $[a, b]$ liegt:

$$P(a \leq X \leq b) = \int_a^b f(x)dx \quad (2.13)$$

Die gesamte Wahrscheinlichkeitsdichte beträgt 1:

$$\int_{-\infty}^{\infty} f(x)dx = 1 \quad (2.14)$$

Normalverteilung

In einem System, das einer Gleichverteilung unterliegt, kann jeder Zustand x aus dem System X mit der selben Wahrscheinlichkeit eintreten:

$$k = \text{Anzahl der Zustände in } X, \quad k \in N \quad (2.15)$$

$$\forall x : P = \frac{1}{k} \quad (2.16)$$

Dies widerspricht jedoch häufig Beobachtungen in der Natur, wo die Verteilung der Wahrscheinlichkeiten eben nicht gleich sind, sondern normalverteilt. Bei der Normalverteilung liegen die meisten Beobachtungen um einen bestimmten Mittelwert μ , auch genannt Erwartungswert oder Median. Schon der Mathematiker Carl Friedrich Gauß⁸ erkannte, dass sich mit der Normalverteilung Messfehler darstellen lassen und so die Zuverlässigkeit von Daten eingeschätzt werden kann. Je näher ein Wert auf der Glockenkurve am Mittelwert liegt, desto höher ist die Wahrscheinlichkeit, dass dieser Wert richtig ist [Dev12]. Die Normalverteilung ist auch als Glockenkurve oder Gaußverteilung

⁸Carl Friedrich Gauß (1777 - 1855) war ein deutscher Mathematiker, Astronom und Physiker.

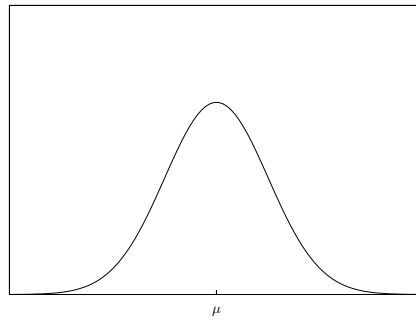


Abbildung 15: Normalverteilung: $\mu = 0$, $\sigma^2 = 2$

bekannt. Die Wahrscheinlichkeitsdichte der Normalverteilung ist gegeben durch [HS11, S. 236ff]:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (2.17)$$

Der Erwartungswert bestimmt die Lage der Verteilung auf der x-Achse. Die Standardabweichung σ ist ein Maß für die Streuung von Daten um den Mittelwert und beschreibt die Form der Kurve. Je größer σ ist, desto flacher und breiter ist die Kurve. Die Normalverteilung kann auch als $N(\mu, \sigma^2)$ geschrieben werden. Andere Verteilungen sind z.B. Lognormalverteilung oder Exponentialverteilung:

Methode von Bayes

Mit der Methode von Thomas Bayes⁹ ist es möglich eine vorhandene Wahrscheinlichkeit, beeinflusst durch ein neues Ereignis, neu zu bewerten. Angenommen es existiert eine Wahrscheinlichkeit für eine Hypothese H . Diese Wahrscheinlichkeit, wird A-priori-Wahrscheinlichkeit für die Hypothese H genannt. Tauchen neue Informationen auf, kann mit Hilfe der Bayes'schen Formel eine neue Wahrscheinlichkeit für die Hypothese H berechnet werden basierend auf der alten Wahrscheinlichkeit und dem Wissen über die neuen Informationen. Diese neue Wahrscheinlichkeit wird A-posteriori-Wahrscheinlichkeit genannt. Diese Methode erlaubt es eine A-priori-Wahrscheinlichkeit zu raten oder zu schätzen und dennoch zu einer guten A-posteriori-Wahrscheinlichkeit zu gelangen, wenn neue Informationen vorliegen¹⁰ [Dev12]. Die gesuchte A-posteriori-Wahrscheinlichkeit

⁹Thomas Bayes (1701 - 1761) war ein englischer Mathematiker und Pfarrer.

¹⁰Jedoch kann mit dieser Methode ein garbage in, garbage out nicht vermieden werden. Ungültige oder fehlerhafte Eingaben führen wahrscheinlich auch zu ungültigen oder falschen Ausgaben.

$P(H|E)$ kann nun aus der A-priori-Wahrscheinlichkeit $P(H)$ und der „inversen“ Wahrscheinlichkeit $P(E|H)$ berechnet werden:

$$P(H|E) = \frac{P(E|H) * P(H)}{P(E)} \quad (2.18)$$

Die A-posteriori-Wahrscheinlichkeit wird auch bedingte Wahrscheinlichkeit genannt.

2.3.2 Markov-Kette

Eine Markov-Kette ist „ein zeit-diskreter stochastischer Prozess (X_n) $n \in \mathbb{N}_0$ mit endlichem Zustandsraum Z “ [MGNR12, S. 188] [MRS08, S. 465ff]. Mittels Markov-Ketten lassen sich zukünftige Ereignisse aus der Kenntnis vergangener Ereignisse bzw. dem gegenwärtigen Zustand vorhersagen [Obs09, S. 12].

$$P(X_{t+1}|X_t, X_{t-1}, \dots, X_0) \quad (2.19)$$

Wenn für die Vorhersage des nächsten Zustandes ausschließlich der gegenwärtige Zustand benötigt wird, handelt es sich um eine Markov-Kette erster Ordnung:

$$P(X_{t+1}|X_t) \quad (2.20)$$

Im aktuellen Zustand ist die gesamte Historie des Prozesses gespeichert. Der Übergang von einem Zustand X_t nach X_{t+1} wird durch eine Übergangswahrscheinlichkeit beschrieben. In einem Hidden Markov Modell (siehe Abbildung 16) sind die Zustände verborgen (engl. hidden). Erst durch eine Messung bzw. Beobachtung können Informationen über den Zustand gewonnen werden. Der verborgene Zustand x_t ist nur von seinem Vorgänger x_{t-1} abhängig.

2.3.3 Bayes-Filter

Der Bayes-Filter ist ein statistischer Filter zur dynamischen Schätzung der A-posteriori-Wahrscheinlichkeit eines Zustands durch Beobachtungen [FHL⁺03] [Obs09, S. 13f]. Der Zustand kann z.B. der Aufenthaltsort einer Person oder eines Objektes sein. Mit Sensoren kann eine Beobachtung über diesen Zustand gemacht werden. In einer Zufallsvariable x_t ist der Zustand, zum Zeitpunkt t , definiert. Dabei repräsentiert $Bel(x_t)$ (engl. belief) die Schätzung der A-posteriori-Wahrscheinlichkeitsdichte über dem Zustandsraum x_t . Der Bayes-Filter versucht aus den vorliegenden Sensordaten sequentiell die A-posteriori-Wahrscheinlichkeitsdichte des Systems über den Zustandsraum zu schätzen. Fließt eine

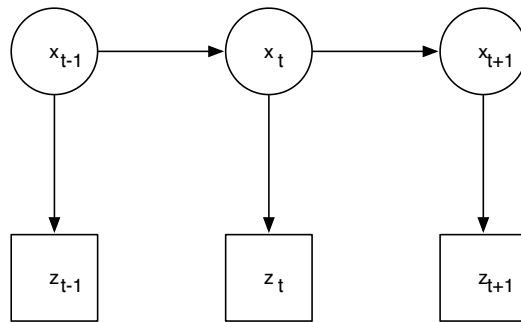


Abbildung 16: Hidden Markov Modell.

neue Beobachtung sequentiell in eine neue Schätzung ein, spricht man auch von einem rekursiven Bayes-Filter. Die Sensordaten bestehen aus einer Sequenz von Beobachtungen z_1, z_2, \dots, z_t . Die Unsicherheit ist dann durch die A-posteriori Verteilung über der Zufallsvariablen x_t , abhängig von allen Sensordaten, definiert als:

$$Bel(x_t) = p(x_t | z_1, z_2, \dots, z_t) \quad (2.21)$$

Eigentlich steigt die Komplexität zur Berechnung der A-posteriori Verteilung exponentiell mit der Zeit an, da die Anzahl der Beobachtungen mit der Zeit zunimmt. Daher nehmen Bayes-Filter an, dass das zugrunde liegende System Eigenschaften einer Markov-Kette hat und dass der Zustand von x_t nur durch den vorherigen Zustand x_{t-1} abhängig ist. Alle anderen Zustände vor x_{t-1} enthalten keine zusätzlichen Informationen.

Der Ablauf des Bayes-Filters kann in drei Schritte eingeteilt werden: Vorhersage, Beobachtung und Aktualisierung [FHL⁺03]. In der Vorhersage wird ein neuer Zustand geschätzt $Bel^-(x_t)$.

$$Bel^-(x_t) \leftarrow \int p(x_t | x_{t-1}) Bel(x_{t-1}) dx_{t-1} \quad (2.22)$$

Dabei beschreibt die bedingte Wahrscheinlichkeit $p(x_t | x_{t-1})$ die Veränderungen des Systems mit der Zeit. Dies kann z.B. das Bewegungsmodell in einem Lokalisierungssystem sein. Ein Objekt befindet sich wahrscheinlich zum Zeitpunkt t an Position x , wenn es sich vorher an x_{t-1} befunden hat. Nach der Vorhersage findet eine Beobachtung statt.

Der Filter berechnet aus der Vorhersage und der Beobachtung eine neue Schätzung:

$$Bel(x_t) \leftarrow \alpha_t p(z_t|x_t) Bel^-(x_t) \quad (2.23)$$

$p(z_t|x_t)$ ist die Wahrscheinlichkeit, dass die Beobachtung z_t an der Position x_t gemacht wurde. Die Beobachtung kann durch einen Sensor erfolgen. Der Parameter α_t ist ein Normalisierungsfaktor, der dafür sorgt, dass die Summe der A-posteriori-Wahrscheinlichkeiten über den gesamten Zustandsraum stets Eins ist. $Bel(x_0)$ wird mit dem Wissen über die Position des zu lokalisierenden Objektes initialisiert. Ist die Position jedoch unbekannt, ist die Wahrscheinlichkeit gleichmäßig über den Zustandsraum verteilt.

Bayes-Filter sind ein abstraktes Modell zur Zustandsschätzung. Eine konkrete Implementierung benötigt die Spezifizierung der Modelle zur Beobachtung $p(z_t|x_t)$ sowie zur Bewegung $p(x_t|x_{t-1})$. Eine Implementierung des Bayes-Filters ist der Partikel-Filter, welcher im nächsten Abschnitt vorgestellt wird.

2.3.4 Partikel Filter

Der Partikel Filter, entwickelt von Gordon et al. [GSS93], ist eine parameterfreie Implementierung des Bayes-Filters [Wid10, S. 28ff]. Der Partikel Filter (auch „Bootstrap Filter“ oder „survival of the fittest“ genannt) approximiert die A-posteriori-Wahrscheinlichkeitsdichte durch eine endliche Anzahl von Partikeln. Dadurch, dass das Verfahren parameterfrei ist, kann eine multimodale Verteilung repräsentiert werden.

Ein Partikel besteht aus einem aktuellen Zustand x_t^i aus dem Zustandsraum X und einem Gewicht w_t^i . Die Menge der Partikel ist:

$$S = \{(x^i, w^i) | i = 1, \dots, N\} \quad (2.24)$$

$i \in \mathbb{N}$ ist das i -te Partikel aus der Menge S . N ist die Anzahl der Partikel in der Menge S . Der Zustandsraum definiert einen Bereich, in dem eine Position festgestellt werden soll. Der Zustand ist konkrete Position aus dem Zustandsraum [Bil12, S. 19ff]. Ein Partikel repräsentiert den Zustand zum Zeitpunkt t .

Die Dichtefunktion kann durch die Summe der Partikel und ihrer Gewichte

$$p(x_t|z_t) \approx \sum_{i=1}^N w_t^i \delta(x_t - x_t^i) \quad (2.25)$$

approximiert werden. Die Summe der Gewichte w_t^i ist Eins. Je kleiner N ist, desto größer ist die Abweichung durch die Approximation. Ristic et al. [RAG04] geben einen Wert von mindestens $N = 500$ an. Je höher der Wert für N ist, desto stärker ist die Annäherung an die Dichtefunktion. Dabei wird jedoch mehr Rechenzeit benötigt. $d(x)$ ist die Kronecker-Delta Funktion, welche mit

$$\delta(x) = \begin{cases} 0, & x \neq 0 \\ 1, & x = 0 \end{cases} \quad (2.26)$$

definiert ist [Bil12, S. 21] [Obs09, S. 28f].

Algorithmus

Im folgenden ist der Algorithmus des Partikel Filters zu sehen. Als Eingabe wird dem

Algorithm 1 Partikel Filter

```

1:  $\bar{X}_t = X_t = \emptyset$ 
2: for  $i = 1$  to  $N$  do
3:    $x_t^i \sim P(x_t|x_{t-1}^i)$ 
4:    $w_t^i = P(z_t|x_t^i)$ 
5: end for
6:  $k = \sum_{i=1}^N w_t^i$ 
7: for  $i = 1$  to  $N$  do
8:    $w_t^i = k^{-1}w_t^i$ 
9:    $\bar{X}_t = \bar{X}_t + \{x_t^i, w_t^i\}$ 
10: end for
11:  $\bar{X}_t = Resample(\bar{X}_t)$ 
12: return  $X_t$ 

```

Partikel Filter eine Menge von Partikeln X_{t-1} aus der vorangegangenen Iteration übergeben. Zusammen mit einer aktuellen Messung z_t .

In Zeile 3 wird ein neues Partikel x_t^i aus der Übergangswahrscheinlichkeit $P(x_t|x_{t-1})$ erstellt. In Bezug auf eine Ortung entspricht das der Bewegung der Partikel (siehe Bewegungsmodell Abschnitt 2.3.4).

In Zeile 4 wird das Gewicht w_t^i berechnet. Das Gewicht gibt die Wahrscheinlichkeit an, mit der das Partikel durch die Messung beeinflusst wurde. Partikel bei denen die Messung wenig Einfluss hatte bekommen ein niedrigeres Gewicht als Partikel bei denen die Messung hohen Einfluss hatte.

In Zeile 7 werden schließlich noch die Gewichte der Partikel normalisiert, so dass gilt: $\sum_{i=1}^N w_t^i = 1$. Zuletzt wird das Resampling durchgeführt, welches im nächsten Abschnitt erläutert wird. Die Ausgabe ist eine Menge von Partikel X_t , welche nun nach $P(x_t|z_t)$, der A-posteriori-Wahrscheinlichkeit, verteilt sind.

Problem der Degeneration

Das Problem der Degeneration beschreibt ein Phänomen, das nach einer bestimmten Anzahl an Iterationen alle bis auf ein Partikel ein Gewicht von Null haben [DGA00, S. 199]. Die sequentielle Berechnung der Gewichte (siehe Gleichung 2.25) führt dazu, dass nur noch wenige Partikel zur Approximation der Dichtefunktion beitragen und die A-posteriori-Wahrscheinlichkeit nicht mehr richtig geschätzt werden kann [Obs09, S. 37ff]. Dieses Problem lässt sich jedoch durch Resampling umgehen [Wid10, S. 30ff] [Bil12, S. 23ff]. Partikel mit geringem Gewicht werden entfernt und Partikel mit starkem Gewicht dupliziert. Danach werden die Gewichte der übrigen Partikel wieder normalisiert, sodass für jedes Gewicht gilt: $\forall w_t^i = \frac{1}{N}$.

Algorithm 2 Resampling

```
1:  $X_t^* = \emptyset$ 
2: for  $j = 1$  to  $N$  do
3:   Wähle Zufallszahl aus  $[0,1]$ 
4:    $s = 0$ 
5:   for  $i = 1$  to  $N$  do
6:      $s = s + w_t^i$ 
7:     if  $s \geq r$  then
8:        $x_t^{*j} = x_t^i$ 
9:        $w_t^{*j} = 1/N$ 
10:       $X_t^* = X_t^* + \{x_t^{*j}, w_t^{*j}\}$ 
11:     end if
12:   end for
13: end for return  $X_t^*$ 
```

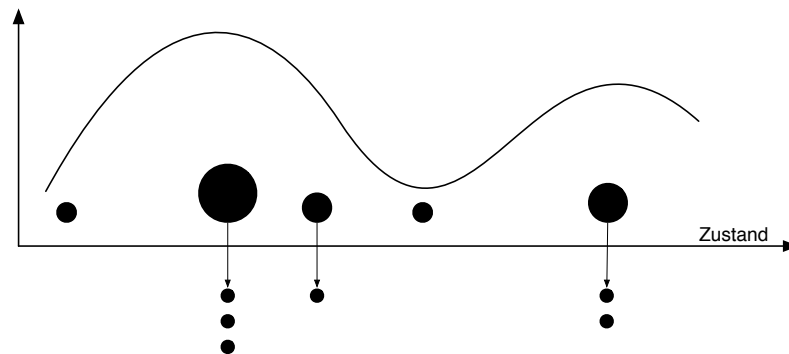


Abbildung 17: Resampling [Bil12, S. 24]

Bewegungsmodell

Mit dem Bewegungsmodell wird die Übergangswahrscheinlichkeit $P(x_t|x_{t-1})$ modelliert. Es beschreibt den Übergang von einem Zustand in einen anderen. Mit diesem Modell kann der Bewegungsablauf, genauer die zurückgelegte Entfernung und Richtung, modelliert werden. Dies erfordert entweder zusätzliche Sensoren, die z.B. die Entfernung sowie die Richtung messen oder eine zufälligen Annahme der Bewegungsparameter [Wid10]. Eine genaue Umsetzung des Bewegungsmodells wird in Abschnitt 4.2.4 auf Seite 53 erläutert.

Messmodell

Das Messmodell beschreibt die Berechnung der Wahrscheinlichkeit $P(z_t|x_t)$. Die berechnete Wahrscheinlichkeit gibt an, wie stark die Messung Einfluss auf ein Partikel hat und modifiziert entsprechend der Wahrscheinlichkeit das Gewicht des Partikels w_t^i . Im Detail wird das Messmodell in Abschnitt 4.2.4 auf Seite 55 beschrieben.

3 Analyse

In diesem Abschnitt werden die verschiedenen Methoden für eine Lokalisierung aus dem Grundlagen Kapitel zusammengefasst und gegenübergestellt. Anschließend wird untersucht, welche Sensoren auf mobilen Geräten zur Verfügung stehen und wie diese für eine Lokalisierung genutzt werden können.

3.1 Vergleich der Technologien für eine Lokalisierung

Im Folgenden sollen die Eigenschaften der verschiedenen Systeme zur Lokalisierung ausgewertet werden.

Active Badge ist auf kleine bis mittlere Raumgrößen beschränkt. Für jeden Raum wird ein weiteres Lesegerät benötigt. Tags und Lesegeräte sind relativ günstig. Active Badge wird durch Sonnenlicht und andere fluoreszierende Stoffe beeinträchtigt. Früher wurde Infrarot noch für die Übertragung von Daten zwischen mobilen Geräten verwendet, mit der Standardisierung von Bluetooth wurde Infrarot jedoch immer mehr zurückgedrängt, wodurch es kaum noch auf heutigen Mobiltelefonen zu finden ist. Daher wäre zusätzliche Hardware notwendig um eine Technologie basierend auf Infrarot mit mobilen Geräten zu verwenden.

Active Bat bietet eine sehr hohe Genauigkeit bis 10 Zentimeter . Alle $10m^2$ muss ein Lesegerät aufgebaut werden. Die Tags und die Lesegeräte sind relativ günstig. Die Auswertung muss über eine zentrale Instanz erfolgen. Die Referenz tags müssen an der Zimmerdecke in einem Raster angebracht werden.

GPS bietet eine Genauigkeit von 1 bis 5 Meter in freiem Gelände. Innerhalb von Gebäuden funktioniert die Positionierung durch Abschirmung nicht mehr. Die meisten neueren Mobiltelefone sind mit einem GPS Empfänger ausgestattet.

Mit RADAR wurde eine Genauigkeit von 3 bis 4,3 Meter erreicht. RADAR benötigt

lediglich einige WLAN Basisstationen.

Technologie	Genauigkeit (Präzision)	Skalierung	Kosten	Einschränkungen
GPS	1 - 5m (95 - 99%)	24 Satelliten, beliebig viele Empfänger	Teure Infrastruktur, Empfänger ca. 75€	nicht in Gebäuden
Active Badge	Raumgröße	eine Basisstation pro Raum	Tags und Basisstationen relativ günstig	Sonnenlicht und fluoreszierende Stoffe
Active Bat	10cm (95%)	alle 10m ² eine Basisstation	günstige Tags und Sensoren	Sensoren müssen an der Decke angebracht werden, Anti-Kollisionsverfahren
RADAR	3 - 4,3m (50%)	drei Basisstationen pro Flur	ca. 75€ pro Basisstation	
RFID	1m (50%)	beliebig viele Tags	günstige Tags, Lesegeräte relativ teuer	viele Tags müssen an der Decke angebracht werden

Tabelle 1: Übersicht der Technologien.

Die Auswertung der Technologien zeigt, dass jede Technologie ihre Vor- und Nachteile hat. Verfahren wie RFID oder Active Bat benötigen neben einer Infrastruktur auch eine zentrale Instanz, welche die Daten auswertet. Manche Verfahren verwenden Technologien, welche nur noch in wenigen oder gar keinen mobilen Gerät eingebaut sind z.B. Infrarot oder Ultraschall. Im nächsten Abschnitt soll untersucht werden, welche Sensoren auf einem mobilen Gerät zur Verfügung stehen und wie diese verwendet werden können.

3.2 Einsatzmöglichkeiten der Sensorik

Licht-Sensor

Der Licht-Sensor ermittelt die Lichtintensität der Umgebung. Die ermittelten Werte werden in Lux (lx) angegeben. Ein Lux ist definiert durch:

$$lx = \frac{lm}{m^2} \quad (3.1)$$

Lumen lm (lat. Licht) ist die Einheit des Lichtstroms. Lumen berücksichtigt die Lichtempfindlichkeit des menschlichen Auges. Senden zwei Lichtquellen den selben Lichtstrom aus, werden diese als gleich hell wahrgenommen [20]. Ein Lux ist die Menge an Lichtstrom, die auf einen Quadratmeter fällt. Es kann davon ausgegangen werden, dass die Lichtintensität in der Nähe eines Fenster höher ist als auf der Fenster abgewandten Seite eines Raumes. Die natürliche Veränderung der Lichtintensität innerhalb eines Tages führt jedoch dazu, dass der Lichtsensor nur in Räumen mit gleichmäßiger Beleuchtung eingesetzt werden kann [Bun12].

Magnetfeld-Sensor

Der Magnetfeld-Sensor kann das ihn umgebene magnetische Feld messen. Dabei wird die Magnetfeldstärke oder der magnetische Fluss in eine elektrische Größe umgewandelt [21]. Die berechneten Werte werden in mikro Tesla (μT) angegeben. Bilke hat in seiner Arbeit [Bil12] zur Positionierung mit Magnet- und Licht-Sensoren eine mittlere Genauigkeit von 4 Metern erzielt.

Luftdruck-Sensor

Da GPS in Gebäuden nicht funktioniert, soll untersucht werden, ob sich der Luftdruck-Sensor zur Bestimmung einer vertikalen Positionsveränderung nutzen lässt. Mit einem Luftdruck-Sensor kann der aktuelle Luftdruck in der Umgebung gemessen werden. Nach dem pascal'schen Gesetz wird der Luftdruck aus dem hydrostatischen Druck berechnet, welcher sich für Flüssigkeiten oder Gase unter Einfluss eines Schwerfeldes ergibt. Der Druck ist die Kraft, die auf eine Fläche wirkt. Der Luftdruck ist die „Gewichtskraft der Luftsäule, die auf der Erdoberfläche oder einem auf ihr befindlichen Körper steht“ [22].

Der Luftdruck wird in Pascal (kurz Pa, nach dem franz. Mathematiker Blaise Pascal benannt) bzw. in HektoPascal (hPa) angegeben. „Ein Pascal entspricht dem Druck von einem Newton pro Quadratmeter.“ $Pa = N/m^2$ [23]. Die Einheit Hektopascal bezieht sich auf eine Fläche von $100cm^2$ bzw. $1dm^2$. Auf Höhe des Meeresspiegels beträgt der mittlere Luftdruck $1013 hPa$ bei $15\text{ }^\circ C$ [ZS10, S. 115]. Der Luftdruck sinkt etwa um $1 hPa$ je 8 Meter Höhenanstieg. Des weiteren unterliegt der Luftdruck natürlichen Schwankungen während eines Tages, wie in Abbildung 18 zu sehen ist. Stündliche Messungen innerhalb eines Tages haben gezeigt, dass sich der Luftdruck mit einer Differenz von $\Delta 38,96 hPa$ verändert hat. Zwischen 9 und 10 Uhr gab es einen relativ starken Abfall des Luftdrucks wie Abbildung 18(b) verdeutlicht. In Tabelle 2 sind die ermittelten Maxima und Minima

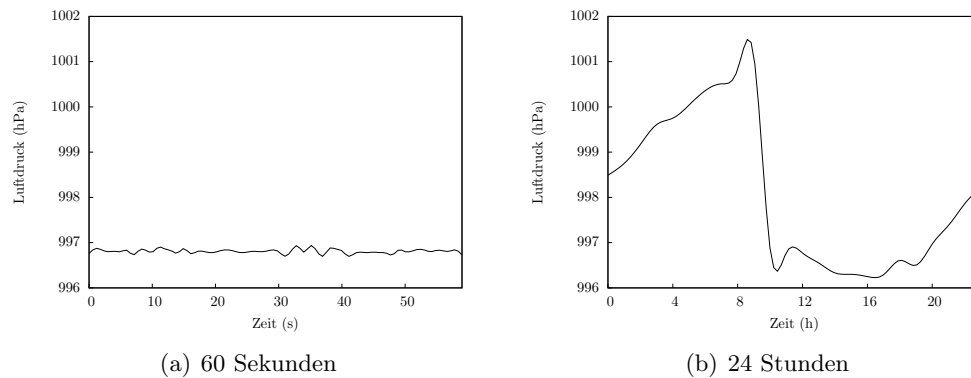


Abbildung 18: Veränderung des Luftdrucks innerhalb einer Minute und im Verlauf eines Tages.

sowie die Differenz und den sich daraus ergebenden Höhenunterschied von verschiedenen Messungen zu sehen. Durch die Schwankungen des Luftdrucks ist es nicht möglich diese

Intervall	Max (hPa)	Min (hPa)	$\Delta(\text{Min,Max})$ (hPa)	Δ Höhe (m)
24 Stunden	1001,12	996,25	4,87	38,96
60 Sekunden	996,93	996,70	0,23	1,84

Tabelle 2: Vergleich der Luftdruck Messung innerhalb 60 Sekunden und 24 Stunden.

absolut zu betrachten und z.B. durch eine Szenenanalyse auszuwerten. Vielmehr können jedoch die relativen Unterschiede während eines kurzen Zeitraums betrachtet werden. Abbildung 18(a) zeigt den Verlauf des Luftdrucks innerhalb einer Minute. Der Verlauf des Luftdrucks innerhalb einer Minute hat ein Minima bei $996,7 \text{ hPa}$ und ein Maxima bei $996,93 \text{ hPa}$. Die Differenz von Minima und Maxima liegt bei $\Delta 0,23 \text{ hPa}$. Um eine vertikale Positionsveränderung festzustellen, sollte die Luftdruckdifferenz zwischen zwei Etagen deutlich größer sein als $\Delta 0,23 \text{ hPa}$. Zwischen der dritten und der zweiten Etage des TGS, wurde eine Luftdruckdifferenz von $\Delta 0,5 \text{ hPa}$ gemessen. Dieser Wert unterscheidet sich deutlich von den natürlichen Schwankungen. Tritt also eine Veränderung von mehr als $\Delta 0,5 \text{ hPa}$ innerhalb einer Minute oder weniger auf, so kann davon ausgegangen werden, dass sich eine vertikale Positionsveränderung ereignet hat¹¹. Mit Hilfe der Barometrischen Höhenformel kann der (Luft)Druck p_h als Funktion der Höhe h , bei konstanter Temperatur und einer von der Höhe unabhängigen Fallbeschleunigung

¹¹Auswirkungen durch plötzliche Wetteränderungen, welche Einfluss auf den Sensor haben, sind zu erwarten, sollen in dieser Arbeit aber nicht weiter berücksichtigt werden.

g bestimmt werden [DKV09, S. 136f]:

$$p(h) = p_0 e^{\frac{-\rho_0 g h}{p_0}} \quad (3.2)$$

„Für Luft von $0\text{ }^\circ\text{C}$ gilt mit $\rho_0 = 1,293 \frac{\text{kg}}{\text{m}^3}$ und $p_0 = 1013 \text{ hPa}$ “:

$$p(h) = p_0 e^{\frac{-h}{7,96 \text{ km}}} \quad (3.3)$$

In Abbildung 19 ist die Funktion abgebildet. Es ist zu erkennen, dass der Luftdruck nicht linear mit der Höhe abnimmt. Die Formel ist nur eine Annäherung an die Wirklichkeit. Für kleine Intervalle kann aufgrund der natürlichen Schwankungen keine eindeutige Aussage über die Höhe gemacht werden.

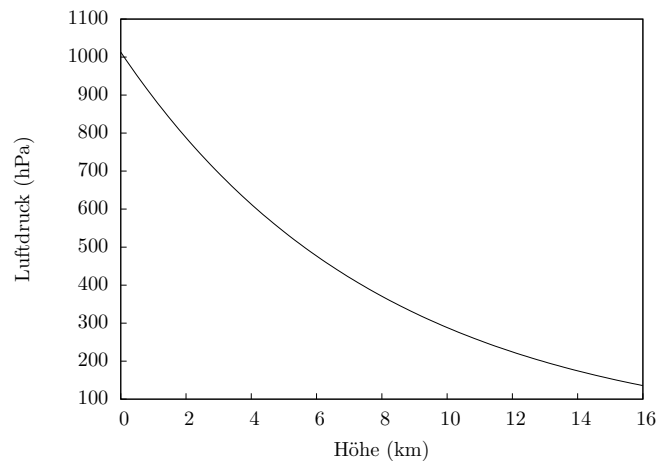


Abbildung 19: Verhältnis des Luftdrucks und der Höhe mit der Barometrischen Höhenformel.

Beschleunigungs-Sensor

Der Beschleunigungs-Sensor ermittelt die Beschleunigung mittels „[...] einer oder mehrerer Probmassen, die unter dem Einfluss der zu messenden Beschleunigung ausgelenkt werden.“ [Wen07, S. 65f]. „Die Beschleunigung (\vec{a}) gibt an, wie schnell (t) sich die Ge-

schwindigkeit (v) eines Körpers ändert.“[ZS10, S. 23f]:

$$\vec{a} = \frac{\Delta v}{\Delta t} \quad (3.4)$$

$$\vec{a} = \frac{m * s^{-1}}{s} = \frac{m}{s^2} \quad (3.5)$$

Die Beschleunigung kann auch über das Kraft-Masse-Verhältnis betrachtet werden. Ein Körper wird beschleunigt, wenn eine Kraft F^{12} auf einen Körper mit einer Masse m wirkt [Ger06].

$$\vec{a} = \frac{F}{m} \quad (3.6)$$

$$\vec{a} = \frac{kg * m * s^{-2}}{kg} = \frac{m}{s^2} \quad (3.7)$$

Die Beschleunigung wird in $\frac{m}{s^2}$ angegeben. Zu beachten ist, dass die Gravitation mit $g = 9.81 \frac{m}{s^2}$ auf den Sensor einwirkt. Befindet sich das mobile Gerät in Ruhelage, dann gibt der Sensor $9.81 m/s^2$, als Betrag der Beschleunigung, aus. Befindet sich das Gerät im freien Fall gibt der Sensor als Betrag $0 m/s^2$ aus [24]. Die Beschleunigung kann dazu genutzt werden, um zu bestimmen, ob sich ein Benutzer bewegt oder nicht. Die Beschleunigung wird in einem drei Achsen Koordinatensystem angegeben. Auf der z-Achse werden Beschleunigungen nach oben oder unten registriert. Abbildung 20 zeigt die gemessene Beschleunigung auf der z Achse. Dabei ist zu erkennen, dass bei einer

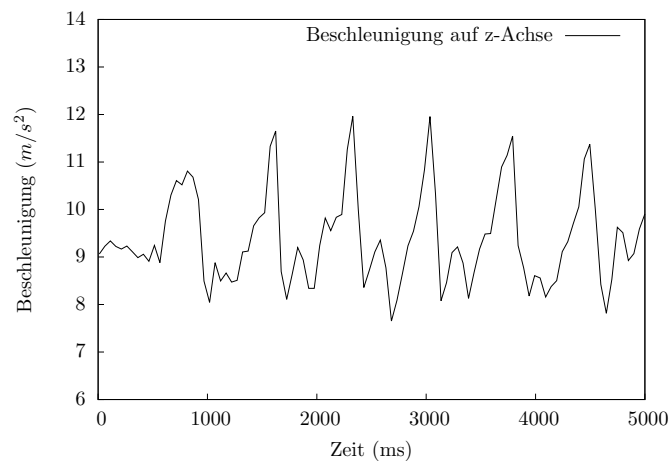


Abbildung 20: Zeigt die Auf- und Abwärtsbewegung beim langsamen gehen.

¹²Krafteinheit ist Newton $1N = 1kg * m * s^{-2}$ [Ger06].

langsamen Laufgeschwindigkeit die Werte auf der z -Achse auf- und absteigen. Dies ist durch die Laufbewegung zu erklären bei der sich der Körper, wenn die Person ein Bein nach vorn verlegt, leicht absenkt und wieder aufwärts bewegt, sobald beide Beine parallel zu einander stehen.

Aus diesen Messwerten, kann abgelesen werden, dass eine Bewegung beginnt, wenn auf der z -Achse entweder $11 \frac{m}{s^2}$ überschritten oder $8,5 \frac{m}{s^2}$ unterschritten werden.

$$Bewegung = \begin{cases} \top, & z > 11 \frac{m}{s^2} \quad \vee \quad z < 8,5 \frac{m}{s^2} \\ \perp & \end{cases} \quad (3.8)$$

Das Ergebnis einer Bewegungsmessung soll zwei Zustände haben: Stehen und Bewegen. Wie schnell sich eine Person bewegt soll nicht weiter untersucht werden. Es reicht zu wissen ob eine Person steht oder sich gerade bewegt.

Kompass

Mit einem Kompass kann eine Richtung angegeben werden, indem z.B. die Himmelsrichtung bestimmt wird, wobei zwischen Magnet- und Kreisel-Kompass zu unterschieden ist [Band 12, S. 229f][Bro86]. Der Magnet-Kompass nutzt das Magnetfeld der Erde, welches auf die Magnetnadel eine Richtkraft ausübt. Die Magnetnadel wird dadurch auf die magnetische Nord-Süd-Richtung eingestellt¹³. Ein Kreisel-Kompass verwendet die Rotation der Erde zur Ausrichtung, indem er sich parallel zur Erdrotationsachse ausrichtet. Da an den geographischen Polen die Drehachse der Erde nahezu senkrecht aus der Erdoberfläche austritt, funktionieren Kreisel-Kompass dort nicht. Ein Kreisel-Kompass ist empfindlich gegenüber Beschleunigungen.

Ein Kompass gibt die Richtung in Grad an. Soll daraus eine der vier Himmelsrichtungen Nord, Süd, Ost, West bestimmt werden, müssen den Richtungen feste Bereiche in Grad vorgegeben werden. Davon ausgehend das 0° genau zum magnetisch Nordpool zeigt, befindet sich die Himmelsrichtung jeweils im Abstand von 90° zueinander. Der

¹³Ausgenommen sind die magnetischen Pole. Dort kann die Magnetnadel sich nicht auf einen Punkt ausrichten, da das Magnetfeld in alle Richtungen gleichmäßig wirkt.

genaue Bereich jeder Himmelsrichtung ist wie folgt definiert:

$$Himmelsrichtung = \begin{cases} Nord, & 315^\circ \leq \alpha \vee \alpha < 45^\circ \\ Ost, & 45^\circ \leq \alpha < 135^\circ \\ Sued, & 135^\circ \leq \alpha < 225^\circ \\ West, & 225^\circ \leq \alpha < 315^\circ \end{cases} \quad (3.9)$$

Eine Richtung von 40° entspricht somit der Richtung Nord, wie in Abbildung 21 zu sehen ist.

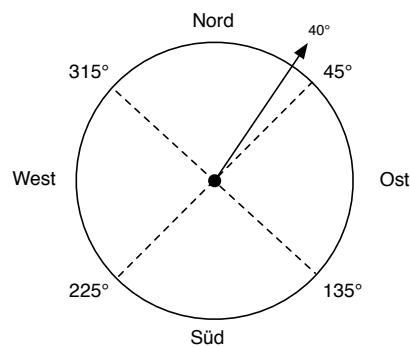


Abbildung 21: Ein Kompass gibt die aktuelle Richtung mit $\alpha = 40^\circ$ an. Dies entspricht der Richtung Nord.

Gyroskop

Allgemein betrachtet ist jeder Körper, der eine Drehbewegung ausführt ein Kreisel wie z.B. rotierende Flüssigkeiten in einem Behälter [Fab80, S. 17]. In diesem Fall geht es aber um einen besonderen Kreisel, dem Gyroskop. Ein Gyroskop (siehe Abbildung 50(a) auf Seite 92), auch Kreiselinstrument genannt, ist ein sich schnell rotierender, symmetrischer Kreisel, der sich in einem beweglichen Lager dreht [25]. Gyroskope werden als Navigationsinstrumente in der Luft- und Raumfahrt eingesetzt.

Eine Form der Navigation ist die Trägheitsnavigation, bei der die Massenträgheit gegenüber Beschleunigungen und Drehbeschleunigungen genutzt wird, um damit die Beschleunigung und die Drehgeschwindigkeit zu messen. Ist die Beschleunigung und die Richtung bekannt, kann durch Integration über die Zeit, die Geschwindigkeit sowie die zurückgelegte Distanz ermittelt werden [Wen07, S. 28f]. Diese Art der Navigation wird

auch als Inertialnavigation bezeichnet. Ein Trägheitsnavigationssystem (engl. Inertial Navigation System (INS)) hat den Vorteil, dass es sich um ein autonomes Verfahren handelt, welches nicht auf zusätzliche Hilfe von außen angewiesen ist und sich nicht durch ausgesendete Signale verrät [Fab80, S. 319f]. Für die Trägheitsnavigation werden an Kreiselinstrumente sehr hohe Anforderungen bezüglich der Genauigkeit gestellt.

3.3 Anforderungen

Funktionale Anforderungen

Im Folgenden sind die wesentlichen Funktionen beschrieben, welche im Prototypen umgesetzt werden sollen.

F01 Check-In im Museum

Der Benutzer soll in einem bestimmten Gebäude einchecken können, damit die entsprechenden Informationen, wie eine Karte sowie Lokalisierungs- und Objektdaten geladen werden können.

F02 Lokalisierung des Benutzers

Der Benutzer soll in einem Museum lokalisiert werden. Die Position kann eine symbolische Position sein, die aussagt, in welchem Raum sich der Benutzer gerade befindet.

F03 Energiesparen

Da die Nutzung der Sensoren Energie benötigt, besonders ein Bluetooth oder WLAN Scann, sollen die Sensoren nur zum Einsatz kommen, wenn sie auch benötigt werden. Die Sensoren sollen seltener aktiviert werden, wenn sich der Benutzer nur langsam oder gar nicht bewegt. Bewegt er sich dagegen schnell oder über einen längeren Zeitraum, soll davon ausgegangen werden, dass er seine Position verändert hat.

F04 Objekterkennung

Die Objekte in einem Raum sollen über visuelle Verfahren erkannt werden. Der Benutzer richtet dazu die Kamera seines mobilen Gerätes auf ein bestimmtes Objekt. Sind Daten zu dem Objekt vorhanden sollen diese dem Benutzer präsentiert werden.

Nichtfunktionale Anforderungen

NF01 Generalisierbarkeit

Die Anwendung soll zu beliebig vielen Objekten Informationen in angemessener Zeit bereitstellen können.

NF02 Leichtgewichtige Infrastruktur

Die Anwendung soll nur eine leichtgewichtige Infrastruktur voraussetzen, welche wenig Zeit und Geld für Installation und Wartung benötigt.

4 Konzept für die Lokalisierung

In diesem Kapitel wird erläutert, wie eine Lokalisierung mit Hilfe eines mobilen Gerätes umgesetzt werden kann. Wie aus der Analyse hervorgeht sind bestimmte Technologien, wie z.B. Infrarot oder Ultraschall auf den meisten mobilen Geräten nicht verfügbar. Andere Technologien wie RFID benötigen eine komplexere Infrastruktur und zusätzliche Hardware. Darum soll eine Lokalisierung nur mit den Technologien umgesetzt werden, die auf den meisten mobilen Geräten zur Verfügung stehen. Dies sind z.B. WLAN und Bluetooth. Mit Hilfe dieser beiden Funk-Technologien soll eine Lokalisierung in einem Gebäude realisiert werden. Weitere Sensoren, wie Kompass, Beschleunigungs-Sensor oder Luftdruckmesser sollen eine Lokalisierung unterstützen.

Es wird beschrieben, welche Daten gesammelt werden müssen und welche Algorithmen angewendet werden. Die Lokalisierung erfolgt in zwei Phasen. Zuerst werden in der Offline Phase die Signalstärken von WLAN und Bluetooth Basisstationen aufgezeichnet. In der zweiten Phase, der Online Phase, sollen die aktuellen Messwerte mit den aufgezeichneten Werten aus der Offline Phase verglichen werden. Die Annahme ist, dass die Signalstärken der verschiedenen Basisstationen an verschiedenen Orten unterschiedlich sind und somit einen Punkt im Raum eindeutig beschreiben.

4.1 Bestimmung des aktuellen Gebäudes

Die Bestimmung des Gebäudes ist wichtig, um die entsprechenden Daten für die Lokalisierung und der Objekte zu laden. Dies kann durch verschiedene Technologien realisiert werden.

- Zur Bestimmung des Gebäudes können die umliegenden Base Transceiver Station (BTS) (dt. Funkzelle) ausgewertet werden. Eine BTS stellt die Funkverbindung mit den mobilen Geräten her. Eine BTS kann eine Fläche (Zelle) mit einem Radius von bis zu 15 km abdecken¹⁴. Da jedoch die Anzahl der Nutzer beschränkt

¹⁴Theoretisch sind zwar bis zu 35 km möglich, diese werden aber selbst auf dem Land nicht erreicht.

ist, sind die Zellen in dicht besiedelten Gebieten deutlich kleiner (Reichweite 100 bis 4000 Meter) [Sau10, S. 35f]. Durch Auswertung der umliegenden Funkzellen kann mittels einer zellbasierten Ortung (siehe Abbildung 13 auf Seite 20) eine Zuordnung zum aktuellen Standort gemacht werden. Problematisch ist hier jedoch, wenn zwei Gebäude sich direkt nebeneinander befinden, dann könnte es sein, dass beide Gebäude in der selben Funkzelle liegen. Ein anderes Problem taucht auf, wenn Funkzellen ausgetauscht werden und möglicherweise eine andere Identifikation aussenden.

- Mit Near Field Communication (NFC) können Daten über kurze Reichweiten übertragen werden. Ein NFC-Tag könnte wie ein QR-Tag am Eingang des Gebäudes angebracht werden. Der NFC-Tag kann dann vom Benutzer eingelesen werden um so das aktuelle Gebäude zu identifizieren. Das Problem ist, dass NFC noch nicht in allen mobilen Geräten integriert ist.
- GPS kann nicht innerhalb von Gebäuden genutzt werden. Jedoch könnte es zur Bestimmung des aktuellen Gebäudes verwendet werden. Solange der Benutzer sich vor dem Gebäude befindet, bekommt er GPS Daten. Sobald er das Gebäude betritt, bricht der Sichtkontakt zu den Satelliten ab. Eine genaue Positionierung ist dann nicht mehr möglich. Jedoch kann mit den letzten gültigen GPS Daten eine Position bestimmt werden, dass der Benutzer sich vor diesem Gebäude befunden hat. Die Auswertung der GPS Daten ist sehr viel genauer als die Funkzellen-Ortung und einfacher als eine Lösung mit QR oder NFC tags. Daher soll die Bestimmung des aktuellen Gebäudes mittels GPS umgesetzt werden. Dazu wird die aktuelle Position des Benutzers mit den gespeicherten Positionen von Gebäuden verglichen. Die Berechnung der Entfernung zweier GPS Positionen ist in den Grundlagen 2.2.3 ab Seite 23 beschrieben.

4.2 Lokalisierung innerhalb eines Gebäude

4.2.1 Aufnahme der Basisdaten in der Offline Phase

In der Offline Phase werden die Daten der WLAN und Bluetooth Basisstationen an bestimmten Punkten aufgezeichnet. Für jede Basisstation (B) wird die Signalstärke (S) ermittelt und gespeichert. Die Idee ist, dass an verschiedenen Punkten im Raum unterschiedliche Signalstärken von verschiedenen Basisstationen gemessen werden. Im besten Fall steht damit für jeden Punkt im Raum eine eindeutige Signatur, ähnlich dem eines

Fingerabdrucks zur Verfügung. Statt Fingerabdruck wird im Folgenden von einem Fingerprint gesprochen. Ein Fingerprint (FP) speichert einen Vektor mit Signalstärken zu n Basisstationen, mit $n \in \mathbb{N}$:

$$\vec{S} = (S_1, S_2, \dots, S_n) \quad (4.1)$$

Da es durch die Person, welche die Aufzeichnungen durchführt, zu Dämpfungs-Effekten der Signale kommt, werden für jede Position die Messungen aus genau einer von vier Richtungen aufgezeichnet:

$$D = \{\text{Nord, Süd, Ost, West}\} \quad (4.2)$$

Jedem Fingerprint ist eine Richtung ($d \in D$) zugeordnet. Zusätzlich wird für jeden Fingerprint eine eindeutige Position $\vec{P} = (x, y)$ gespeichert. Die Position kann später auf einer digitalen Karte angegeben werden¹⁵. Ein Fingerprint besteht somit aus dem Tupel:

$$FP = (\vec{P}, d, \vec{S}), \quad d \in D \quad (4.3)$$

Da es natürliche Schwankungen geben kann, welche die Signalstärken beeinflussen, soll die Aufzeichnung über einen bestimmten Zeitraum (Δt) gemacht werden und anschließend das arithmetische Mittel aller aufgezeichneter Werte für jede Basisstation berechnet werden. Die berechneten Durchschnittswerte werden für jede Basisstation und zu jeder Richtung gespeichert.

Anordnung der Fingerprints

Prinzipiell gibt es keine Anforderungen an die Anordnung der Fingerprints. Je nach Raumgröße sollten im Abstand von 1 bis 2 Metern Fingerprints erstellt werden. Im Folgenden wird eine Gitterstruktur bevorzugt, deren daraus resultierenden Vor- oder Nachteile sollen nicht weiter untersucht werden. Die Anzahl und Anordnung der Fingerprints ist natürlich von der Aufteilung und Größe des Raums abhängig. Abbildung 22 zeigt eine mögliche Anordnung der Fingerprints (rote Punkte).

¹⁵Die Position wird als Pixelwert (x, y) angegeben. Die Position bezieht sich somit nur auf das für die Entwicklung verwendete Gerät und dessen Bildschirmgröße. Andere Geräte, mit anderen Bildschirmgrößen, werden vorerst nicht unterstützt.

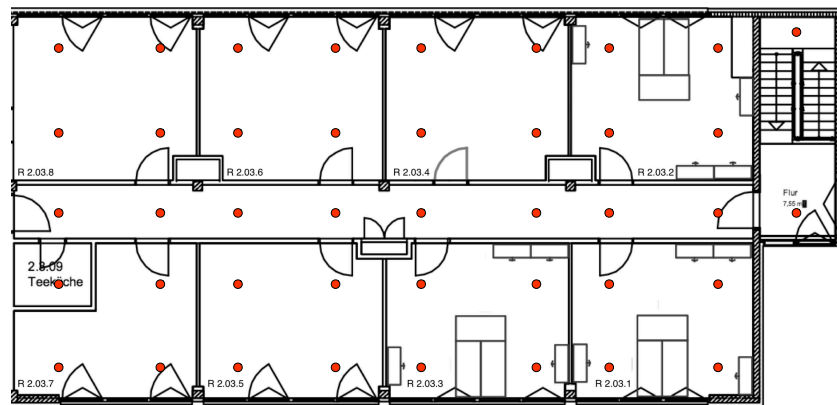


Abbildung 22: Anordnung der Fingerprints (rote Punkte) auf einer Karte des TGS, Gebäude 2 in der dritten Etage.

4.2.2 Bestimmung einer Position in der Online Phase

In der Online Phase werden die Werte der Sensoren ausgewertet, um daraus eine Position abzuleiten. Zunächst wird untersucht, wie auf Basis der Signalstärke eine Position ermittelt werden kann. Dieser Fall berücksichtigt jedoch nur den aktuellen Zustand. Der letzte und die vorangegangenen Zustände werden bei dieser Art der Positionsbestimmung nicht berücksichtigt. Darum soll ein Konzept zur dynamischen Zustandsschätzung beschrieben werden, bei der der vorangegangene Zustand mit in die Berechnung der neuen Position einbezogen wird.

Die aufgezeichneten Fingerprints aus der Offline Phase dienen nun als Referenzwerte zu den aktuellen Messwerten. Da jeder Fingerprint Messdaten aus vier verschiedenen Richtungen enthält sollen nur Daten aus der Richtung verwendet werden, in die der Benutzer gerade schaut. Die Bestimmung der Richtung erfolgt mit dem Kompass und ist in Abschnitt 3.2 beschrieben. Nun kann die Ähnlichkeit des Vektors aus dem Fingerprint mit dem Vektor der aktuellen Messung verglichen werden. Es stehen verschiedene Methoden zur Berechnung der Ähnlichkeit zweier Vektoren zur Verfügung z.B. der euklidische Abstand oder die Kosinus Ähnlichkeit.

Euklidischer Abstand

Mit dem euklidischen Abstand kann die Länge zwischen zwei Vektoren ermittelt werden. Die Länge d zweier n -dimensionaler Vektoren \vec{a} und \vec{b} ist wie folgt definiert:

$$d(\vec{a}, \vec{b}) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2} \quad (4.4)$$

Die Länge zwischen zwei Vektoren kann als Ähnlichkeitsmerkmal angesehen werden. Je kürzer die Länge ist, desto dichter liegen die Vektoren beieinander und desto ähnlicher sind sie. Der Abstand wird aus den Signalstärken aller Basisstation k berechnet, die sowohl im Fingerprint z^+ als auch in der aktuellen Messung z vorhanden sind: $k \in z^+ \cap z$.

$$d(k_{z^+}^{\vec{a}}, k_z^{\vec{b}}) = \sqrt{(k_{z^+}^1 - k_z^1)^2 + \dots + (k_{z^+}^n - k_z^n)^2} \quad (4.5)$$

Die Abstände werden den entsprechenden Fingerprints zugeordnet. Der Fingerprint mit dem geringsten Abstand, zur aktuellen Messung, kann folglich als der Fingerprint angesehen werden, der mit der aktuellen Messung am besten Übereinstimmt. Die ermittelten Abstände sagen jedoch nichts über die Qualität der aktuellen Messung aus. Wie verhält es sich jedoch mit Fingerprints, bei denen die ermittelten Abstände zur aktuellen Messung sehr dicht beieinander liegen? Hier kann nicht einfach der beste Fingerprint genommen werden, denn möglicherweise ist dies nicht der richtige. Es muss eine Aussage gemacht werden ob der beste Vektor wirklich gut ist. Dazu können die Vektoren des besten und des zweitbesten Fingerprints direkt miteinander verglichen werden. Nun kann das Verhältnis der zuvor berechneten Abstände der beiden Vektoren betrachtet werden. Ein Schwellwert T definiert, ob die beiden Vektoren ähnlich sind oder nicht. Ein anderes Ähnlichkeitsmaß ist die Kosinus-Ähnlichkeit.

Kosinus-Ähnlichkeit

Die Kosinus-Ähnlichkeit (*sim*) bestimmt den Winkel zwischen zwei gegebenen Vektoren \vec{a} und \vec{b} , indem der Kosinus des eingeschlossenen Winkels θ der beiden Vektoren berechnet wird [MRS08][26].

$$sim(\vec{a}, \vec{b}) = \frac{\sum_{i=1}^n a_i \times b_i}{\sqrt{\sum_{i=1}^n (a_i)^2} \times \sqrt{\sum_{i=1}^n (b_i)^2}} \quad (4.6)$$

Die Kosinus-Ähnlichkeit gibt an, ob zwei Vektoren in dieselbe Richtung zeigen. Die Werte liegen dabei zwischen -1 und 1. Wobei ein Wert von 0 eine lineare Unabhängigkeit der beiden Vektoren bedeutet und somit absolut keine Ähnlichkeit besteht. Bei 1 oder -1 stehen beide Vektoren parallel zueinander.

Bestimmung einer Position

Nachdem ein Fingerprint mit der größten Übereinstimmung zum aktuellen Messwert ermittelt wurde, wird die Position dieses Fingerprints als die wahrscheinlichste Position des Benutzers angenommen. Diese Annahme basiert im Wesentlichen auf dem CoO Verfahren (siehe Abschnitt 2.2.1).

Verbesserung der ermittelten Position

Die Positionsbestimmung kann noch verbessert werden, indem die k-besten Fingerprints mit in die Positionsfindung einbezogen werden. Die k-besten Fingerprints werden nach absteigender Reihenfolge aus dem ermittelten euklidischen Abstand bestimmt. Der beste Fingerprint ist der mit dem geringsten euklidischen Abstand zum aktuellen Messwert. Aus diesen k-besten Fingerprints werden die ermittelten euklidischen Abstände genommen, um ein Gewicht w_i zu berechnen. Das Gewicht gibt an wie stark der Einfluss der jeweiligen Fingerprints auf die tatsächliche Position ist. Die Summe der Gewichte aller Fingerprints ist Eins. Das Gewicht jedes Fingerprints wird mit seiner Position multipliziert.

$$\vec{p} = \sum_{i=1}^k \vec{p}_i * w_i \quad (4.7)$$

4.2.3 Raumbasiert Lokalisierung

Ein andere Möglichkeit der Positionierung ist, alle Messwerte eines Raumes zu betrachten. Unabhängig von der Position einer Person im Raum, ist es das Ziel eine Aussage zu machen, in welchem der Räume die Person sich befindet. Dazu werden wie zuvor die Messerwerte aller Fingerprints mit den aktuellen Messwerten verglichen und ein Ähnlichkeitswert berechnet. Anstatt den besten Fingerprint auszuwählen, soll das arithmetische Mittel aller Fingerprints eines Raums berechnet werden. Die Annahme ist, dass im Durchschnitt die Werte für enen Raum stabiler als die der einzelner Fingerprints sind. Abbildung 23 zeigt ein Beispiel für die berechneten Mittelwerte in jedem Raum. Diese zeigt, dass der geringste Mittelwert in Raum 2.3.03 vorhanden ist. Hier existiert

die größte Übereinstimmung bezogen auf eine aktuelle Messung. Ob dieser Ansatz funktioniert, soll eine Implementierung zeigen.

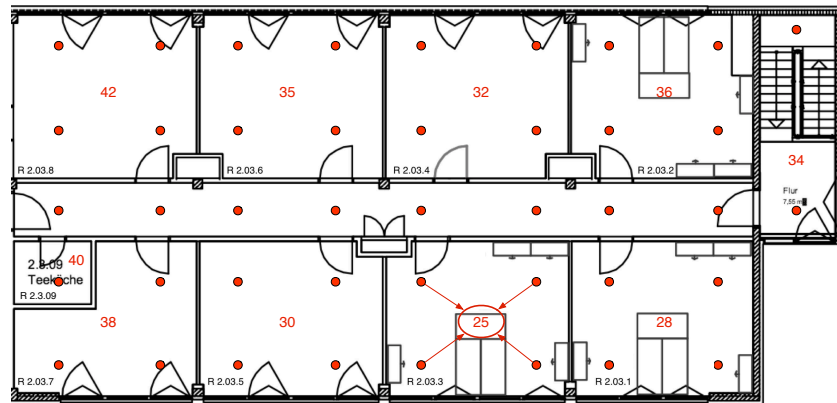


Abbildung 23: Arithmetisches Mittel aller Fingerprints in jedem Raum.

4.2.4 Dynamischer Zustandsschätzer: Partikel Filter

Die bisher vorgestellten Methoden zur Lokalisierung berücksichtigen nur den aktuellen Zustand einer Messung. Es kann zwar eine Aussage gemacht werden, welche Fingerprints am ähnlichsten sind, dennoch werden daraus keine weiteren Informationen entnommen um eine neue Messung zu bewerten. Mit Hilfe eines Bayes Filters (siehe Abschnitt 2.3) basierend auf der Methode von Thomas Bayes soll ein neuer Zustand durch eine aktuelle Messung bewertet werden. Damit soll verhindert werden, dass eine neue Position stark von der letzten Position abweicht. Dazu wird im folgenden das Bewegungs- und Messmodell beschrieben.

Bewegungsmodell

Das Bewegungsmodell beschreibt, wie sich die Partikel im Zustandsraum in Bezug auf die reale Bewegung des Benutzers mit dem mobilen Gerät ausbreiten. Das Bewegungsmodell soll die Geschwindigkeit sowie die Ausbreitungsrichtung der Partikel modellieren.

- **Geschwindigkeit**

Die Geschwindigkeit eines Partikels hängt von der geschätzten Geschwindigkeit des Benutzers ab. Das Bewegungsmodell geht davon aus, dass sich eine Personen nur mit einer maximalen Geschwindigkeit Max_{vt} fortbewegen kann und je größer die Zeitdifferenz Δt zwischen den Messungen ist desto höher ist die Wahrscheinlichkeit,

dass sich die Geschwindigkeit geändert hat [Bil12, S. 38f]. Nach [Wid10, S. 34] kann die Geschwindigkeit wie folgt modelliert werden:

$$v_t = N(v_{t-1}, \sigma_v), \quad v_t = \begin{cases} v_t, & 0 \leq v_t \leq Max_v t \\ |v_t|, & v_t < 0 \\ Max_v t, & Max_v t < v_t \end{cases} \quad (4.8)$$

$$\sigma_v = \min(Max_{\Delta t}, \sqrt{\Delta t}) \quad (4.9)$$

Die maximale Geschwindigkeit $Max_v t$ wird auf $5 \frac{m}{s}$ gesetzt. N ist eine Gauß basierte Zufallszahl. Die maximale Zeitdifferenz $Max_{\Delta t}$ beträgt $3s$.

- **Richtung**

Die Richtung eines Partikels berechnet sich aus der letzten Richtung und einer wahrscheinlichen Richtungsänderung, die abhängig von der Geschwindigkeit ist. Es ist wahrscheinlich, dass sich die Richtung stärker ändert, je langsamer sich das Partikel bewegt [Wid10, S. 34f] [Bil12, S 39f]. Die Richtung kann wie folgt modelliert werden:

$$\alpha_t = N(\alpha_{t-1}, \sigma_h), \quad \alpha_t = \begin{cases} \alpha_t, & -\pi \leq \alpha_t \leq \pi \\ \alpha_t + 2\pi, & \alpha_t < -\pi \\ \alpha_t - 2\pi, & \alpha_t > \pi \end{cases} \quad (4.10)$$

$$\sigma_h = 0.4\pi - \arctan\left(\frac{\sqrt{v_{t-1}}}{2}\right) \quad (4.11)$$

α_t ist die Richtung zum Zeitpunkt t . Liegen ausreichend genaue Kompassdaten vor, kann die Richtung auch durch einen Kompass bestimmt werden.

- **Bewegung der Partikel**

Der Zustand eines Partikels x^i zum Zeitpunkt t , kann nun durch die Geschwindigkeit v_t (Gleichung 4.9) und der Richtung α_t (Gleichung 4.11) des Partikels bestimmt werden [Wid10, S. 35f]:

$$x_t^i = \begin{bmatrix} x_t^i \\ y_t^i \end{bmatrix} = \begin{bmatrix} x_{t-1}^i + v_t^i \cos(\alpha_t^i) \Delta t + \eta_t \\ y_{t-1}^i + v_t^i \sin(\alpha_t^i) \Delta t + \eta_t \end{bmatrix} \quad (4.12)$$

η_t beschreibt ein Rauschen basierend auf einer Gaußverteilung. Es wird benutzt um zu verhindern, dass sich die Partikel in einem Punkt sammeln. Das Bewegungsmodell beschreibt die Übergangswahrscheinlichkeit der Partikel vom letzten

in den aktuellen Zustand. Anschließend müssen noch die Messwerte der Sensoren einbezogen werden, um die A-posteriori-Wahrscheinlichkeit zu erhalten.

Messmodell

Mit dem Messmodell wird die A-posteriori-Wahrscheinlichkeit berechnet. Es basiert auf dem Vergleich von Fingerprints [Wid10]. Eine Messung z besteht aus den Signalstärken von WLAN oder Bluetooth Basisstationen zum Zeitpunkt t :

$$z_t = \{\vec{s}\} \quad (4.13)$$

Nun wird berechnet, wie viel Einfluss eine Messung auf den Zustand eines Partikels hat. Die Wahrscheinlichkeit $P(z_t|x_t)$ ergibt sich aus den Produkten der jeweiligen Messungen:

$$P(z_t|x_t) = \prod_{k \in z_t} P(z_t^k|x_t) \quad (4.14)$$

Bei der Auswertung von Signalstärken kann es zu verschiedenen Einflüssen kommen. Zum Beispiel kann der Messwert verrauscht sein. Das bedeutet, dass die Signalstärke am selben Ort natürlichen Schwankungen unterliegt. Ausserdem können Signale von Basisstationen ganz wegfallen, oder Signale von Basisstationen gemessen werden, die zuvor nicht da waren. Wie diese Einflüsse auf eine Messung in die Berechnung mit einbezogen werden können, soll im Folgenden beschrieben werden.

- **Messwert und Rauschen**

Die Signalstärke, die ein Sensor misst, kann durch Reflexion und Absorption an Wänden bzw. Personen oder durch ungenaue Sensoren verrauscht sein. So kann es zu verschiedenen Zeiten für ein und die selbe Position zu Ungenauigkeiten in der Messung kommen. Ist eine Basisstation sowohl in der Messung als auch im Fingerprint vorhanden, soll die Wahrscheinlichkeit für diese Messung berechnet werden. Zusätzlich soll ein Rauschen für diese Messung durch eine Gaußverteilung, mit einem Mittelwert von z_t^{+k} und einer Standardabweichung von σ_+ dargestellt werden [Wid10, S. 39]. Die Wahrscheinlichkeit P_r kann dann folgendermaßen berechnet werden.

$$P_r(z_t^k|x_t) = N(z_t^k, z_t^{+k}, \sigma_+) \quad (4.15)$$

$$= \frac{1}{\sigma_+ * \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{z_t^k - z_t^{+k}}{\sigma_+} \right)^2} \quad (4.16)$$

Dabei ist z_t^{+k} der gespeicherte Wert des Fingerprints und z_t^k der aktuell gemessene Wert der k -ten Basisstation. k ist ein Element aus der Menge aller Basisstationen, die sowohl im Fingerprint als auch im Messwert vorkommen: $k \in z_t \cap z_t^+$.

- **Fehlende Basisstationen**

Ein anderer Fall tritt ein, wenn eine Basisstation im Fingerprint existiert, aber nicht in der Messung auftaucht. Das kann entweder bedeuten, dass die Messung an einer anderen Position stattgefunden hat oder dass eine Basisstation ausgefallen ist. Die Wahrscheinlichkeit, dass eine Basisstation fehlt (P_f) kann ebenfalls mit einer Gauß-Verteilung modelliert werden [Wid10, S. 40].

$$P_f(z_t^k | x_t) = N(\eta, 0, \sigma_+) \quad (4.17)$$

$$= \frac{1}{\sigma_+ * \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{\eta}{\sigma_+} \right)^2} \quad (4.18)$$

k ist ein Element aus der Menge der Basisstationen im Fingerprint abzüglich der Basisstationen im Messwert: $k \in z_t^+ \setminus z_t$.

η ist ein Parameter für die fehlende Basisstation. Fällt z.B. eine Basisstation aus bei der zuvor eine hohe Signalstärke gemessen wurde, hat dies größeren Einfluss auf die Messung und η fällt auch dementsprechend größer aus. Auf der anderen Seite hat ein Ausfall einer Basisstation bei der zuvor eine geringe Signalstärke gemessen wurde, weniger starken Einfluss auf die Messung und dementsprechend fällt auch η geringer aus. η berechnet sich nach [Wid10, S. 45] wie folgt:

$$\eta = \begin{cases} (5/30) \cdot (z_t + 100), & z_t \leq -70 \\ (15/25) \cdot (z_t + 70) + 5, & -70 < z_t \leq -45 \\ 20, & -45 < z_t \end{cases} \quad (4.19)$$

- **Zusätzliche Basisstationen**

Der umgekehrte Fall tritt ein, wenn der Wert einer weiteren Basisstation k in der aktuellen Messung z_t auftaucht, aber nicht im Fingerprint z_t^+ vorkommt. Auch diese Wahrscheinlichkeit P_e kann mit einer Gaußverteilung dargestellt werden [Wid10,

S. 40f]:

$$P_e(z_t^k|x_t) = N(\eta, 0, \sigma_+) \quad (4.20)$$

$$= \frac{1}{\sigma_+ * \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{\eta}{\sigma_+}\right)^2} \quad (4.21)$$

k ist ein Element aus der Menge aller Basisstationen im Messwert abzüglich der Basisstationen im Fingerprint: $k \in z_t \setminus z_t^+$.

η berechnet sich ähnlich wie im Fall einer fehlenden Basisstation. Ist die Signalstärke einer zusätzlichen Basisstation sehr hoch, bekommt η einen hohen Wert, da es unwahrscheinlich ist, dass eine zuvor nicht erkannte Basisstation nun mit einer sehr hohen Signalstärke sendet. Wird hingegen eine zusätzliche Basisstation mit geringerer Signalstärke erkannt, dann bekommt η einen geringen Wert, weil es durchaus vorkommen kann, dass eine Basisstation mit geringer Signalstärke vorher nicht erkannt wurde.

Die gesamte Wahrscheinlichkeit einer Messung wird wie folgt berechnet [Wid10, S. 40f]:

$$P(z_t|x_t) = \left[\prod_{k \in K} P_{rauschen}(z_t^k|x_t) \right]^{\frac{1}{|K|}} \cdot \prod_{k \in L} P_{miss}(z_t^k|x_t) \cdot \prod_{k \in M} P_{extra}(z_t^k|x_t) \cdot \beta \quad (4.22)$$

$$\beta = \frac{|K|}{|K| + |L| + |M|} \quad (4.23)$$

K ist die Menge der Basisstationen, die sowohl im Fingerprint als auch in der aktuellen Messung vorkommen ($z_t \cap z_t^+$). L ist die Menge der Basisstationen, die im Fingerprint vorkommen aber nicht in der aktuellen Messung ($z_t^+ \setminus z_t$) und M ist die Menge der Basisstationen, die in der aktuellen Messung aber nicht im Fingerprint auftauchen ($z_t \setminus z_t^+$).

Bestimmung einer Position

Aus den Partikeln, welche die Wahrscheinlichkeitsdichte approximieren, soll nun ein eindeutiger Zustand, folglich eine Position ermittelt werden. Da die Partikel in einer Wolke in dem Zustandsraum verteilt sind, wird ein Konzept benötigt, um daraus eine eindeutige Position, genauer gesagt den Erwartungswert bestimmen zu können.

- Da die Partikel die Wahrscheinlichkeitsdichte repräsentieren, können die Zustände aller Partikel unter Berücksichtigung ihres Gewichtes summiert werden. [Obs09, S.

44].

$$\hat{x}_t = \sum_{i=1}^N w_t^i x_t^i \quad (4.24)$$

Der Zustand \hat{x}_t repräsentiert nun den Erwartungswert der Wahrscheinlichkeitsdichtefunktion. Die Position aus diesem Zustand ist dann die wahrscheinlichste Position.

- Eine andere Möglichkeit wäre, das Partikel mit dem größten Gewicht auszuwählen [Obs09, S. 44]:

$$\hat{x}_t = x_t^{i_{max}} \quad (4.25)$$

$$i_{max} = \max(w_t^i | i = 1, \dots, N) \quad (4.26)$$

Die Position des Partikels mit dem größten Gewicht, ist dann die gesuchte Position.

- Eine dritte Möglichkeit wäre jedem Partikel seinem nächsten Fingerprint zuzuordnen. Die Position des Fingerprint, welcher die meisten Partikel um sich vereint, ist dann wahrscheinlich die gesuchte Position.

5 Visuelle Erkennung von Objekten

In diesem Kapitel werden die Konzepte und Lösungsmöglichkeiten für eine visuelle Erkennung von Objekten beschrieben. An dieser Stelle soll zunächst eine Unterscheidung der Begriffe Erkennung und Identifizierung vorgenommen werden. Mit Erkennung ist gemeint, dass ein Objekt erkannt und richtig zugeordnet wird. Eine Identifizierung ermöglicht prinzipiell auch eine Unterscheidung gleicher Objekte z.B. durch Auslesen einer eindeutigen Identifikationsnummer. Im Folgenden geht es um eine Erkennung von Objekten. Eine Unterscheidung gleich aussehender Objekte kann z.B. durch eine räumliche Trennung vorgenommen werden. Für die visuelle Erkennung soll das SIFT Verfahren angewendet werden. Dieses Kapitel beschreibt die Einflüsse auf die visuelle Erkennung und die Algorithmen mit denen die Objekte erkannt werden können.

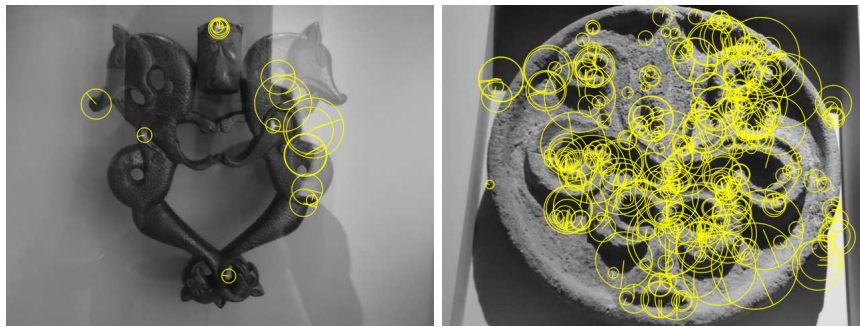
5.1 Einfluss auf die visuelle Erkennung

Form

Grundsätzlich sind flache Objekte gegenüber runden Objekten vorzuziehen. Flache Objekte haben nur endlich viele Seiten die zu betrachten sind. So hat ein Buch zwei bis drei Seiten: Titel- und Rückseite, sowie den Buchrücken. Die restlichen drei Seiten eines Buches sind informationslos. Beispielsweise bei einem Würfel wären es sechs Seiten. Bei runden Formen wie z.B. einem Zylinder gibt es potenziell unendlich viele Seiten, von denen das Objekt betrachtet werden kann.

Kontrast

Je mehr Übergänge es von dunklen zu hellen Bildbereichen gibt, desto eher können Schlüsselpunkte gefunden werden. Abbildung 24 zeigt, dass die Anzahl der Schlüsselpunkte abhängig von der Struktur der Objekte ist. Objekte mit monotonen Farbverläufen und wenig Struktur weisen weniger Schlüsselpunkte auf als Objekte mit stärkeren Farbverläufen oder gröberer Struktur. Jedoch sagt die Anzahl der Schlüsselpunkte noch nichts über die Qualität der Erkennung aus.



(a) Wenige Schlüsselpunkte

(b) Viele Schlüsselpunkte

Abbildung 24: Die Anzahl der Schlüsselpunkte ist abhängig von der Struktur des Objektes. Bildquelle: Pergamonmuseum Berlin - Museum für Islamische Kunst.

Anzahl der Objekte

Bei der Erstellung der Objektmerkmale sollte beachtet werden, dass nur dieses Objekt im Fokus der Kamera ist (siehe Abbildung 25(a)), da sonst Bildmerkmale von weiteren Objekten berechnet werden die nicht dazu gehören. Jedoch ist es nicht immer möglich nur ein einziges Objekt in den Fokus der Kamera zu bekommen. Das kann dann passieren, wenn man nicht dicht genug an das Objekt herankommt, wie es z.B. bei Objekten hinter Vitrinen der Fall ist (siehe Abbildung 25(b)).



(a) Einzelnes Objekt

(b) Gruppe von Objekten

Abbildung 25: Aufnahme eines Objektes. Auf der linken Seite füllt die blaue Vase den Fokus der Kamera optimal aus. In der rechten Abbildung ist die blaue Vase zusammen mit anderen Objekten zu sehen. Bildquelle: Pergamonmuseum Berlin - Museum für Islamische Kunst.

Schwierigkeiten bei der Objekterkennung

Gerade in Museen befinden sich viele Objekte zum Schutz hinter Vitrinen. Hier ist es meist nicht möglich nahe an die Objekte heranzukommen. Durch die Glasscheibe kann es zusätzliche zu Unschärfe und Reflexionen kommen, welche die Bilderkennung beeinflussen können. Aufgrund des größeren Abstandes ist es nicht immer möglich, nur das gewünschte Objekt in den Fokus der Kamera zu bekommen. Folglich können Merkmale nicht mehr nur auf dem gewünschten Objekt, sondern auf mehreren verschiedenen Objekten berechnet werden. Bei der Erstellung der Objekt-Bilder muss man sich fragen, wie nahe kommt der Benutzer an das Objekt heran. Ausserdem verfügen die meisten mobilen Geräte nicht über eine optische Zoom Funktion.

Auflösung der Bilder

Bevor Objekte erkannt werden können, müssen auf den aufgenommenen Bildern die Merkmale berechnet werden. Je mehr potenzielle Merkmale berechnet werden, desto eher ist es wahrscheinlich, dass das Objekt in einem Bild erkannt wird. Dass die Auflösung entscheidend zur Anzahl der gefundenen Schlüsselpunkte beiträgt, zeigt Tabelle 3. Diese zeigt ein Objekt, das in verschiedenen Auflösungen aufgenommen wurde. Je höher die Auflösung des Bildes ist, desto mehr Schlüsselpunkte können gefunden werden. Damit verlängert sich jedoch die Berechnungszeit.


Bild	Auflösung (Mega Pixel)	Berechnete Schlüsselpunkte	Zeit (s)
	0.3	2816	0.7
	1	5239	1.7
	2	9356	3.2
	3	11841	5.8
	5	20415	8.1

Tabelle 3: Einfluss der Auflösung eines Bildes auf die Anzahl der Schlüsselpunkte.

5.2 Objekterkennung: Nächste Nachbar Suche

Die Objekterkennung erfolgt durch das Vergleichen der Deskriptoren (Vektor von Merkmalen um einen Schlüsselpunkt) in zwei Bildern. Dazu wird zu jedem Deskriptor in einem Bild der ähnlichste Deskriptor (nächste Nachbar) in einem anderen Bild gesucht. Es wird also zu jedem Deskriptor aus Bild A ein passender Deskriptor in Bild B gefunden, vorausgesetzt Bild B hat mindestens ein Schlüsselpunkt. Der nächste Nachbar

wird durch den Abstand zweier Deskriptoren, z.B. mit dem euklidischen Abstand, bestimmt. Der nächste Nachbar eines Deskriptors D_i aus Bild A ist dann der Deskriptor D_n aus Bild B, welcher den kürzesten Abstand zu D_i hat. Der nächste Nachbar zu einem Deskriptor wird auch als Match bezeichnet. Die Summe der Abstände aller gefundenen Matches ist dann das Ergebnis aus dem Vergleich zweier Bilder. Wenn zwei Bilder das selbe Objekt zeigen, dann ist es wahrscheinlicher, dass die Summe der Abstände der Matches kleiner ist, da hier die Deskriptoren ähnlicher zueinander sind als bei Bildern, welche unterschiedliche Objekte zeigen. Das Ergebnis der Objekterkennung ist dann das Bild, bei dem die kleinste Summe der Abstände der Matches berechnet wurde. Das Problem bei diesem Verfahren ist, dass die Abstandssumme keine eindeutige Aussage über ein gefundenes Objekt machen kann. Es kann durchaus Bilder mit anderen Objekten geben, die dennoch kürzere Abstände berechnen, als Bilder mit demselben Objekt. Da ein Bild A mit n Schlüsselpunkten verglichen zu Bild B mit Null Schlüsselpunkten keine Matches haben kann, ist somit die Summe der Abstände gleich null. Zusätzlich wird es immer ein Bild geben zu dem die Abstände am geringsten. Dieses Verfahren ist also nur bedingt für eine Objekterkennung geeignet, da es hier durchaus zu einer Erkennung falscher Objekte kommen kann.

Auswahl stabiler Matches

Statt nur den einen nächsten Nachbarn zu suchen, sollen nun die k -nächsten Nachbarn gesucht werden. Somit soll ermöglicht werden stabile Matches zu bekommen. Für alle Deskriptoren in Bild A sollen die k -nächsten Nachbarn in Bild B gesucht werden. Das selbe wird für die umgekehrte Richtung wiederholt. In Bild B werden die k -nächsten Nachbarn aus Bild A gesucht. Für jeden Deskriptor aus dem einen Bild existieren nun k -nächste Nachbarn aus dem anderen Bild. Im Folgenden sollen nur die zwei nächsten Nachbarn gesucht werden, also $k = 2$.

Die Qualität der k -nächsten Nachbarn soll mittels eines Verhältnis-Tests überprüft werden. Der Verhältnis-Test prüft die Ähnlichkeit des nächsten und des zweit-nächsten Nachbarn. Ob zwei Deskriptoren als ähnlich gelten, wird durch einen Schwellwert (engl. threshold) definiert. Der Schwellwert ist im Intervall $[0, 1]$ definiert. Sind der nächste und der zweit-nächste Deskriptor sehr ähnlich, kann nicht garantiert werden, dass der nächste Deskriptor wirklich der richtige ist. Daher können diese Matches nicht weiter betrachtet werden. Besteht jedoch eine geringe Ähnlichkeit zwischen dem nächsten und dem zweit-nächsten Deskriptor, kann angenommen werden, dass der nächste Deskriptor

qualitativ ein Kandidat für einen stabilen Match ist.

Der zweite Test untersucht, ob ein Kandidat für einen stabilen Match, symmetrisch ist. Das bedeutet, dass beide Deskriptoren den jeweils anderen Deskriptor als nächsten Nachbar haben müssen. Abbildung 26 zeigt, dass Deskriptor L aus Bild A als nächsten Nachbarn Deskriptor M in Bild B hat. Deskriptor M hat aber nicht L, sondern N

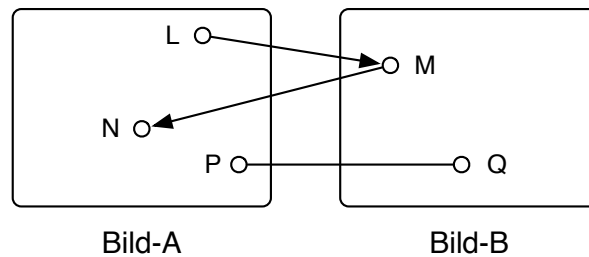


Abbildung 26: Symmetrie Test.

als nächsten Nachbarn. Die Matches der beiden Deskriptoren L und M sind also nicht symmetrisch und werden verworfen. Anders sieht es bei den Deskriptoren P und Q aus. Beide haben den jeweils anderen als nächsten Nachbarn. Der Match (P, Q) ist also symmetrisch.

Herausforderungen der Nächsten Nachbar Suche

Eine lineare Suche nach dem nächsten Nachbarn in allen Referenz-Bildern kann zu einem hohen Rechenaufwand führen. Bei n Deskriptoren im Anfrage-Bild und m Deskriptoren in allen Referenz-Bildern ergibt das $N \cdot M$ Berechnungen. Da in jedem Bild abhängig von der Konfiguration hunderte bis tausende Schlüsselpunkte gefunden werden, kann eine Suche nach dem passenden Bild sehr viel Zeit in Anspruch nehmen. Diesen Aufwand gilt es zu reduzieren. Eine Möglichkeit den Aufwand zu verringern wäre es, eine Signatur für jeden Merkmals-Vektor zu berechnen. Mit Hilfe dieser Signatur könnte der Suchraum eingeschränkt werden [Pre07, S. 96]. Ein anderes Verfahren wäre den wahrscheinlich nächsten Nachbarn durch heuristische Methoden zu ermitteln. Durch verringerten Rechenaufwand würde diese Methode ermöglichen schnell zu einem akzeptablen Ergebnis zu kommen [Pre07, S. 96]. Eine weitere Möglichkeit den Suchraum einzuschränken wäre, das Wissen über den aktuellen Aufenthaltsort zu verwenden. Wenn man weiß, an welchem Standort sich ein Objekt befindet, kann durch eine Lokalisierung der Suchraum soweit eingeschränkt werden, dass sehr viel weniger Vergleiche gemacht werden müssen.

5.3 Match-Scores

Mittels dem Verhältnis- und Symmetrie-Test können nun stabile Matches in einem Bild identifiziert werden. Die Wahrscheinlichkeit, dass zwei symmetrische Matches auf unterschiedlichen Objekten gefunden werden, ist nun geringer aber immer noch möglich. Bisher wurde angenommen, dass das Bild mit den meisten stabilen Matches das Objekt darstellt, was gesucht ist. Diese Annahme kann aber immer noch zu falschen Ergebnissen führen. Daher wäre es sinnvoll, eine qualitative Aussage über die Güte des Bildes zu machen, auf dem die meisten Matches gefunden worden sind. Daher soll nun ein Match-Score eingeführt werden. Der Score beschreibt das Verhältnis der Anzahl der Matches im besten Bild mit der Anzahl der Matches aus anderen Bildern. Der Score ist definiert für den Bereich $[0, 1]$. Je höher der Score für ein Bild ist, desto höher ist die Qualität der Matches in diesem Bild. Je kleiner der Score wird, desto niedriger ist die Qualität der Matches im Bild, da in anderen Bildern ebenfalls stabile Matches gefunden worden sind.

Wenn für ein Objekt mehrere Referenzbilder existieren, dann wird der Match-Score nicht durch Bilder des selben Objektes beeinflusst. Denn diese können natürlich ähnlich zueinander sein und sind es wahrscheinlich auch. Ein Bild wird also nur mit Referenzbildern von anderen Objekten verglichen.

6 Systementwurf

In diesem Kapitel wird der Systementwurf der Anwendung beschrieben. Dabei soll das Datenmodell sowie die einzelnen Komponenten allgemein in ihren Funktionen erläutert werden.

6.1 Datenschicht

Datenmodell

Das Datenmodell beschreibt die Entitäten, die für diese Arbeit benötigt werden. Die wichtigsten Entitäten sind Objekt, Raum und Fingerprint. Jeder Raum kann Objekte und Fingerprints enthalten. Weitere Entitäten sind Features, Gebäude sowie die verwendeten Sensoren WLAN und Bluetooth. Jedes Objekt hat Features (Merkmale), die für die visuelle Objekterkennung benötigt werden. Ein Fingerprint enthält die Daten der Messungen der jeweiligen Sensoren. Ein Raum ist Teil eines Gebäudes. Die Abbildungen 57 und 58 im Anhang ab Seite II zeigen das ER-Modell.

Datenbasis

Damit die Anwendung funktionieren kann, müssen bestimmte Daten wie Gebäude, Räume und Objekte eingepflegt werden. Dies könnte z.B. über ein Web-Interface auf einem zentralen Server erfolgen. Eine berechtigte Person kann hier neue Objekte erstellen und diese den Räumen an einer bestimmten Position zuweisen. Für die Zeit der Entwicklung reicht es jedoch aus, die Daten lokal auf dem mobilen Testgerät zur Verfügung zu stellen. Konfigurationsdateien sollen dazu verwendet werden, um die Datenbasis einfach auf das Testgerät zu überspielen. Grundlegende Daten wie Gebäude, Räume und Objekte, die nicht dynamisch zur Laufzeit generiert werden, sollen in Konfigurationsdateien definiert werden. Bei der Initialisierung der Datenbank können die Daten aus den Konfigurationsdateien in die Datenbank übertragen werden.

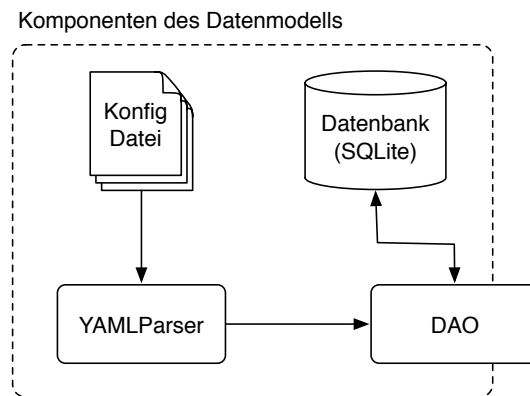


Abbildung 27: Komponenten der Datenmodells.

Datenzugriff

Der Zugriff auf die Datenbank erfolgt über eine Data Access Object (DAO) Schnittstelle. Die DAO kapselt den Aufbau und Kommunikation mit der Datenbank. Sie bietet Methoden zum Hinzufügen und Abfragen von Daten. Damit bleibt die interne Datenbankstruktur verborgen (*information hiding*).

6.2 Lokalisierung

Die Lokalisierung enthält Komponenten für die Aufnahme der Sensordaten und der Fingerprints (Record Fingerprint), für eine Simulation zum Testen der Lokalisierung und einer aktiven Lokalisierung (Partikel Filter) wie Abbildung 28 zeigt. Alle diese Kom-

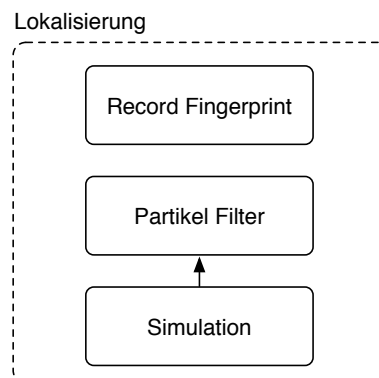


Abbildung 28: Komponenten der Lokalisierung.

ponenten benötigen eine Karte, auf der die aktuelle Umgebung dargestellt wird. Für die Aufnahme der Fingerprints wird die aktuelle Position auf einer Karte mit einer Touch-Geste angegeben. Vor Aufzeichnung der Sensordaten an dieser Position, wird die jeweilige Richtung angegeben sowie die gewünschten Sensoren ausgewählt wie in Abbildung 29 zu sehen ist. Die Messdaten werden nach der Messung automatisch gespeichert. Die eigentliche Lokalisierung wird vom Partikel Filter übernommen. Dieser wertet die

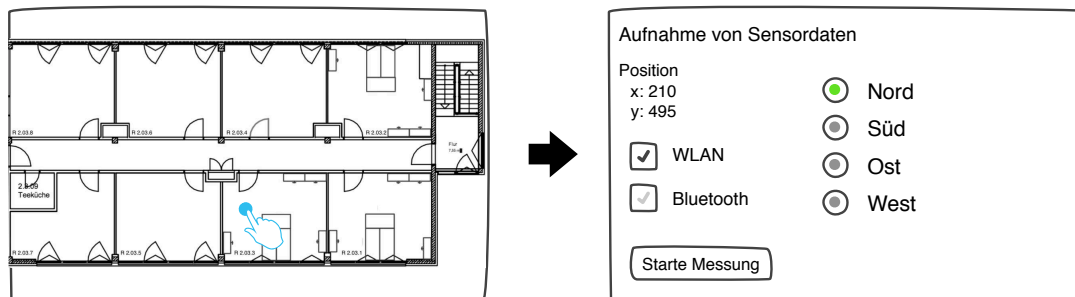


Abbildung 29: Auf einer Karte wird eine Position mit einer Touch-Geste bestimmt (links). Dann kann eine Messung für verschiedene Richtungen an dieser Position gemacht werden.

Daten der Sensoren aus und berechnet daraus eine wahrscheinliche Position. Um vergleichbare Werte zu erhalten und nicht jedes mal durch die Räumlichkeiten gehen zu müssen und um Zeit zu sparen, sollen Simulationsdaten zur Verfügung stehen. Diese Daten sollen einmalig aufgezeichnet und unabhängig von der Datenbank im JavaScript Object Notation (JSON) Format gespeichert werden. Um die Daten nach der Reihenfolge der Aufzeichnung sortieren zu können, wird jedem Datensatz ein Zeitstempel (engl. timestamp) zugewiesen. Alle Daten beziehen sich dabei auf die dritte Etage im zweiten Gebäude des TGS. In Listing A.1 (Anhang Seite I) ist das Schema zu sehen, wie die Simulationsdaten gespeichert werden. Das JSON Format ermöglicht eine einfache Auswertung durch Maschinen und Menschen.

6.3 Visuelle Erkennung

Die visuelle Erkennung teilt sich in zwei Komponenten auf. Die erste Komponenten (Record Features) dient zur Aufnahme von Bildern zu einem Objekt und der Berechnung von Merkmalen aus diesen Bildern. Abbildung 30 zeigt die View für die Aufnahme der Merkmale zu einem Objekt. Für die Aufnahme eines Bildes zu einem Objekt wählt ein Autor zuerst das Gebäude aus. Je nachdem welches Gebäude gewählt wurde, wird

The screenshot shows a software interface with a central area labeled 'Vorschau Foto'. To the right of this area are three dropdown menus: 'Auswahl Gebäude' with 'TGS' selected, 'Auswahl Raum' with '2.3.03' selected, and 'Auswahl Objekt' with 'HP-Drucker' selected. Below the preview area are three buttons: 'Aufnahme Foto', 'Merkmale speichern', and 'Bild speichern'.

Abbildung 30: Aufnahme von Bildern zu einem Objekt und Berechnung der Merkmale.

anschließend eine Liste mit Räumen für dieses Gebäude geladen. Danach wählt der Autor einen Raum aus, was wiederum eine Liste mit Objekten lädt, die sich in diesem Raum befinden. Zuletzt wählt er das Objekt aus, zu dem das Bild aufgenommen werden soll. Die Daten (Gebäude, Raum, Objekt) müssen schon in der Datenbank vorhanden sein. Nun kann der Autor ein oder mehrere Bilder zu diesem Objekt aufnehmen. Nach der Aufnahme werden sofort die Schlüsselpunkte und Deskriptoren auf dem Bild berechnet. Die Schlüsselpunkte werden mit dem Bild als Vorschau dem Autor angezeigt. Dieser kann nun entscheiden, ob die Aufnahme gut ist und seiner Meinung nach ausreichend Schlüsselpunkte berechnet worden sind. Die Anzahl der Schlüsselpunkte sagt noch nichts darüber aus, ob damit eine gute Erkennung möglich ist, kann aber bereits ein Hinweis sein. Wenig bis keine Schlüsselpunkte werden wahrscheinlich keine gute Erkennung ermöglichen. Wenn der Autor mit der Aufnahme einverstanden ist, kann er diese speichern. Dabei werden jedoch nur die Schlüsselpunkte und die Deskriptoren gespeichert, nicht das Bild selber. Damit ein Benutzer der Anwendung sehen kann, welches Objekt erkannt worden ist, kann der Autor ein ausgewähltes Bild als Beispiel für dieses Objekt hinterlegen.

Die zweite Komponente (Object Tracking) sorgt für die Erkennung der Objekte durch Vergleich eines Testbildes mit einer Menge von Referenzbildern. Dazu nimmt der Benutzer ein Bild von einem Objekt auf. Dieses Bild wird mit einer Menge von gespeicherten Bildern zu verschiedenen Objekten verglichen. Das Ergebnis ist das Objekt, das diesem Bild am ähnlichsten ist.

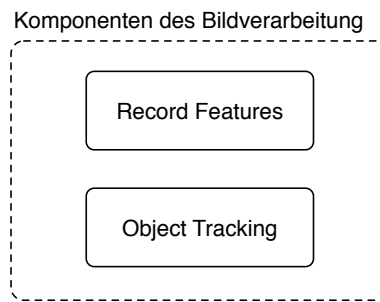


Abbildung 31: Komponenten der Objekterkennung.

Überblick aller Komponenten

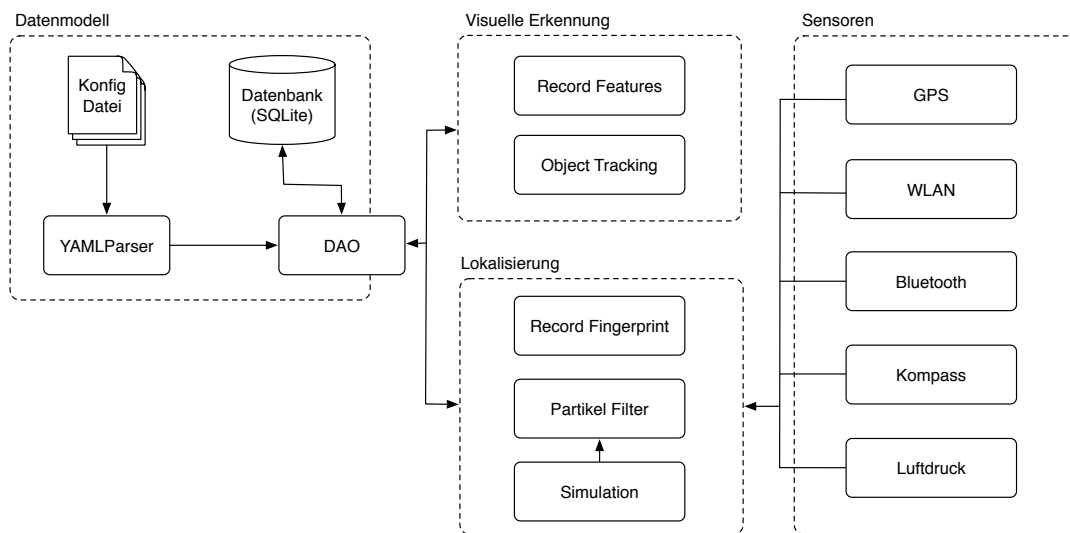


Abbildung 32: Zusammenspiel aller Komponenten.

7 Implementierung

In diesem Kapitel wird die konkrete Umsetzung der Lokalisierung und der visuellen Objekterkennung beschrieben. Zuerst hat sich die Frage nach der mobilen Plattform gestellt. Grundsätzlich standen drei Plattformen zur Auswahl: iOS (Apple), Android (Google) und Windows Phone (Microsoft). Da Android einen offenen Zugang zu der WLAN und Bluetooth API bietet, im Gegensatz zu iOS, und weil für Android geeignete Hardware zur Verfügung stand, fiel die Wahl der Entwicklungsplattform auf Android. Die Anwendung wird auf einem Motorola XOOM Tablet entwickelt und getestet, welches mit Android Version 3.2¹⁶ läuft. Die Entwicklung und das Deployment wird durch die Eclipse IDE unterstützt. Android wird in der Programmiersprache Java geschrieben. Für die visuelle Erkennung wurde die freie Bibliothek OpenCV [Lag11][27] verwendet, welche eine sehr große Auswahl an Algorithmen aus dem Bereich Bildverarbeitung bietet. Da OpenCV ursprünglich für C++ entwickelt wurde, besteht die Möglichkeit C++ Code einzubinden und auszuführen. Das hat den Vorteil, dass bereits vorhandener C++ Code wiederverwendet werden kann und nicht in Java neu geschrieben werden muss. Rechenintensive Prozesse können durch eine native Ausführung beschleunigt werden. Möglich wird das mit dem Android Native Development Kit (NDK)¹⁷, welches es erlaubt nativen C und C++ Code auf Android Geräten zu kompilieren. Die Kommunikation von C/C++ und Java erfolgt mit dem Java Native Interface (JNI) [28]. Jedoch sollte darauf geachtet werden, dass ein Aufruf einer nativen Methode ebenfalls Zeit kostet und mit bedacht angewendet werden sollte. Dieses Kapitel teilt sich in die Abschnitte: Datenbank, Lokalisierung und Visuelle Objekterkennung auf, welche die wichtigsten Komponenten der Anwendung darstellen.

¹⁶Android API 3.2; API Level: 13; Versionsname: Honeycomb

¹⁷Infos zum NDK: <http://developer.android.com/sdk/ndk/overview.html>

7.1 Datenbank

7.1.1 Initialisierung & Konfiguration

Grundlegende Daten, die für den Prototypen benötigt werden wie Gebäude, Räume und Objekte, sollen über Konfigurationsdateien in die Datenbank geschrieben werden. Die Konfigurationsdateien erlauben eine einfache Erweiterung der Datenbasis. Diese sind in der Auszeichnungssprache *YAML* geschrieben. Im Anhang A.3 auf Seite III sind Auszüge der Konfigurationsdateien abgebildet. Die Konfigurationsdateien werden aus dem Ordner `assets/db-config` gelesen und in die Datenbank geschrieben. Mit Hilfe der *YAML* Bibliothek `snakeyaml`¹⁸ wurde ein *YAML* Parser geschrieben, der die Werte aus der Konfigurationsdatei einliest. Anschließend werden die gelesenen Werte aus der Konfigurationsdatei in einer Liste der DAO übergeben, welche die Daten in die entsprechenden Tabellen der Datenbank einträgt. Sollte es zu Konflikten mit bereits existierenden Daten kommen, werden die neuen Daten nicht übernommen, sondern die alten Daten bleiben erhalten. Jede Tabelle hat ihre eigene Konfigurationsdatei. Sollen neue Einträge in die Datenbank übernommen werden, können diese einfach in der entsprechenden Konfigurationsdatei hinzugefügt werden. Änderungen an bestehenden Einträgen sind über die Konfigurationsdateien jedoch nicht möglich und müssten direkt in der Datenbank gemacht werden. Bei einer sorgfältigen Konfiguration sollte jedoch keine Notwendigkeit eines direkten Datenbankeingriffs bestehen. Die Konfigurationsdateien haben den Vorteil, dass sie leichter zu lesen und zu verwalten sind und somit nicht direkt in der Datenbank gearbeitet werden muss. Die Initialisierung der Daten kann auf beliebigen Geräten erfolgen.

7.1.2 Backup der Daten

Da sowohl die aufgezeichneten Objekt Merkmale als auch die Fingerprints der Sensoren sich nur auf dem Testgerät befinden und nicht an einen zentralen Server gesendet werden, wurde über eine Backup Strategie nachgedacht. Sowohl die Datenbank als auch das Verzeichnis mit den gespeicherten Schlüsselpunkten und Deskriptoren soll gesichert werden. Dies kann mit einem Datenbank-Dump durchgeführt werden. Dabei werden alle Daten in den Tabellen als *SQL* Anweisung gespeichert, sodass nach Ausführen der *SQL* Anweisung alle Daten wieder zur Verfügung stehen. Da es auf *Android* Geräten ohne *Root*-Rechte aus Sicherheitsgründen nicht möglich ist, von außen auf die Datenbank zuzugreifen, muss die Datenbank innerhalb der Anwendung zuerst auf die *SD-Karte* ko-

¹⁸Projektwebsite `snakeyaml`: <http://code.google.com/p/snakeyaml/>

piert werden. Anschließend kann die Datenbank sowie das Verzeichnis mit den Objekt Merkmalen auf einen Rechner kopiert werden. Der Datenbank-Dump sowie die Objekt Merkmale werden in den *assets* Ordner kopiert. Die Daten in diesem Ordner werden bei der Installation der Anwendung vom PC auf das Gerät mit übertragen und stehen der Anwendung zur Verfügung. Kommt es zu einem Verlust von Daten wird die SQL Anweisung des Datenbank-Dump ausgeführt und die Objekt Merkmale in das entsprechende Verzeichnis kopiert. Der Datenbank-Dump und die Objekt Merkmale werden zusätzlich in das Entwicklungs-Repository eingetragen und sind somit ebenfalls vor Datenverlust auf dem Entwicklungsrechner geschützt.

7.2 Lokalisierung

7.2.1 Gebäude Check-In

Sobald die Anwendung gestartet wird, wird die letzte bekannte GPS Position abgefragt. Diese wird mit den gespeicherten GPS Positionen aller Gebäude in der Datenbank verglichen. Das Gebäude, das am nächsten an der aktuellen Position dran ist, wird dem Benutzer vorgeschlagen. Der Benutzer kann entscheiden, ob der Vorschlag korrekt ist. Ist das Gebäude falsch, kann der Benutzer das Gebäude aus einer Liste auswählen. Die Gebäude in der Liste sind aufsteigend nach der Entfernung zum aktuellen Standort sortiert.

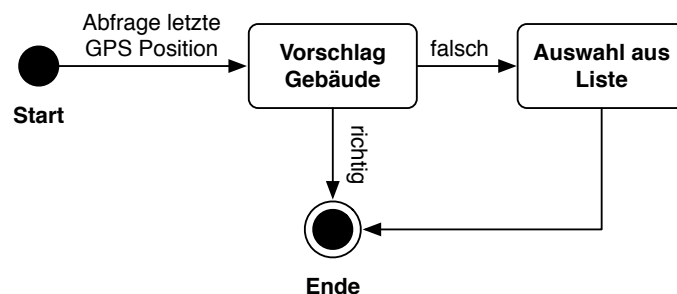


Abbildung 33: Zustandsdiagramm. Auswahl des aktuellen Gebäudes.

7.2.2 Vertikale Positionsänderung

Mit einem Luftdruck-Sensor soll eine vertikale Positionsveränderung festgestellt werden. Es soll keine konkrete Position ermittelt, sondern nur eine aktuelle Veränderung der

vertikalen Lage bestimmt werden. Wie in Abschnitt 3.2 beschrieben unterliegt der Luftdruck natürlichen Schwankungen. Jedoch kann in kurzen Zeitabständen der Luftdruck auf Veränderungen untersucht werden. Dazu wurde eine Klasse `Pressure` geschrieben. Die Klasse `Pressure` misst in Abständen von einer Minute den Luftdruck. Wenn der Betrag der Differenz des aktuellen Messwertes und des letzten Messwert größer ist als der Schwellwert von $\Delta 0,5hPa$, hat sich eine Veränderung in der vertikalen Position ereignet. Bisher wurde jedoch nur festgestellt, dass sich eine vertikale Bewegung ereignet hat, jedoch nicht in welche Richtung, also nach oben oder nach unten. Dazu wird der Wert der aktuellen Messung mit der letzten Messung verglichen. Ist der Wert der aktuellen Messung kleiner als die der letzten Messung, fand eine Bewegung nach oben statt, da der Luftdruck mit der Höhe abnimmt. Umgekehrt fand eine Bewegung nach unten statt, wenn der aktuelle Messwert größer als der letzte Messwert ist. Da sich pro Etage eine Messdifferenz von $0,5hPa$ ergeben hat, ergibt sich die Anzahl der zurückgelegten Etagen aus dem Vielfachen von $0,5hPa$.

7.2.3 Bewegungsmessung

Um zu ermitteln, ob eine Bewegung statt fand, wird der Beschleunigungssensor permanent ausgewertet. Der Beschleunigungssensor misst die Beschleunigung auf drei Achsen. Das Koordinatensystem bezieht sich dabei auf das mobile Gerät. Wird das mobile Gerät gedreht, dann wird auch das Koordinatensystem gedreht. Es handelt sich hierbei um ein lokales Koordinatensystem (siehe Abbildung 34). D.h. je nachdem in welcher Lage

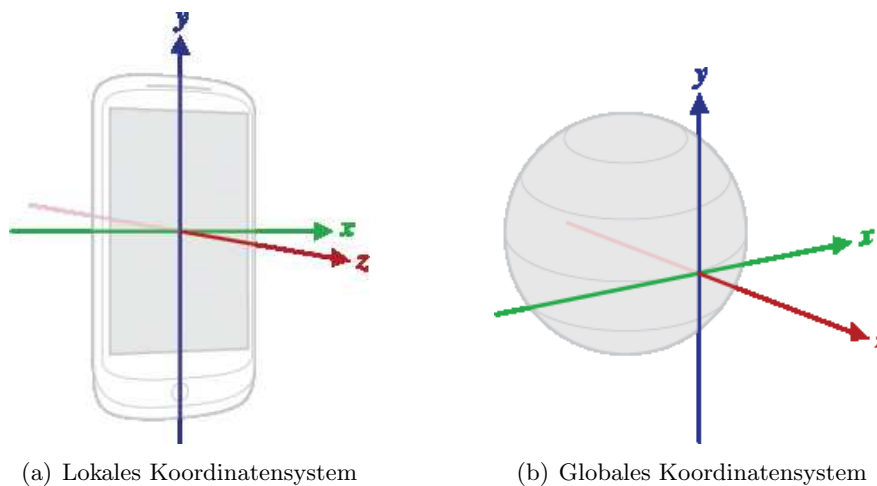


Abbildung 34: Koordinatensysteme in Android [24].

sich das mobile Gerät befindet, beziehen sich die Messwerte auf unterschiedliche Achsen. Da nicht vorhergesagt werden kann in welcher Position ein Benutzer sein mobiles Gerät hält, ist es nicht möglich die Messwerte auf den Achsen richtig zu interpretieren. Daher ist es notwendig, dass lokale Koordinatensystem in einer globales Koordinatensystem zu transformieren. Unabhängig von der Lage des mobilen Gerätes werden die Messwerte nun stets auf den selben Achsen wiedergegeben. Beschleunigungen in der vertikalen werden auf der z-Achse und Beschleunigungen in der horizontalen auf der x- und y-Achse dargestellt. Die Beschleunigungsdaten werden im Sekundentakt ausgelesen. In einer Queue werden die letzten 60 Beschleunigungswerte gespeichert. Wenn die Queue mit 60 Einträgen gefüllt ist, wird der älteste Eintrag entfernt, bevor ein neuer hinzugefügt wird. Nun kann in einem regelmäßigen Intervall geprüft werden ob z.B. auf der z-Achse eine Beschleunigung stattgefunden hat und wie stark diese war. Dabei wird der maximale Wert auf der z-Achse gespeichert. Wenn dieser Wert größer als der definierte Schwellwert von $11m/s^2$ ist, hat sich eine Beschleunigung nach oben ereignet. Nun kann davon ausgegangen werden, dass sich die Person zusammen mit dem mobilen Gerät bewegt hat.

7.2.4 Aufnahme der Sensordaten und Erstellung von Fingerprints

Der erste Schritt zur Umsetzung der Lokalisierung besteht in der Aufnahme der Sensordaten. Dabei werden wie in Abschnitt 4.2.1 beschrieben die Signalstärken von WLAN und Bluetooth an bestimmten Punkten gemessen. Der Versuchsaufbau befindet sich auf der dritten und zweiten Etage des TGS. Jede Sensoraufzeichnung wird einem konkreten Fingerprint zugeordnet. Ein Fingerprint ist einem konkreten Raum zugeordnet und beschreibt zusätzlich eine absolute Position auf einer Karte. Durch einen Touch des Benutzers auf die Karte wird die Position gespeichert. Die Position wird in Pixel angegeben. Nachdem die Position für den Fingerprint angegeben worden ist, wechselt die Ansicht zu einer anderen Activity¹⁹, in der die Aufzeichnung der WLAN und Bluetooth Signalstärken erfolgen kann. Da die Lokalisierung später auch eine symbolische Position liefern soll, muss der Raum angegeben werden, zu dem dieser Fingerprint gehört. Vor jeder Aufzeichnung muss die Richtung angegeben werden, in der die Person, welche die Aufzeichnung macht, schaut. Anschließend kann die Aufzeichnung der Signalstärke für WLAN und Bluetooth durchgeführt werden. Die Aufzeichnung der Signalstärken wurde in je einen Service für WLAN und einen für Bluetooth ausgelagert. In einem Service können Berechnungen unabhängig von einer konkreten Activity durchgeführt werden.

¹⁹Eine Activity stellt Elemente auf dem Display dar und ist für die Interaktion mit dem Benutzer verantwortlich.

Die RecordFingerprintActivity ist für die Aufzeichnung der Sensordaten verantwortlich.

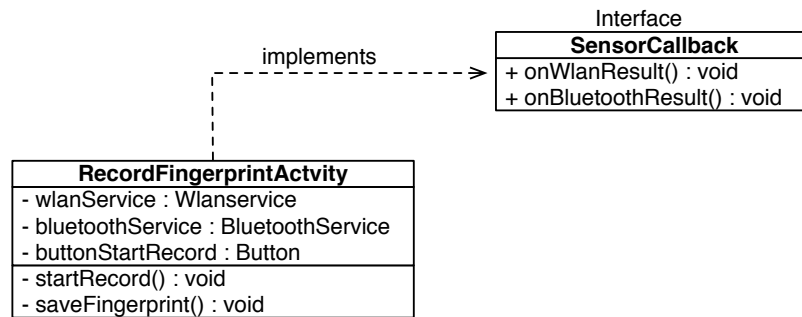


Abbildung 35: Klassendiagramm RecordFingerprintActivity.

Sie stellt alle Views, Text- und Button-Elemente auf dem Bildschirm dar und nimmt die Eingaben des Benutzers entgegen. Außerdem bindet sie sich an den entsprechenden Service. Der Service scannt die umliegenden WLAN bzw. Bluetooth Netzwerke und teilt das Ergebnis der RecordFingerprintActivity mit. Abbildung 35 zeigt ein Klassendiagramm der RecordFingerprintActivity. Um die Ergebnisse der WLAN und Bluetooth Messung

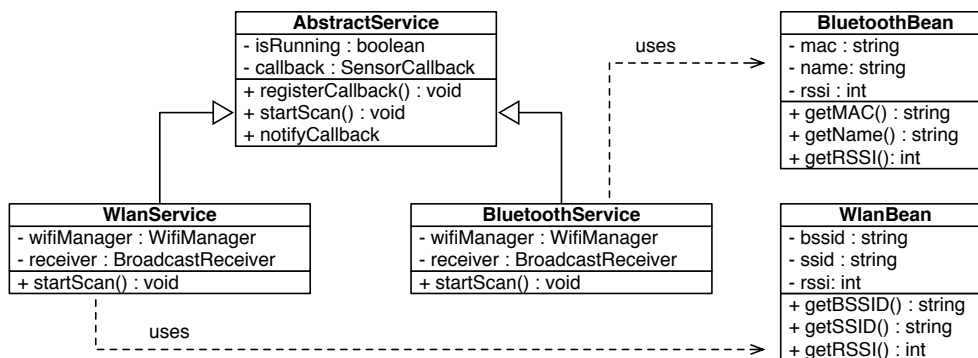


Abbildung 36: Klassendiagramm Service und Beans.

zu speichern wurden Beans geschrieben, welche die Attribute kapseln. Mit Hilfe der Beans ist eine einfache Übertragung der Daten zwischen verschiedenen Instanzen wie z.B. einem Service, einer Activity und der Datenbank möglich. Abbildung 36 zeigt die Services und die Beans für den Austausch der Sensordaten. Die Services übergeben eine Liste von WLAN bzw. Bluetooth Beans als Ergebnis an die RecordFingerprintActivity. Anschließend werden die ermittelten WLAN und Bluetooth Beans der DAO übergeben und in die Datenbank gespeichert. Abbildung 37 stellt den Ablauf vereinfacht in einem Sequenzdiagramm dar. Da es sinnvoll ist zu wissen, wo bereits Fingerprints erstellt wor-

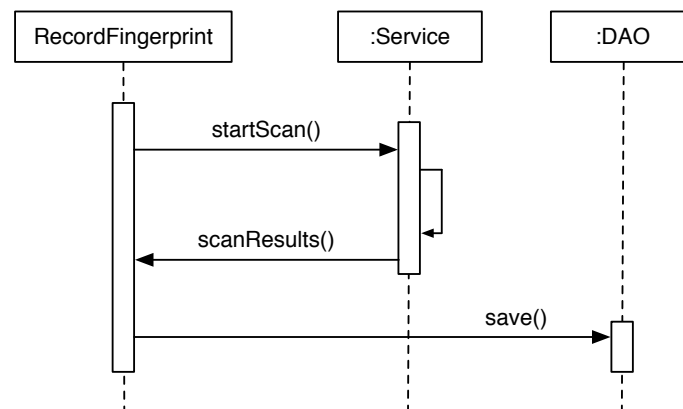


Abbildung 37: Sequenzdiagramm der Aufnahme der Fingerprints.

den sind, können diese auf der Karte eingblendet werden wie in Abbildung 22 auf Seite 50 zu sehen ist.

7.2.5 Partikel-Filter

Für den Partikel-Filter wurde eine gleichnamige Klasse entworfen. Diese Klasse ist für die Initialisierung der Partikel zuständig. Die Klasse Partikel stellt den Zustand eines einzelnen Partikels dar, dementsprechend die Position, das Gewicht sowie die Geschwindigkeit und die Richtung, in der sich das Partikel zuletzt bewegt hat. Abbildung 38 zeigt die wichtigsten Klassen für den Partikel-Filter. Der Partikel-Filter kann von einer beliebigen Activity genutzt werden. Die Activity registriert sich selbst als Callback beim Partikel-Filter, um über neue Positionen informiert zu werden. Die Activity erstellt einen WLAN oder Bluetooth Service und übergibt den Partikel-Filter direkt als Callback, so dass neue Sensorwerte direkt an den Partikel-Filter übergeben werden (siehe Abbildung 39). Der Partikel-Filter ist nicht bekannt, woher die Daten kommen. So sind die Komponenten voneinander unabhängig und können leicht durch andere Komponenten erweitert werden.

Visualisierung der Partikel

Für die Dauer der Entwicklung war es nötig die Partikel zu visualisieren um die korrekte Funktionsweise des Partikel-Filters zu testen. Dazu wurde eine Klasse PartikelView angelegt, welche die Position der Partikel als Punkt auf einer Karte darstellt. Die Klasse PartikelView erbt von der Klasse View. Eine View ist die allgemeinste Komponente,

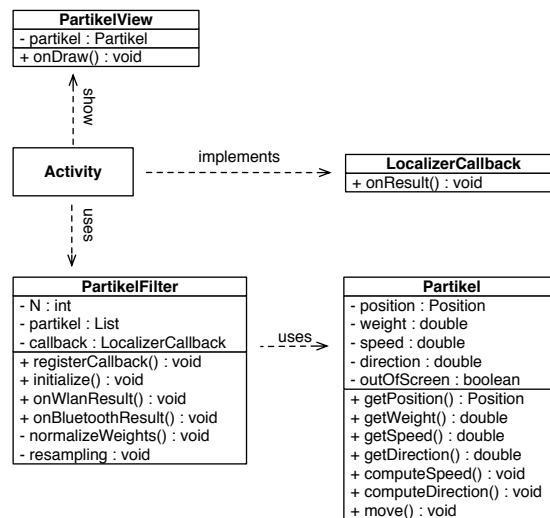


Abbildung 38: Klassendiagramme des Partikel-Filters.

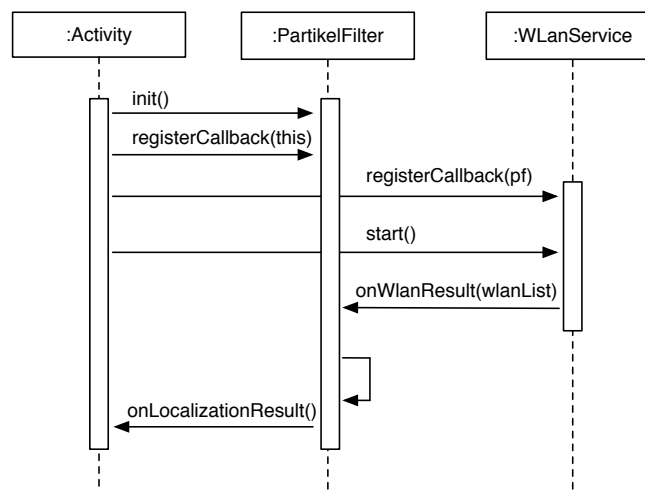


Abbildung 39: Dieses Sequenzdiagramm zeigt die Verwendung eines Partikel-Filters und eines Services.

die etwas darstellen kann. Jede Komponente, die etwas darstellt wie z.B. ein Button, ein Bild oder Text, ist von der Klasse View abgeleitet. Eine View bietet eine Methode `onDraw()` an. In dieser Methode können beliebige Dinge auf eine Leinwand (engl. canvas) gezeichnet werden. Die PartikelView zeichnet an der Position, wo sich ein Partikel befindet, einen Punkt. Abbildung 40 zeigt die Initialisierung der Partikel über den gesamten Zustandsraum. Nach einer Messung und einer Bewegung ordnen sich die Partikel im Zu-

standsraum neu an. Abbildung 41 zeigt, dass die meisten Partikel um einen bestimmten Punkt in einer Wolke konzentriert sind. Aus dem Zustand aller Partikel kann nun eine konkrete Position abgeleitet werden.

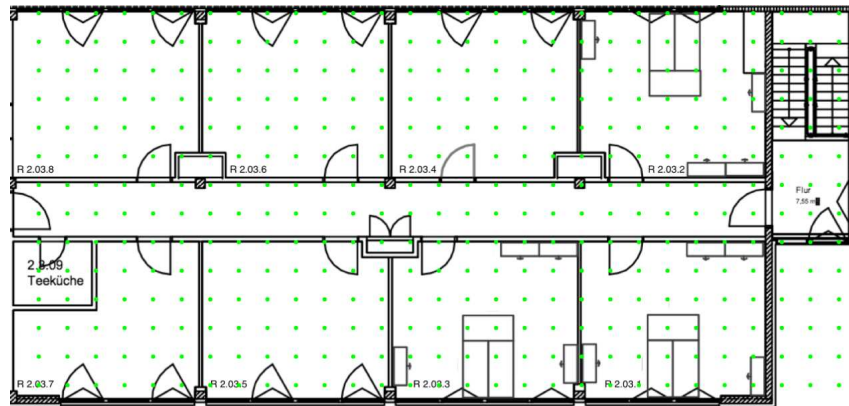


Abbildung 40: Screenshot: Gleichverteilung aller Partikel (grüne Punkte) während der Initialisierungsphase.

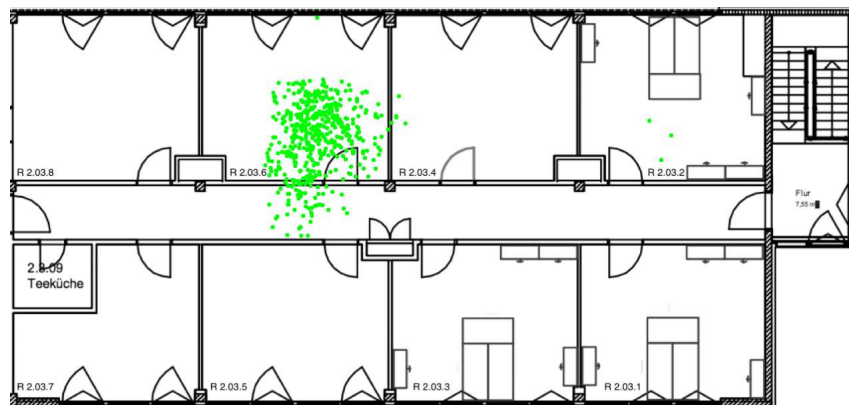


Abbildung 41: Screenshot: Partikel Wolke entspricht dem Ort der wahrscheinlichsten Position.

7.3 Visuelle Erkennung

7.3.1 Digitalisierung der Objekte

Nun kann die Funktionalität der Aufnahme eines Bildes in der Anwendung programmiert werden. Da es jedoch bereits ausreichend Kamera Anwendungen gibt, wäre es besser auf bestehende Anwendungen zurückzugreifen. Hier bietet Android die Möglichkeit externe Anwendungen aus der eigenen Anwendung zu verwenden. Da auf jedem Android eine Kamera Anwendung vorinstalliert ist, wäre es sinnvoll diese zu nutzen. Mit Hilfe des Konzeptes der *Intents* kann mit anderen Anwendungen kommuniziert und Daten ausgetauscht werden. Außerdem ist es damit möglich andere Anwendungen zu starten²⁰. Gibt es mehrere Anwendungen, die auf die Anfrage reagieren, kann der Benutzer sich die gewünschte Anwendung aus einer Liste auswählen. Der Code-Auszug 7.1 zeigt das starten einer externen Kamera Anwendung.

Code 7.1 Starten einer Kamera App.

```
protected void startCameraActivity() {  
    // Erzeuge eine Datei wo das Bild gespeichert werden kann.  
    File file = new File(mPath);  
    // Uri auf Datei  
5    Uri outputFileUri = Uri.fromFile( file );  
  
    // Erstelle Intent mit einer Action, die eine Anwendung auswählt,  
    // welche Bilder aufnehmen kann.  
    Intent intent = new Intent(android.provider.MediaStore.  
        ACTION_IMAGE_CAPTURE );  
10    // Übergebe Parameter an Intent.  
    intent.putExtra( MediaStore.EXTRA_OUTPUT, outputFileUri );  
  
    // Starte Activity  
    startActivityForResult(intent, mrequestCode);  
15 }
```

Mit Hilfe der Kamera Anwendung kann nun ein Bild vom Objekt aufgenommen werden. Nach Beendigung der Kamera Anwendung ist das Bild unter der definierten URI gespeichert. Anschließend können auf diesem Bild die Merkmale berechnet werden.

²⁰Natürlich können mit Hilfe der Intents nur externe Anwendungen genutzt werden, wenn diese das explizit unterstützen und erlauben.

Berechnung der Merkmale

Zur Berechnung der Merkmale eines Bildes werden die Algorithmen aus der OpenCV Bibliothek verwendet. Das Bild wird aus der Datei als Graubild gelesen und in der Klasse `cv::Mat` abgelegt (Zeile 3). Die Klasse `cv::Mat` stellt ein zweidimensionales numerisches Array dar, kann also sowohl eine Matrix, ein Bild oder andere zweidimensionale Daten repräsentieren [29]. Anschließend werden die Schlüsselpunkte berechnet. OpenCV bietet verschiedene Schlüsselpunkt Detektoren. Neben SIFT gibt es noch SURF (Speeded Up Robust Features [BTG06]) und FAST Schlüsselpunkt Detektoren. Der Code-Auszug 7.2 zeigt die Berechnung der Schlüsselpunkte und Deskriptoren in C++. Abbildung 42 zeigt die berechneten Schlüsselpunkte aus einem Bild. Die Schlüsselpunkte und Deskriptoren müssen noch gespeichert werden.

Code 7.2 Berechnete der Schlüsselpunkte und Deskriptoren.

```
void computeKeypointsAndDescriptors(cv::Mat &image, std::vector<cv::
    KeyPoint> keypoints, cv::Mat &descriptors;) {
2   cv::Ptr<cv::FeatureDetector> detector;
    detector = cv::FeatureDetector::create("SIFT");
4   detector->detect(image, keypoints); // Ermittle Schlüsselpunkte

6   cv::Ptr<cv::DescriptorExtractor> extractor;
    extractor = cv::DescriptorExtractor::create("SIFT");
8   // Extrahiere Deskriptoren
    extractor->compute(image, keypoints, descriptors);
10 }
```

Die Schlüsselpunkte und Deskriptoren müssen noch gespeichert werden.

Speichern und laden der Schlüsselpunkte und Deskriptoren

Damit die Objekte bzw. die Bilder später verglichen werden können, müssen die berechneten Schlüsselpunkte und Deskriptoren in einer Datenbank gespeichert werden. Da es in OpenCV mit Java zur Zeit nicht möglich ist die Schlüsselpunkte in der Klasse `cv::Mat` zu serialisieren, muss ein Umweg über C++ gemacht werden. In C++ können die Deskriptoren aus der Klasse `cv::Mat` mit einem `cv::FileStorage` in eine XML oder YAML Datei geschrieben werden wie im Code-Auszug 7.3 zu sehen ist. Mit Hilfe des JNI Frameworks wird die C++ Funktion `save()` aufgerufen, welche die Deskriptoren in eine XML Datei schreibt. Der Funktion wird eine Referenz auf das `cv::Mat` Objekt sowie dem Pfad, wo die Datei gespeichert werden soll, übergeben. Die XML-Datei bekommt selbst

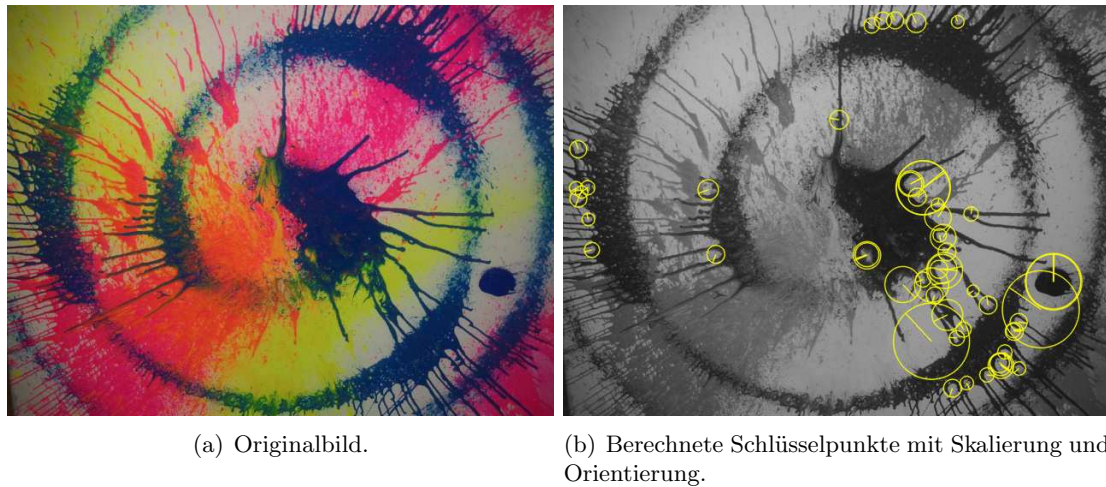


Abbildung 42: Berechnung der Schlüsselpunkte auf einem Bild [Bun12].

einen Zeitstempel als Dateiname. Um die SQLite Datenbank nicht unnötig aufzublähen werden nur die Pfade auf die XML-Dateien mit den Deskriptoren in die Datenbank geschrieben.

Code 7.3 Speichern und laden der Deskriptoren in XML.

```

void save(const char* file, cv::Mat descriptors) {
2   cv::FileStorage fs(file, cv::FileStorage::WRITE);
   fs << "descriptors" << descriptors;
4   fs.release();
}

6

void load(const char* file, cv::Mat &descriptors) {
8   cv::FileStorage fs(file, cv::FileStorage::READ);
   fs["descriptors"] >> descriptors;
10  fs.release();
}

```

7.3.2 Objekterkennung: Nächste Nachbar Suche

Damit ein Objekt visuell erkannt werden kann, soll ein aktuelles Bild mit gespeicherten Bildern in der Datenbank verglichen werden. Da eine Erkennung aus einem Live-Video Stream aufwendiger ist, soll im ersten Schritt ein Foto vom Objekt gemacht werden, das erkannt werden soll. Für die Erkennung werden die Deskriptoren eines Bildes mit den

Deskriptoren eines anderen Bildes verglichen. Es wird eine simple Suche nach dem ähnlichsten Deskriptoren vorgenommen. Diese Verfahren kann als nächste Nachbar Suche oder **Bruteforce** Methode bezeichnet werden. Dabei kann es durchaus dazu kommen, dass ein Deskriptor der nächste Nachbar von mehreren Deskriptoren im anderen Bild ist (Mehrfachzuordnung). Der nächste Nachbar wird im folgenden auch als Match bezeichnet. Ein Match wird in einem Vektor von `cv::DMatch` Objekten abgelegt. Ein `DMatch` Objekt enthält die Abstände und eine Referenz zu allen nächsten Schlüsselpunkten. In Abbildung 43 ist zu erkennen das einige Schlüsselpunkte aus dem linken Bild auf den selben Schlüsselpunkt im rechten Bild passen (grün). Andere Schlüsselpunkte im rechten Bild konnte kein Schlüsselpunkt aus dem linken Bild zugeordnet werden (rot).

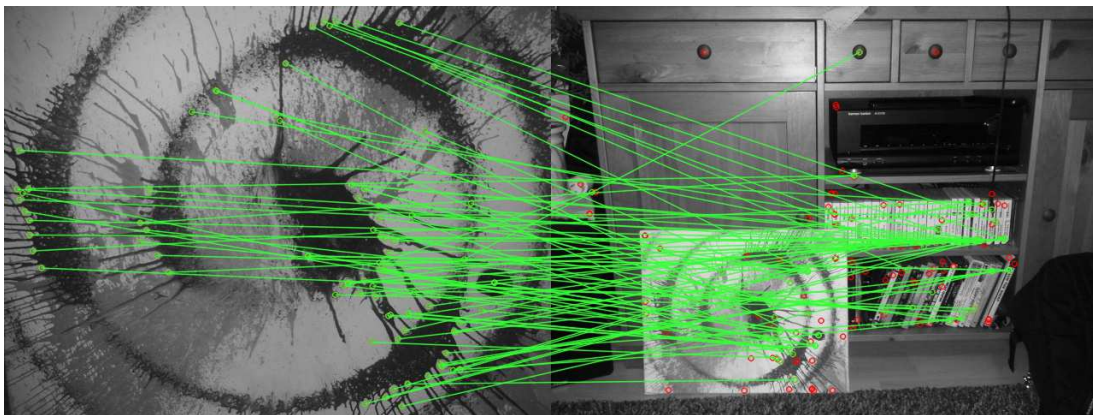


Abbildung 43: Nächste Nachbar Suche der Schlüsselpunkte im Original- und Referenzbild. Anzahl der berechneten Schlüsselpunkte links: 89 und rechts: 157 Grün sind die gefundenen Matches. [Bun12].

Stabile Matches

Wie in Abbildung 43 zu erkennen wurden viele Schlüsselpunkte aus dem Originalbild nicht korrekt den Schlüsselpunkten im Referenzbild zugeordnet. Das liegt zum einen daran, dass im Referenzbild auf dem entsprechenden Objekt nicht so viele Schlüsselpunkte gefunden wurden wie im Originalbild. Um die Qualität des Matchings zu verbessern, soll daher nach stabilen Matches gesucht werden, wie in Abschnitt 5.2 auf Seite 61 erläutert wurde.

Stabile Matches sollen möglichst nur auf Schlüsselpunkten zwischen zwei Bildern ermittelt werden, die das selbe Objekt zeigen. Das Ergebnis der Suche nach einem Objekt

in einem Testbild wäre das Referenzbild mit den meisten gefundenen stabilen Matches. Bevor stabile Matches ermittelt werden können muss eine nächste Nachbar Suche durchgeführt werden. Zu jedem Deskriptor im in einem Bild werden die, zwei nächsten Deskriptoren (Nachbarn) im anderen Bild gesucht. Theoretisch können auch n nächste Nachbarn gesucht werden, jedoch wird sich in diesem Fall auf die zwei nächsten Nachbarn beschränkt.

Nun liegen zwei Listen mit Matches der Deskriptoren vor (vom linken im rechten Bild, und vom rechten im linken Bild). Der Verhältnistest (siehe Algorithmus 3) prüft nun die Ähnlichkeit der zwei gefundenen nächsten Deskriptoren eines Matches. In Zeile 4 wird geprüft, ob das Verhältnis des kürzesten Matches und des zweit-kürzesten Matches unter dem Schwellwert liegt. Wenn dem so ist, wird der Match in die Liste der stabilen Match-Kandidaten aufgenommen. Tabelle 4 zeigt die Anzahl der Matches die auf den

Algorithm 3 Verhältnis Test

```
1: stabileMatchKandidaten =  $\emptyset$ 
2: for all Matches in Liste do
3:   if 2 NN existieren then
4:     if (match[0].distance / match[1].distance < Schwellwert) then
5:       stabileMatchKandidaten = stabileMatchKandidaten + match
6:     end if
7:   end if
8: end for
```

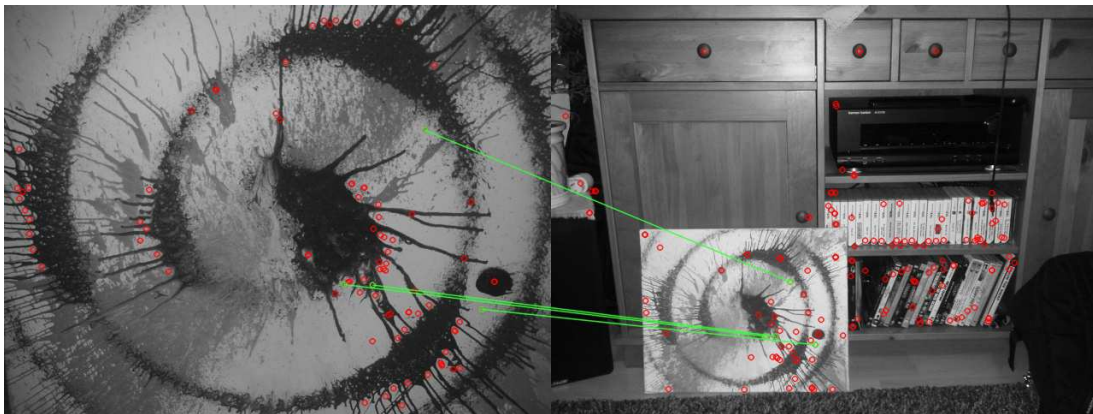
Schlüsselpunkten berechnet worden sind. Im linken Bild wurden 89 und im rechten Bild 157 Schlüsselpunkte berechnet. Nach der nächsten Nachbar Suche wurden 89 Matches im rechten Bild gefunden und 157 Matches im linkes Bild. Also zu 89 Schlüsselpunkten im linken Bild wurden 89 passende Schlüsselpunkte im rechten Bild gefunden und für 157 Schlüsselpunkte wurden 157 passende Schlüsselpunkte im linken Bild gefunden. Nach dem Verhältnis Test sind im linken noch 14 und im rechten Bild 15 Matches übrig. Anschließend folgt der Symmetrietest (siehe Algorithmus 4), bei dem geprüft wird ob in zwei Listen mit Matches, zwei Deskriptoren aus verschiedenen Bildern gegenseitig als beste Matches identifiziert worden sind. Ein Match zeigt dabei auf den Schlüsselpunkt aus dem Trainingsbild (trainIdx) und dem passenden Schlüsselpunkt im Suchbild (queryIdx). Wenn dem so ist, dann wird er Match als symmetrischer Match in der Liste gespeichert. Tabelle 4 und Abbildung 44 zeigen, dass nach dem Verhältnis- und dem Symmetrietest noch vier stabile Matches in beiden Bildern übrig geblieben sind.

Algorithm 4 Symmetrie Test: Liste matches1, Liste matches2, k=2

```

1: symMatches =  $\emptyset$ 
2: for i = 0; i < länge matches1 do
3:   for n = 0; n < länge matches2 do
4:     if (matches1[i].queryIdx == matches2[n].trainIdx) &&
        (matches1[i].trainIdx == matches2[n].queryIdx) then
5:       symMatches = symMatches + matches1[i]
6:     end if
7:   end for
8: end for

```


 Abbildung 44: Stabile Matches, *Schwellwert* = 0.75 [Bun12].

	Links	Rechts
Schlüsselpunkte	89	157
Matches	157	89
nach Verhältnis Test	15	14
nach Symmetrie Test	4	4

Tabelle 4: Aussortierung schlechter Matches.

Match-Score

Der Match-Score soll eine Aussage über die Qualität eines Matches von einem Bild mit mehreren Referenzbildern machen. Nachdem ein Bild mit mehreren Referenzbildern verglichen wurde, wird die Anzahl der stabilen Matches zusammen mit dem jeweiligen Referenzbild in einer Liste gespeichert. Diese Liste wird absteigend nach der Anzahl der Matches sortiert (siehe Algorithmus 5). Das erste Element in der Liste enthält nun die

Algorithm 5 Berechnung der Matches. **Parameter:** bild, listRefBilder

```
1: Liste matchesList
2: for  $i = 0 \rightarrow \text{länge}(\text{listRefBilder})$  do
3:   matchesList  $\leftarrow$  matchesList + berechneMatches(bild, listRefBilder[i])
4: end for
5: return sortiereAbsteigend(matchesList)
```

meisten Matches, das zweite Element die zweit-meisten usw.. Sollte die Liste leer sein, wurde kein passendes Referenzbild gefunden und der Score beträgt null. Enthält die Liste nur ein Element, wurde nur zu einem Bild stabile Matches gefunden. Der Match-Score ist in diesem Fall exakt eins, da es keine anderen stabilen Matches aus anderen Bildern gibt. Ist jedoch mehr als ein Element in der Liste enthalten, werden beginnend beim zweiten Element die stabilen Matches mit denen des ersten Elements ins Verhältnis gesetzt und vom Score abgezogen.

Algorithm 6 Bestimmung des Match-Scores. **Parameter:** sortMatchesList

```
1: if länge(sortMatchesList) = 0 then
2:   return 0
3: end if
4: if länge(sortMatchesList) = 1 then
5:   return 1
6: end if
7: if länge(sortMatchesList) > 1 then
8:   best  $\leftarrow$  sortMatchesList
9:   similarity  $\leftarrow$  1
10:  for  $i = 1 \rightarrow \text{länge}(\text{sortMatchesList})$  do
11:    score  $\leftarrow$  score - (sortMatchesList[i] / best)
12:    if score  $\leq$  0 then
13:      return 0
14:    end if
15:  end for
16:  return score
17: end if
```

Beispiel-Rechnung zur Bestimmung des Match-Scores

Bild	Referenzbild	Matches
Ω	A	294
Ω	B	18
Ω	C	3

$$\begin{aligned} \text{score} &= 1 - (18/294) - (3/294) \\ &= 0,93 \end{aligned}$$

Der Match eines Bildes Ω zum Referenzbild A hat einen Score von 0,93. Legt man eine Grenze für den Score bei 0,5 fest dann, kann gesagt werden, dass der Match zu dem Referenzbild A ein qualitativ guter Match ist. Es kann davon ausgegangen werden, dass das Referenzbild A mit hoher Wahrscheinlichkeit zum Bild Ω gehört.

Bild	Referenzbild	Matches
Ω	A	312
Ω	B	146
Ω	C	87

$$\begin{aligned} \text{score} &= 1 - (146/312) - (87/312) \\ &= 0,25 \end{aligned}$$

In der zweiten Rechnung liegt der Match-Score mit 0,25 unter der Grenze von 0,5. Nun kann nicht mehr davon ausgegangen werden, dass das Referenzbild A zum Bild Ω gehört.

8 Auswertung

In diesem Kapitel soll die visuelle Erkennung und die Lokalisierung bezüglich ihrer Genauigkeit untersucht werden.

8.1 Lokalisierung

In diesem Abschnitt soll untersucht werden, ob und wie genau die entwickelten Lösungsansätze für die Lokalisierung funktionieren. Dazu wird die Genauigkeit des Partikel-Filters bezüglich WLAN und Bluetooth untersucht. Zuerst werden Daten für eine Simulation aufgenommen. Konkret werden an 28 Punkten auf der dritten Etage des TGS Sensorwerte aufgezeichnet. Diese Daten stellen die aktuellen Messwerte dar, die mit den gespeicherten Daten verglichen werden, um eine Position zu berechnen. Zu jedem Datensatz wird die aktuelle Position gespeichert, um daraus ermitteln zu können, wie weit die berechnete Position von der tatsächlichen Position abweicht. Zusammen mit einem Zeitstempel werden die Daten im JSON Format gespeichert. Die Auswertung der Daten erfolgt in sequentieller Reihenfolge geordnet nach den Zeitstempeln. Um aus dem Zustand der Partikel eine konkrete Position zu ermitteln, sollen drei mögliche Methoden getestet werden.

1. **Summe aller Partikel:** Die Summe aller Partikel ergibt die geschätzte Position.
2. **Größtes Partikel-Gewicht:** Das Partikel mit dem größten Gewicht ergibt die wahrscheinlichste Position.
3. **FP mit meisten Partikeln:** Der Fingerprint, welcher die meisten Partikel um sich hat ist die wahrscheinlichste Position.

Außerdem gibt es noch eine vierte Methode (**Raum Mittel**), welche unabhängig vom Partikel-Filter die Ähnlichkeit der Fingerprints in Bezug zur aktuellen Messung berechnet. Aus den ermittelten Ergebnissen jedes Fingerprints wird pro Raum das arithmetische Mittel gebildet. Der Raum mit dem kleinsten Mittelwert ist der wahrscheinlichste Ort, an dem man sich gerade aufhält.

Für die Simulation wurde der Partikel-Filter konfiguriert und mit den Simulationsdaten ausgestattet. Es hat sich gezeigt, dass ein einziger Durchlauf des Partikel-Filters nicht immer zu den selben Ergebnissen führt. Geringe Abweichungen sind in mehreren Iterationen der Simulation zu erkennen. Um dennoch eine konkrete Aussage über die Genauigkeit machen zu können, soll ein Durchlauf der Simulation genau 500 mal wiederholt werden. Aus den daraus ermittelten Ergebnissen kann ein Mittelwert berechnet werden. Abbildung 45 zeigt das Ergebnis der ermittelten Genauigkeiten aus 500 Wiederholungen. Jeder Punkt stellt die mittlere Genauigkeit bei einem Durchlauf dar. Es ist zu erkennen, dass die mittlere Genauigkeit von ca. 5 bis zu 12 Meter streut. Jedoch ist eine dichtere Ansammlung im Bereich von 5 bis 6 Meter zu erkennen. Die mittlere Genauigkeit bezieht sich in diesem Fall auf die die Summe aller Partikel und der daraus abgeleiteten Position.

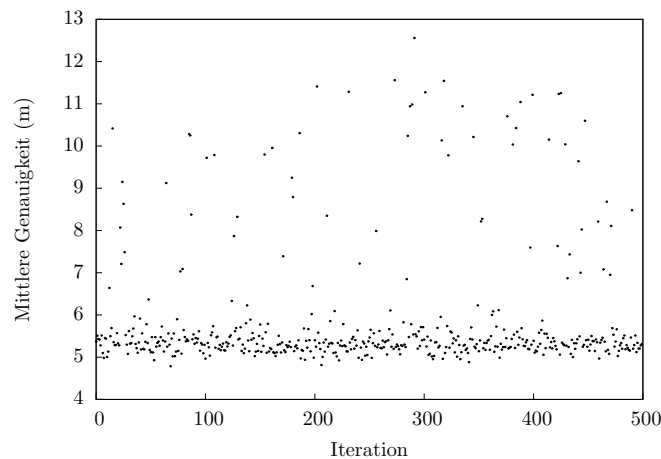


Abbildung 45: Mittlere Genauigkeit (Summe aller Partikel) mit dem Partikel Filter in 500 Iterationen.

WLAN

Tabelle 5 zeigt die Ergebnisse für die Lokalisierung mittels WLAN. Es ist zu erkennen, dass die Summe aller Partikel die höchste mittlere Genauigkeit mit $5,86m$ bei einer Präzision von 84% liefert. Die Präzision bezieht sich dabei auf den Mittelwert. Außerdem ist die Standardabweichung und der maximal gemessene Wert angegeben. Alle anderen Methoden der Positionsbestimmung liefern schlechtere Werte. Eine Untersuchung der

	Mittel (m)	Präzision	Stabw (m)	Max (m)
Summe aller Partikel	5,85	0,84	3,43	13,24
Größtes Partikel-Gewicht	6,57	0,80	3,92	15,91
FP mit meisten Partikel	6,23	0,79	3,51	13,92
Raum Mittel	7,86	0,46	4,46	17,80

Tabelle 5: Ergebnisse der Genauigkeit der Lokalisierung mit WLAN.

Ergebnisse pro Raum hat ergeben, dass die Genauigkeit nicht in jedem Raum gleich ist. Mitunter gab es hier starke Schwankungen wie in Abbildung 46 zu erkennen ist. Die zugehörige Tabelle 11 kann im Anhang auf Seite V eingesehen werden. Die Positionsbestimmung wurde auch hier mit der Summe aller Partikel vorgenommen. Jedoch ist auch zu erkennen, dass eine Lokalisierung in einzelnen Räumen durchaus gut funktioniert.

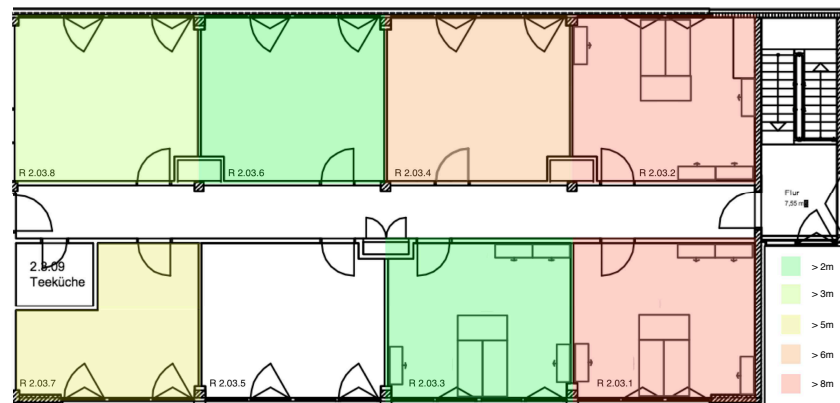


Abbildung 46: Grafische Darstellung der mittleren Genauigkeit in den einzelnen Räumen. Die Farben geben die mittlere Genauigkeit in diesem Raum an.

Bluetooth

Auch für Bluetooth ergibt die Summe aller Partikel die beste Genauigkeit mit 6,56m bei einer Präzision von 58%. Die Werte sind alle etwas schlechter als bei WLAN. Stellt man nun direkt die Ergebnisse für WLAN und Bluetooth für die Summe aller Partikel gegenüber, wird deutlich, dass WLAN bis zu einer Präzision von 80% eine bessere Genauigkeit liefert als Bluetooth. Jedoch ist Bluetooth ab einer Präzision von 90% bzgl. der Genauigkeit gegenüber WLAN im Vorteil.

	Mittel (m)	Präzision	Stabw (m)	Max (m)
Summe aller Partikel	6,56	0,58	3,04	12,80
Größtes Partikel-Gewicht	6,64	0,61	3,10	13,03
FP mit meisten Partikel	6,93	0,58	3,41	13,09
Raum Mittel	7,98	0,54	4,99	18,15

Tabelle 6: Ergebnisse der Genauigkeit der Lokalisierung mit Bluetooth.

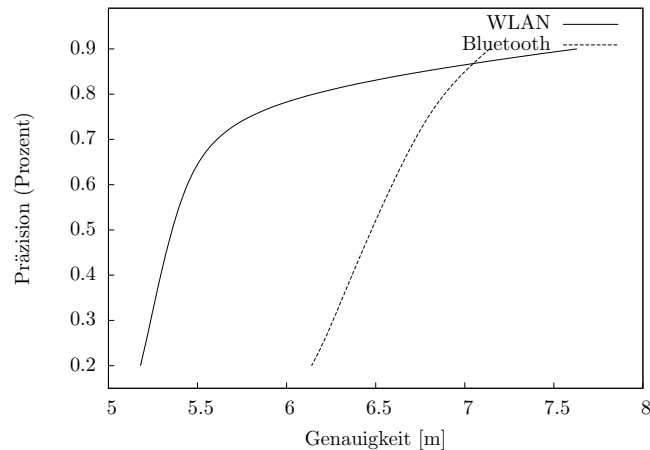


Abbildung 47: Gegenüberstellung von WLAN und Bluetooth bzgl. der Genauigkeit und der Präzision.

Fazit

Vergleicht man die ermittelte Genauigkeit für WLAN und Bluetooth mit anderen Lösungen, wird deutlich, dass die erzielten Ergebnisse nicht mit anderen Lösungen mithalten können. Die Ergebnisse bezogen sich jedoch auf relativ kleine Räume von ca. 25m^2 . Sollten für größere Räume wie sie z.B. im Museum vorkommen die Ergebnisse ähnlich sein, kann durchaus mit einer mittleren Genauigkeit von $5,85\text{m}$ für WLAN eine Lokalisierung umgesetzt werden. Diese hätte den Vorteil, dass sie ohne eine aufwendige Infrastruktur auskommt und von den meisten Benutzern mit einem Smartphone verwendet werden kann. Andere Lösungen wie Infrarot, Ultraschall oder RFID sind zwar genauer, benötigen aber zusätzliche Hardware und können daher nicht ohne Weiteres für eine breite Nutzerschicht angeboten werden.

Energieverbrauch

Da verschiedene Sensoren für die Lokalisierung genutzt werden, ist es sinnvoll zu untersuchen, wie viel Energie die Nutzung der Sensoren verbraucht. Das ist wichtig, da auf mobilen Geräten nur eine begrenzte Energieversorgung zur Verfügung steht. Im Folgenden Versuch soll der Akkuverbrauch für die einzelnen Sensoren untersucht werden. Um den Akkuverbrauch möglichst nur für die Sensoren zu bestimmen, wurden alle anderen Hintergrunddienste deaktiviert und das Display blieb während des gesamten Versuchs ausgeschaltet. Die Messung des Energieverbrauchs wird für genau einen Sensoren ausgeführt. Ein Sensor soll ununterbrochen aktiv sein. Zu Beginn und nach jeder weiteren Stunde wird der aktuelle Stand des Akkus ausgelesen und gespeichert. Abbildung 48 zeigt den gemessenen Energieverbrauch. Es ist zu erkennen, dass WLAN nach sieben Stunden Laufzeit nur 10% der gesamten Energie verbraucht hat. Bluetooth hat in der selben Zeit mehr als 20% der Energie verbraucht. Der Luftdruck-, Beschleunigungs- und Kompass-Sensor haben einzeln und auch zusammen keine Energie verbraucht. Dass der Energieverbrauch dieser Sensoren geringer ist als bei WLAN und Bluetooth, ist damit zu erklären, da WLAN und Bluetooth Energie für ein elektromagnetisches Übertragungsmedium bereitstellen müssen, was zu einem erhöhten Energieverbrauch führt. Die Energiemessung hat jedoch gezeigt, dass es durchaus sinnvoll ist, die WLAN und Bluetooth Sensoren nur bei Bedarf zu verwenden. Denn es muss davon ausgegangen werden, dass verschiedene Geräte über unterschiedliche Akkuleistungen verfügen. Außerdem kann nicht von einer vollen Akkuleistung ausgegangen werden. Daher ist das Energiesparen auf mobilen Geräten sehr wichtig.

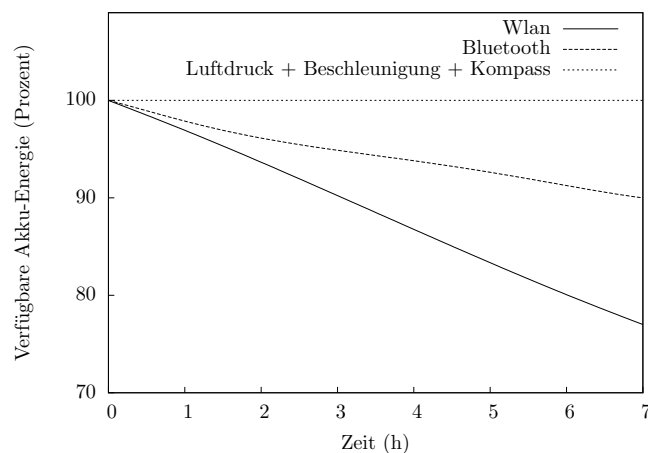


Abbildung 48: Zeigt den Energieverbrauch der verschiedenen Sensoren.

8.2 Visuelle Erkennung

In diesem Abschnitt soll ausgewertet werden, wie gut die visuelle Erkennung funktioniert. Wie schon in Abschnitt 5.1 beschrieben, lassen sich nicht alle Objekte gleich gut erkennen. Dies liegt hauptsächlich an der Form oder der Farbe der Objekte. Im Folgenden werden Beispiele verschiedener Objekte gezeigt. Abbildung 49 zeigt eine Kugel und einen Würfel. Beide zeigen einen monotonen Farbverlauf. Auf diesen Objekten können kaum Merkmale berechnet werden. Daher sind einfarbige Objekte unabhängig von der Aufnahmeperspektive nicht für eine Erkennung geeignet.

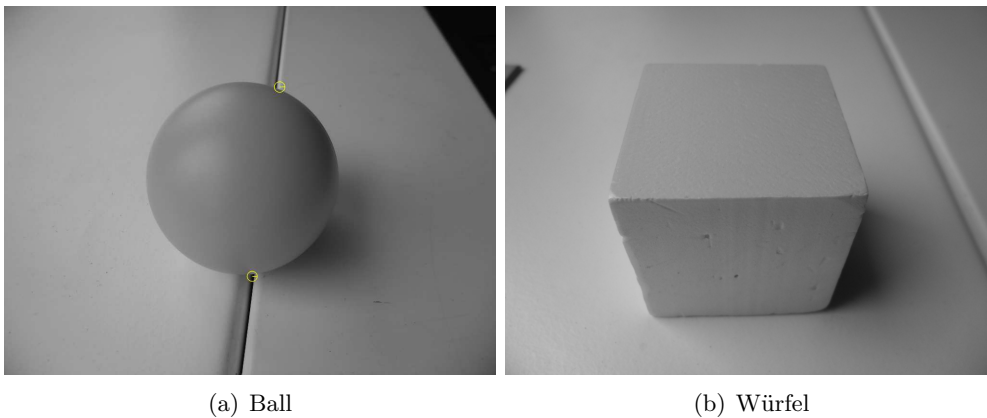


Abbildung 49: Einfache Objekte mit monotonen Farbverlauf.



Abbildung 50: Durchsichtige Objekte.

Abbildung 50 zeigt ein Gyroskop und ein Glas. Auf dem Gyroskop können zwar Merkmale berechnet werden, aber viele davon liegen nicht auf dem Objekt selber. Auf dem Glas wurden keine Merkmale gefunden. Objekte, die durchsichtig sind, eignen sich ebenfalls schlecht für eine Objekterkennung. Abbildung 51 zeigt zwei Tastaturen, eine mit schwar-

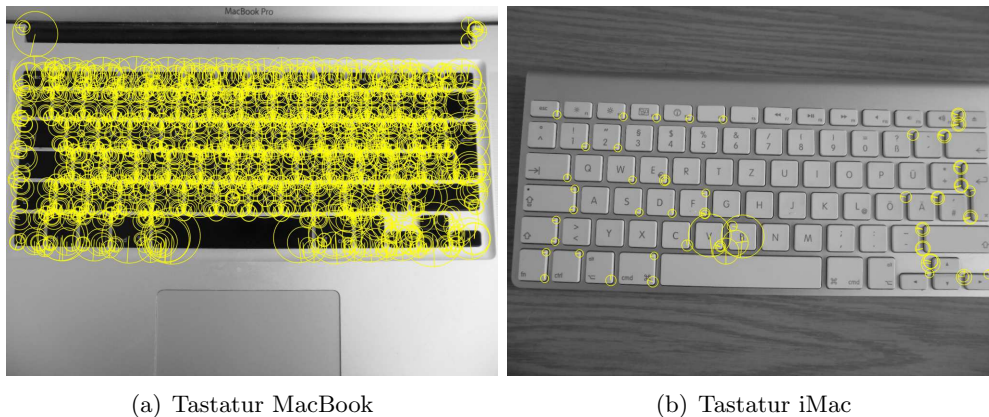


Abbildung 51: Gleiche Objekttypen mit unterschiedlich vielen Schlüsselpunkten.

zen und eine mit weißen Tasten. Auf der Tastatur mit den schwarzen Tasten wurden sehr viele Merkmale gefunden, weil schwarz sich sehr gut vom silbernen Hintergrund abhebt. Auf der weißen Tastatur konnten nicht so viele Merkmale gefunden werden, da sich weiß nicht sehr stark vom silbernen Hintergrund unterscheidet. Hier wird deutlich, wie wichtig es ist Objekte mit starken Kontrasten auszuwählen. Abbildung 52 zeigt ein Spiel

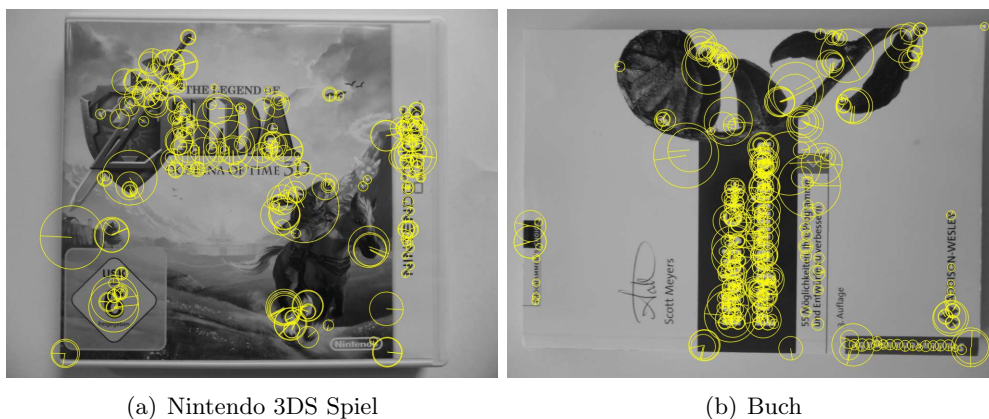


Abbildung 52: Objekte mit vielen Farben und vielen starken Kontrasten.

und ein Buch, beide mit vielen Farben und kontrastreichen Regionen. Daher wurden auf

beiden Objekten auch viele Merkmale erkannt.

8.2.1 Vergleich der Matching Verfahren: Bruteforce vs. Stabile Matches

Nun soll ausgewertet werden, welches der implementierten Verfahren, Bruteforce oder stabile Matches für die visuelle Erkennung am besten geeignet ist. Für das Bruteforce Verfahren soll geprüft werden, ob mit der Summe der Abstände aller Matches eine eindeutige Aussage zur Zugehörigkeit eines Testbildes zu einem Referenzbild gemacht werden kann. Bei der Methode der stabilen Matches soll geprüft werden, ob allein durch die Anzahl guter Matches die gleiche Aussage gemacht werden kann. Zur Verfügung

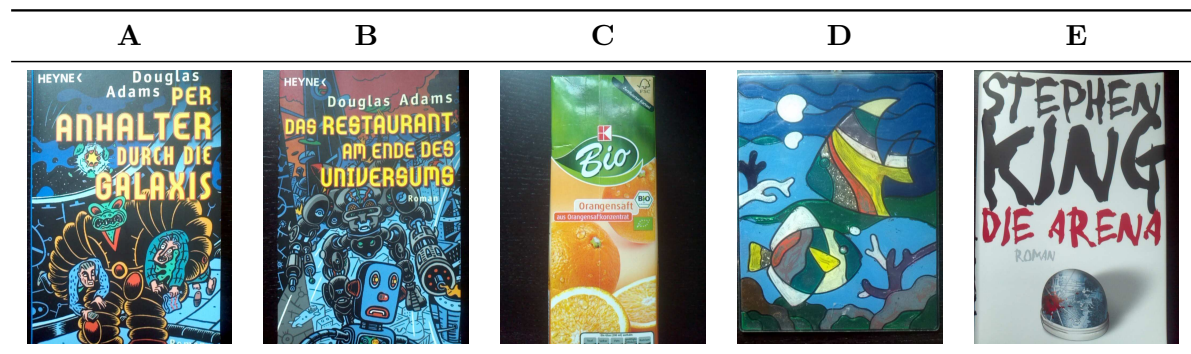


Tabelle 7: Referenzbilder.

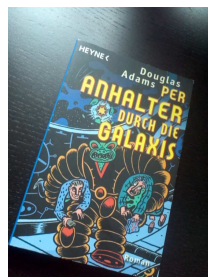


Abbildung 53: Testbild.

steht ein Testbild Ω (siehe Abbildung 53) sowie fünf Referenzbilder (A bis E, siehe Tabelle 7). Das Testbild und das Referenzbild A stellen beide das selbe Objekt dar. Es handelt sich dabei um ein Buch: *Per Anhalter durch die Galaxis* - Douglas Adams. Daher müsste es zwischen den beiden Bildern die meisten Übereinstimmung geben. Das Referenzbild B zeigt ebenfalls ein Buch: *Das Restaurant am Ende des Universums* - Douglas Adams. Beide Objekte haben gewisse Ähnlichkeiten. Da auf bei beiden Büchern

die Schriftart die selbe ist, könnte es bezüglich der Buchstaben zu Übereinstimmungen kommen. So ist der Verlag sowie der Name des Schriftstellers auf beiden Objekten der selbe. Zu den anderen Testbildern C bis E sollte es weniger Übereinstimmungen geben. Tabelle 8 zeigt das Ergebnis des Tests. Für jedes Bild ist die Anzahl der gefundenen Schlüsselpunkte sowie das Ergebnis und die Zeit für die beiden Verfahren dargestellt. Da die Berechnung der Schlüsselpunkte und der Deskriptoren von den beiden Verfahren unabhängig ist und somit auch die selbe Zeit dauern würde, wurde nur die Zeit für das Matching gemessen. Es ist zu erkennen, dass die Brute-force Methode für die Bilder B und A die kleinsten Abstände der Matches ermittelt hat. Das beste Ergebnis wurde für das Referenzbild B ermittelt. Die Ergebnisse für Bild B und A liegen sehr dicht beieinander. Hier wird deutlich das Brute-force bei ähnlichen Bildern keinen guten Ergebnisse liefert. Angenommen das Referenzbild B wäre nicht in den Referenzbildern dabei, dann wäre das Referenzbild A mit deutlichen Abstand zu den anderen Referenzbildern das beste Referenzbild. Das Matching mit Brute-force hat für fünf Referenzbildern insgesamt 1,17 Sekunden gedauert. Betrachtet man dagegen die Anzahl der stabilen Matches, dann fällt auf, dass zum Referenzbild A die meisten Matches gefunden wurden. Weit abgeschlagen auf dem zweiten Platz ist das Referenzbild B. Jedoch gab es hier noch einige wenige stabile Matches, was sich durch die Ähnlichkeit der beiden Bilder erklären lässt. Bei den restlichen Referenzbildern wurden vernachlässigbar wenig bis keine guten Matches gefunden. In der Qualität der Ergebnisse liegen die stabilen Matches deutlich vor der Brute-force Methode. Da die Methode der stabilen Matches zusätzliche Tests benötigt, dauert diese etwas mehr als die doppelte Zeit gegenüber der Brute-force Methode. Insgesamt benötigt diese für alle fünf Referenzbilder 2,52 Sekunden.

Bei der Verwendung von Brute-force sollte darauf geachtet werden, dass ähnliche Objekte nicht in den Referenzbildern vorkommen, da es hier schnell zu falschen Ergebnissen kommen kann. Ansonsten ist das Brute-force Verfahren bezüglich der Zeit klar im Vorteil. Deutlich bessere Ergebnisse lieferte jedoch die Methode der stabilen Matches.

Da die Zeit bei der Erkennung eine wichtige Rolle spielt, soll untersucht werden, ob durch weniger Schlüsselpunkte ähnlich gute Ergebnisse erzielt werden können.

Wenige Schlüsselpunkte

Auch bei weniger Schlüsselpunkten sind die Ergebnisse ähnlich wie zuvor, wie aus Tabelle 9 zu entnehmen ist. Die Brute-force Methode hat wieder das Referenzbild B als ähnlichstes

Referenzbild	Schlüsselpunkte	Bruteforce		Stabile Matches	
		\sum Abstände	Zeit (s)	Matches	Zeit (s)
A	3003	569	0.44	665	0.97
B	3436	559	0.51	28	1.07
C	353	687	0.05	1	0.11
D	402	677	0.06	2	0.13
E	759	676	0.11	0	0.24

Tabelle 8: Ergebnisse des Matching von Bruteforce und der stabilen Matches.

Bild zum Testbild ermittelt. Auch die Methode der stabilen Matches konnte wie zuvor das Referenzbild A als ähnlichstes Bild bestimmen. Jedoch sind die Verhältnisse²¹ mit 1/6 nun weniger eindeutig als noch im vorherigen Test mit 28/665. Die gesamte Zeit, die für das Matching benötigt wird, ergibt für Bruteforce 0,00131 und für die Methode der stabilen Matches 0,00298 Sekunden. Im Verhältnis zu Bruteforce benötigen die stabilen Matches immer noch etwas mehr als die doppelte Zeit. Jedoch macht es durchaus Sinn mit weniger Schlüsselpunkten zu arbeiten um dadurch einen Zeitvorteil zu bekommen.

Referenzbild	Schlüsselpunkte	Bruteforce		Stabile Matches	
		\sum Abstände	Zeit (s)	Matches	Zeit (s)
A	129	14.8	0.00051	6	0.00115
B	168	13.3	0.00055	1	0.00135
C	1	33.1	0.00006	0	0.00008
D	16	16.3	0.00008	1	0.00016
E	10	16.7	0.00011	0	0.00024

Tabelle 9: Ergebnisse des Matching mit BruteForce und stabilen Matches mit reduzierten Schlüsselpunkten.

8.2.2 Qualität der Objekterkennung

Um die Qualität der Objekterkennung zu testen wurden verschiedene Bilder von Objekten aufgenommen. Diese Bilder dienen als Referenzbilder zu bestimmten Objekten. Ein Objekt kann mehrere Referenzbilder haben, um damit eine Erkennung aus mehreren Perspektiven zu ermöglichen. Diese Referenzbilder stellen die Basis für die Objekterkennung dar. Prinzipiell kann nicht ausgeschlossen werden, dass ein falsches Referenzbild

²¹Anzahl stabiler Matches vom zweit-besten Bild im Verhältnis zur Anzahl stabiler Matches vom besten Bild.

dem Testbild zugeordnet wird. Jedoch ist die Wahrscheinlichkeit für falsche Zuordnungen höher, wenn zwei Referenzbilder von verschiedenen Objekten eine hohe Ähnlichkeit ausweisen. Daher soll geprüft werden, ob jedes Referenzbild die größte Ähnlichkeit nur zu sich selbst hat. Mit Hilfe des Match-Scores kann geprüft werden, wie gut der beste Match wirklich ist. Referenzbilder werden aussortiert, wenn sie falschen Referenzbildern zugeordnet werden oder wenn sie dem richtigen Referenzbild zugeordnet werden aber einen niedrigen Match-Score haben. Der Match-Score wird auf 0,75 festgelegt. Referenzbilder mit einem Match-Score $\leq 0,75$ werden aussortiert.

Tabelle 12 (Anhang, Seite V) zeigt, dass nahezu alle Referenzbilder auf sich selber passen. Nachdem nun alle Referenzbilder geprüft wurden und qualitativ schlechte Referenzbilder aussortiert worden sind, kann nun die Qualität der Erkennung mit verschiedenen Testbildern erfolgen. Der erste Versuch wird für 15 Objekte mit 25 Referenzbildern und 40 Testbildern durchgeführt. Dabei wird geprüft, ob ein Testbild auf das richtige Referenzbild passt. Die Ergebnisse werden in einer Kontingenztafel eingetragen. Tabelle 10 zeigt ob ein Match eines Testbildes richtig oder falsch ist. Es ist zu erkennen, dass von 40 Test-

	Korrekt	Falsch
Score $\geq 0,75$	21	1
Score $< 0,75$	11	7

Tabelle 10: Qualität der Objekterkennung Versuch Eins.

bildern, 21 Bilder mit einem Match-Score $\geq 0,75$ und 11 Bilder mit einem Match-Score $< 0,75$ korrekt zugeordnet worden sind. 8 Acht Bilder wurden insgesamt falschen Referenzbildern zugeordnet, eines sogar mit einem Match-Score über dem Schwellwert. Das ergibt eine Trefferquote von 75%. In diesem Durchgang wurden pro Bild durchschnittlich 306 Schlüsselpunkte berechnet. Die Frage ist nun ob die Qualität der Erkennung durch eine Erhöhung bzw. Verminderung der Schlüsselpunkte beeinflusst wird.

Die Anzahl der Schlüsselpunkte lässt sich nicht direkt einstellen. Durch Beeinflussung von Parametern kann jedoch bestimmt werden, ob mehr oder weniger Schlüsselpunkte gefunden werden sollen. Wie viele Schlüsselpunkte gefunden werden, hängt dann von dem Bild und den eingestellten Parametern ab. Daher soll nun die Qualität der Objekterkennung für verschiedene Parameter geprüft werden. Tabelle 13 (Anhang, Seite VI) zeigt, dass bei durchschnittlich 157 Schlüsselpunkten pro Bild die Erkennungsrate bei 60% liegt. Ab 306 Schlüsselpunkte liegt die Erkennungsrate schon bei 80%. Es konnte eine maximale Erkennungsrate von 95% erzielt werden, welche sich durch mehr als 1026

Schlüsselpunkte nicht verbessern ließ. Es kann festgehalten werden, je mehr Schlüsselpunkte vorhanden sind, desto höher ist die Wahrscheinlichkeit das stabile Matches in zwei Bildern gefunden werden. Jedoch steigt damit die Berechnungszeit für einen Vergleich zweier Bilder. Für eine Erkennungsrate von mind. 90% benötigt ein Match zwischen zwei Bildern durchschnittlich 0,062 Sekunden, bei 590 Schlüsselpunkten pro Bild. Bei 1566 Schlüsselpunkten pro Bild benötigt ein Match durchschnittlich 0,42 Sekunden, also bereits sechs mal soviel bei nur drei mal mehr Schlüsselpunkten. Die Wahl der Parameter für die Anzahl der Schlüsselpunkte muss also sorgsam gewählt werden, um zwischen einer guten Erkennungsrate und einer möglichst geringen Berechnungszeit abzuwägen. Die Dauer für die Berechnung der Schlüsselpunkte betrug pro Bild durchschnittlich 0,48 Sekunden und für die Deskriptoren durchschnittlich 0,2 Sekunden. Da die Schlüsselpunkte und die Deskriptoren für die Referenzbilder vorab berechnet und gespeichert werden, müssen diese nur noch auf dem jeweiligen Testbild berechnet werden. Für eine Erkennungsrate von 90% würde sich somit eine Berechnungszeit von: $0,48s + 0,2s + n \cdot 0,062s$ ergeben, wobei $n \in \mathbb{N}$ für die Anzahl der Referenzbilder steht. Die Frage ist, ob die Erkennungsrate noch verbessert werden kann, sodass bei kürzerer Berechnungszeit bessere Ergebnisse erzielt werden.

Analyse mit weniger Referenzbildern

Im zweiten Versuch wird die Anzahl der Referenzobjekte von 15 auf sechs reduziert. Für die sechs Referenzobjekte stehen 13 Referenzbilder zur Verfügung, welche mit 21 Testbildern verglichen werden sollen. Tabelle 14 (Anhang, Seite VII) stellt die Ergebnisse ausführlich dar. Abbildung 54 zeigt die Ergebnisse beider Versuche. Es ist zu erkennen, dass schon ab 200 Schlüsselpunkte pro Bild eine Erkennungsrate von 80% erreicht wird. Ab 399 Schlüsselpunkte werden die Testbilder zu über 90% korrekt erkannt. Für eine Erkennungsrate von 90% beträgt die Berechnungszeit einmalig 0,48s für die Berechnung der Schlüsselpunkte und 0,1s für die Berechnung der Deskriptoren. Die Dauer für einen Match zweier Bilder beträgt durchschnittlich 0,029s. Für ein Testbild und n Referenzbilder, mit $n \in \mathbb{N}$, ergibt sich somit eine Berechnungszeit von: $0,48s + 0,1s + n \cdot 0,029s$. Wählt man n sehr groß, dann fällt nur noch die Dauer für das Matching ins Gewicht. Durch die Reduktion der Referenzobjekte, konnte die Zeit für ein Match zweier Bilder um die Hälfte reduziert werden. Damit wurde gezeigt, dass je weniger Referenzobjekte und -bilder vorhanden sind, desto besser ist die Erkennungsrate, da es zu weniger falschen Zuordnungen kommt und desto kürzer ist auch die Berechnungszeit.

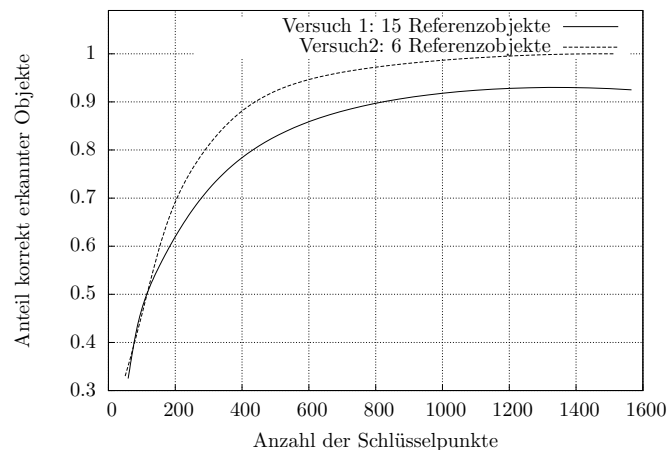


Abbildung 54: Zeigt die Qualität der Objekterkennung für Versuch eins und zwei. Je größer die Anzahl der Schlüsselpunkte ist, desto höher ist der Anteil korrekt erkannter Objekte.

8.2.3 Erkennung zwischen verschiedenen Objekten

Um zu zeigen, dass ein Objekt auch zwischen mehreren Objekten korrekt erkannt wird, sollen drei verschiedene Objekte zusammen auf einem Referenzbild angeordnet werden. Anschließend soll gezeigt werden, dass die Schlüsselpunkte der drei einzelnen Objekten korrekt auf das richtige Objekt in dem Referenzbild zugeordnet werden. Die grünen Linien in Abbildung 55 zeigen die Matches der Schlüsselpunkte. Trotz Verdeckung, Drehung und Skalierung der Objekte werden die Schlüsselpunkte auf den Objekten richtig zugeordnet.

Lediglich in Abbildung 56 ist ein falscher Match zu erkennen. Jedoch wird schnell klar, dass der Match, so falsch eigentlich gar nicht ist. In beiden Bildern sind auf der DVD und dem Spiel die Logos der FSK²² und er USK²³ abgebildet. In beiden Logos wird die Ziffer 6 dargestellt. Der Grund für die falsche Zuordnung der Ziffer 6 ist, dass im rechten Bild auf der DVD an der Ziffer 6 kein Schlüsselpunkt gefunden wurde, jedoch aber auf dem Spiel. Da also im linken und im rechten Bild ein Schlüsselpunkt auf der Ziffer 6 existiert, gab es hier auch einen passenden Match. Wären mehr Schlüsselpunkte berechnet worden, dann würde vermutlich irgendwann auf der DVD an der Ziffer 6 auch ein Schlüsselpunkt lokalisiert werden und würde vermutlich dann zum entsprechenden Schlüsselpunkt im linken Bild passen.

²²FSK: Freiwillige Selbstkontrolle

²³USK: Unterhaltungssoftware Selbstkontrolle



Abbildung 55: Korrekte Zuordnung der Schlüsselpunkte.

Natürlich soll in einer produktiven Anwendung nur ein Objekt auf einem Referenzbild zu sehen sein. Jedoch wäre eine Erkennung von mehreren Objekten eine sinnvolle Erweiterung. Eine Möglichkeit zu erkennen, dass verschiedene Objekte auf dem Bild sind, wäre die Schlüsselpunkte der Matches zu betrachten. Jeder Schlüsselpunkt in einem Bild hat eine eindeutige Indexnummer. Dieser Index ist ein einfacher Zähler, der für jeden weiteren Schlüsselpunkt um eins erhöht wird. Haben zwei Testbilder viele Matches auf einem Referenzbild, könnte geprüft werden, welche Schlüsselpunkte auf dem Referenzbild zu den beiden Testbildern gehören. Wenn es viele Matches aus zwei Testbildern auf das selbe Referenzbild gibt, dann kann davon ausgegangen werden, dass mehr als ein Objekt im Referenzbild abgebildet ist.

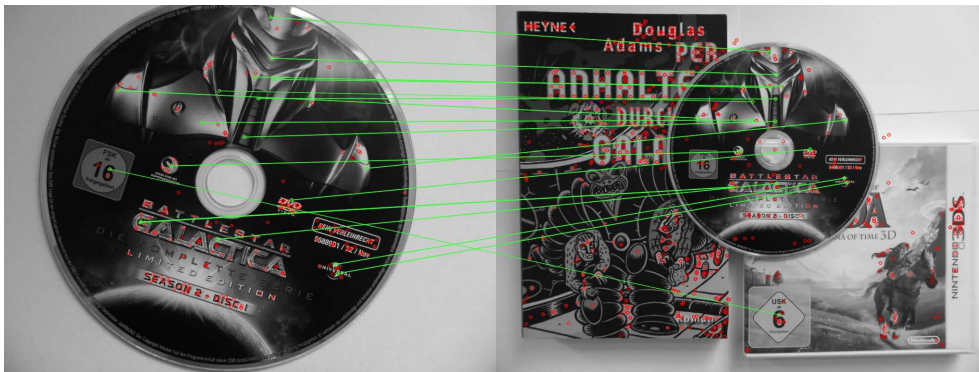


Abbildung 56: Beispiel für Ausreißer.

9 Zusammenfassung und Ausblick

In dieser Arbeit wurde eine visuelle Erkennung von Objekten als Basis für eine Augmented Reality Anwendung prototypisch umgesetzt. Es wurden Grundlagen für eine Augmented Reality Anwendung beschrieben, aus denen ein Konzept erarbeitet und implementiert wurde, mit der Objekte aufgenommen und erkannt werden können. Die Erkennung natürlicher Merkmale in einem Bild mit dem SIFT Verfahren hat sich als effektive Methode zur Erkennung von Bildern herausgestellt.

Es wurde gezeigt, dass bedingt durch ihre Form und Struktur nicht alle Objekte gleich gut erkennbar sind. Flache Objekte mit vielen Farben und starken Kontrasten sind am besten für eine visuelle Erkennung geeignet. Eine sorgfältige Auswahl von Objekten ist für eine AR Anwendung unbedingt notwendig. Die Menge der Schlüsselpunkte, die in einem Bild gefunden werden, ist abhängig von der Auflösung des Bildes. In Bildern mit hoher Auflösung können mehr Schlüsselpunkte gefunden werden, was eine mögliche Erkennung verbessern kann mit dem Preis der deutlich höheren Berechnungszeit sowohl für die Berechnung der Schlüsselpunkte, der Deskriptoren und für das Vergleichen zweier Bilder.

Für die Erkennung der Objekte mussten geeignete Methoden entwickelt werden, um eine qualitativ gute Aussage über das Ergebnis machen zu können. Dabei wurde gezeigt, dass eine einfache Suche nach den ähnlichsten Deskriptoren aus einem anderen Bild nicht sehr effektiv ist. Die Methode der stabilen Matches hat sich als deutlich effektiver erwiesen. Damit lassen sich Objekte sehr viel zuverlässiger erkennen. Auch mit weniger Schlüsselpunkten ist eine richtige Zuordnung eines Testbildes zu verschiedenen Referenzbildern möglich. Jedoch sagt hier die Anzahl der stabilen Matches noch nicht viel über die Qualität des Ergebnisses aus. Erst eine Betrachtung der Anzahl der stabilen Matches aus allen Bildern mit einem Match-Score führte zu einer qualitativen Aussage, ob das Bild mit den meisten stabilen Matches korrekt zugeordnet wurde. Es stellte sich heraus, dass durch eine Verringerung der Anzahl möglicher Objekte in einem Raum die Erkennung bei gleicher Anzahl an Schlüsselpunkten verbessert werden kann.

Anders formuliert, eine Verringerung von Objekten im Raum kann die Berechnungszeit reduzieren und dennoch die selbe Erkennungsrate liefern. Nachdem ein Objekt erkannt worden ist, können Informationen in Form von Bildern und Text zu dem Objekt angezeigt werden. Eine zukünftige Erweiterung könnte die Darstellung von Videos oder virtuelle 3D-Objekte sein. Eine Berechnungszeit von durchschnittlich einer halben Sekunde für ein Bild mit einem MegaPixel ist noch zu viel im damit eine AR Anwendung zu realisieren, welche die Objekte aus einem Live-Video Bild erkennt. Um eine flüssige Augmented Reality Anwendung zu realisieren, muss die Berechnungszeit noch deutlich reduziert werden. Durch eine sorgfältige Wahl der Parameter des SIFT Verfahrens, einer Verringerung der Menge an Objekten in einem Raum sowie der Reduzierung der Auflösung der Bilder lässt sich die Berechnungszeit weiter senken.

Das Verfahren der stabilen Matches und die Methode des Match-Scores können versagen, wenn zwei gleiche Objekte erkannt werden sollen. Daher wurde über eine Möglichkeit nachgedacht, die Objekte räumlich zu trennen und das Wissen über den Aufenthaltsort der Objekte zu verwenden. Da eine Lokalisierung eines jeden Objektes sehr aufwendig ist, sollten die Objekte indirekt lokalisiert werden. Darum wurde über eine Möglichkeit der Lokalisierung eines Benutzers nachgedacht. Unter der Annahme, dass bestimmte Objekte, wie z.B. die Ausstellungsobjekte in einem Museum nur sehr selten bewegt werden, kann durch eine Verknüpfung der Objekte mit einem konkreten Raum und über die Lokalisierung eines Benutzer innerhalb eines Raums eine indirekte Lokalisierung der Objekte vorgenommen werden. Es wurden verschiedene Methoden der Lokalisierung vorgestellt und diskutiert. Es hat sich gezeigt, dass bereits viele Methoden für eine Lokalisierung existieren. Jedoch sind nicht alle auf mobilen Geräten ohne zusätzliche Hardware anwendbar. Hier galt es eine Lösung zu finden, die nicht auf zusätzliche Hardware und aufwändige Installationen angewiesen ist und möglichst Technologien unterstützt, die auf den meisten mobilen Geräten zur Verfügung stehen. Daher wurde eine Lokalisierung mit WLAN und Bluetooth umgesetzt. Es wurde keine Infrastruktur aufgebaut, sondern die vorhandenen WLAN und Bluetooth Basisstationen verwendet. Für die Lokalisierung wurde ein dynamischer Zustandsschätzer auf Basis eines Bayes-Filters implementiert. Eine Lokalisierung mit WLAN ergab im besten Fall eine mittlere Genauigkeit von 5,85m bei einer Präzision von 84%. Die Genauigkeit von Bluetooth lag im besten Fall bei 6,56m bei einer Präzision von 58%. Eine Analyse der Genauigkeit, bezüglich der einzelnen Räume hat gezeigt, dass die Genauigkeit in den Räumen stark schwankt. Jedoch konnte in drei Räumen eine Genauigkeit zwischen 2,5m und 3,7m erzielt werden, was einen relativ guten Wert darstellt. Evtl. kann die Lokalisierung durch zusätzliche

Fingerprints noch verbessert werden. Zwar sind andere Lösungen für eine Lokalisierung genauer, jedoch liegt der Vorteil dieser Anwendung darin, dass die verwendete Technik mit den meisten modernen Smartphones kompatibel ist und die Benutzer nicht auf zusätzliche Hardware angewiesen sind.

Ändern sich jedoch Eigenschaften in der Umgebung, müssen die Fingerprints meistens neu aufgezeichnet werden. Hier könnte eine Lösung entwickelt werden, bei der im produktiven Betrieb die Anwendung auf Änderungen reagiert und diese selbständig im System aktualisiert. Außerdem sollte für eine produktive Anwendung unbedingt auf eine Client Server Architektur gesetzt werden. Die Benutzer bekommen nur die Daten die sie benötigen. Die Anwendung könnte somit potenziell beliebig viele Objekte enthalten. Ein weiteres Problem stellt sich aber mit der Frage nach den Rechten der Inhalte und der Bilder. Möchte ein Museum, dass Texte und Bilder auf den mobilen Geräten der Benutzer vorhanden sind? Hier fallen viele Probleme bezüglich der Rechte der Daten an, die dem Benutzer zur Verfügung gestellt werden.

Der Energieverbrauch der beiden Funktechnologien liegt noch im akzeptablen Bereich. Durch Ausnutzen von Bewegungsinformationen ist es möglich den Energieverbrauch zu reduzieren, indem die benötigten Sensoren bei Bedarf aktiviert werden.

Um eine bessere Lokalisierung zu ermöglichen könnten größere Räume durch virtuelle Bereiche aufgeteilt werden. Diese könnten z.B. durch Magnetfeld tags eingeteilt werden. Diese tags senden ein spezielles Magnetfeld aus, das von einem Sensor und der Anwendung erkannt werden kann. Eine andere Möglichkeit wäre es die Sendereichweite der WLAN Basisstationen zu reduzieren, da es hier zu weniger Überschneidungen mit anderen Funkzellen kommt. Der Check-In in ein Gebäude könnte z.B. für ein Museum neben GPS auch durch QR-Tags an Eintrittskarten gelöst werden. So könnte zusätzlich Werbung für die Anwendung gemacht werden.

Augmented Reality hält immer mehr Einzug in unser alltägliches Leben. Dabei sind Möglichkeiten für Anwendungen mit Augmented Reality sind noch lange nicht ausgeschöpft. Vor allem im Kontext eines Museums sind hier viele Szenarien denkbar. Vorstellbar wäre z.B. eine virtuelle Schatzsuche in einem Museum, bei der die Benutzer reale Objekte finden können, mit denen sie virtuelle Gegenstände einsammeln und in ihre eigene digitale Schatztruhe ablegen können.

Die Entwicklung von AR Anwendungen hat gerade erst begonnen. So arbeitet Google zur Zeit an der Entwicklung einer AR-Brille mit Namen Project Glass, die uns im Alltag in vielen Lebenssituation helfen soll.

A Anhang

A.1 Speicherung der Simulationsdaten

Das folgende Listing zeigt, wie die aufgezeichneten Simulationsdaten gespeichert werden. Die Daten werden im JSON Format gespeichert. Es handelt sich um eine Liste, deren Elemente nach einem Zeitstempel sortiert sind. Jede Messung hat einen Typ, der den Sensor beschreibt sowie den gemessenen Wert.

Code A.0 Simulationsdaten im JSON Format.

```
{
  "simulation-array": [
    {
      "timestamp": 1337360951,
      "room-name": "2.3.03",
      "wlan-array": [{
        "bssid": "F0:7D:68:83:A0:FC",
        "ssid": "name_funknezt",
        "dBm": -73
      }]
    },
    {
      "timestamp": 1337362632,
      "room-name": "2.3.02",
      "wlan-array": [{
        "bssid": "14:D6:4D:C7:8F:64",
        "ssid": "name_funknezt",
        "dBm": -68
      }]
    }
  ]
}
```

A.2 ER Datenmodell

Hier ist das detaillierte ER-Modell zu sehen. Zur Verbesserung der Übersichtlichkeit wurde es in zwei Teile aufgeteilt. Im ersten Teil sind die Beziehungen der Räume, Objekte und Fingerprints zu sehen. Jedes Objekt verfügt über eine Bezeichnung und eine optionale Beschreibung. Zu jedem Objekt können Features (Merkmale) gespeichert werden. Die Features dienen zur visuellen Erkennung der Objekte. Es werden nur die Referenzen auf die Dateien der Features in der Datenbank gespeichert. Zur Lokalisierung verfügt jeder Raum über Fingerprints. Diese kapseln die gemessenen Sensordaten. Ein Fingerprint verfügt über eine eindeutige Position. Abbildung 58 zeigt, dass jeder Fingerprint

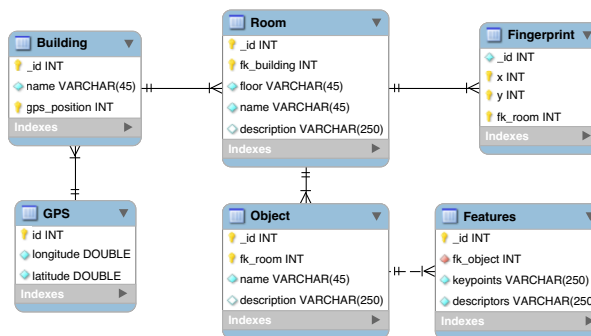


Abbildung 57: ER Modell 1

über verschiedene Sensorwerte verfügt. Ein Sensorwert ist eindeutig für eine Position und einer von vier Himmelsrichtungen bestimmt. Die Himmelsrichtungen sind in der Tabelle *Direction* definiert: (Nord, Süd, Ost und West).

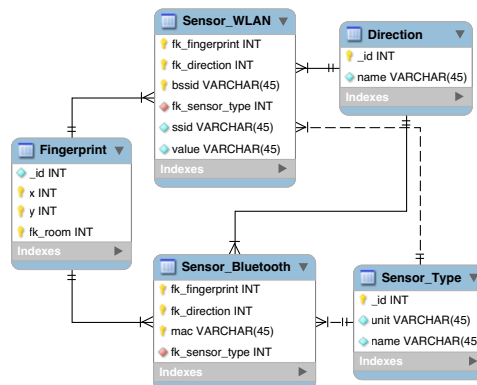


Abbildung 58: ER Modell 2

A.3 Konfigurationsdateien

In diesem Abschnitt sind Auszüge der Konfigurationsdateien dargestellt. Die Konfigurationsdateien werden genutzt, um die Datenbank auf dem Entwicklungsgerät zu initialisieren. Jede Konfigurationsdatei bildet genau eine Instanz einer Entität ab. Auf diese Weise können die Daten einfach über eine Konfigurationsdatei erstellt, geändert und geladen werden ohne direkt in die Datenbank eingreifen zu müssen. Für jede Entität steht eine eigene Konfigurationsdatei zur Verfügung. Es wäre auch möglich alle Konfigurationen in eine Datei zu schreiben. Dann bräuchte jede Instanz noch einen Eintrag, zu welcher Entität diese gehört.

```
5  ---
    _id: 1
    Bezeichnung: TGS
    ---
    _id: 2
    Bezeichnung: Wilhelminenhof Gebäude C
```

Listing A.1: Konfigurationsdatei für die Gebäude.

```
5  ---
    _id: 1
    FK_Gebaeude: 1
    Etage: 3
    Bezeichnung: 2.3.03
    Beschreibung: Arbeitsraum
    ---
10  _id: 2
    FK_Gebaeude: 1
    Etage: 3
    Bezeichnung: 2.3.04
    Beschreibung: Arbeitsraum
```

Listing A.2: Konfigurationsdatei für die Räume.

```
5  ---
    _id: 1
    FK_Raum: 1
    Bezeichnung: Drucker
    Beschreibung: Ein Drucker von HP mit Scanner.
    ---
    _id: 2
    FK_Raum: 2
```

```
10 | Bezeichnung: Projektionslampe  
    | Beschreibung: Eine Projektionslampe für Lichtinstallationen.
```

Listing A.3: Konfigurationsdatei für die Objekte.

A.4 Tabellen zur Auswertung der Lokalisierung und der visuellen Erkennung

Raum	Mittel (m)	Stabw (m)	Max (m)
2.3.01	8,09	1,44	9,57
2.3.02	8,31	3,26	11,98
2.3.03	2,45	1,01	3,72
2.3.04	6,67	1,47	8,45
2.3.06	2,52	0,97	3,83
2.3.07	5,32	2,16	8,56
2.3.08	3,74	1,29	5,23

Tabelle 11: Zeigt die Ergebnisse der WLAN Messung in den einzelnen Räumen des TGS, Gebäude 2, dritte Etage. Für jeden Raum wurde die mittlere Genauigkeit, die Standardabweichung (Stabw) und der maximale Wert angegeben.

Referenzbild	Bester Match	Match-Score
Beamer	Beamer	0,97
Box1_a	Box1_a	0,94
Box1_b	Box1_b	0,98
Box1_c	Box1_c	0,95
Box1_d	Box1_d	0,92
Box1_e	Box1_e	0,97
Box1_f	Box1_f	1,00
Buch1_a	Buch1_a	0,97
Buch1_b	Buch1_b	0,83
Buch1_c	Buch1_c	0,72
Buch2	Buch2	0,91
Buch3	Buch3	0,94
Drucker	Drucker	0,94
Glas1_a	Glas1_a	0,50
Glas1_b	—	0,00
Gyroskop1_a	Gyroskop1_b	0,96
Gyroskop1_b	Gyroskop1_b	0,94
Gyroskop1_c	Gyroskop1_c	0,81
Gyroskop1_d	Gyroskop1_d	0,88
Gyroskop1_e	Gyroskop1_e	0,98

iPhone	—	0,00
Karton	Karton	0,97
Locher	Locher	0,66
Maus1	Maus1	0,64
Maus2	Maus2	0,58
Maus3	Maus3	0,90
Maus4	Maus4	0,79
Tasse	Tasse	0,94
Tastatur1_a	Tastatur1_a	0,98
Tastatur2	Tastatur2	0,98
Tastatur3_a	—	0,00
Tastatur3_b	Tastatur3_b	0,59
Tastatur4	Tastatur4	0,96
Telefon	Telefon	0,92

Tabelle 12: Zeigt welche Referenzbilder geeignet sind. Jedes Referenzbild wird mit allen anderen Referenzbilder verglichen. Rot markierte Zeilen haben einen Match-Score $\leq 0,75$ und werden aussortiert, da sie zu anderen Referenzbildern zu ähnlich sind.

Schlüsselpunkte	Anteil Korrekt	Zeit pro Match (s)
59	0,325	0,0008
71	0,375	0,0011
87	0,450	0,0016
106	0,550	0,0023
128	0,475	0,0034
157	0,600	0,0049
195	0,625	0,0074
244	0,650	0,0113
306	0,800	0,0175
403	0,850	0,0298
590	0,900	0,0622
1026	0,950	0,1814
1566	0,925	0,4150

Tabelle 13: Anteil der korrekt erkannten Testbilder bei unterschiedlich vielen Schlüsselpunkten. Versuch mit 25 Referenzbilder von 15 Objekten und 40 Testbildern.

Schlüsselpunkte	Anteil Korrekt	Zeit pro Match (s)
50	0,33	0,0009
110	0,47	0,0025
134	0,57	0,0036
165	0,61	0,0055
198	0,80	0,0075
235	0,71	0,0103
278	0,80	0,0144
334	0,85	0,0206
399	0,95	0,0294
479	0,95	0,0421
597	0,95	0,0652
817	1,00	0,1205
1311	1,00	0,3052
1961	1,00	0,6771

Tabelle 14: Anteil der korrekt erkannten Testbilder bei unterschiedlich vielen Schlüsselpunkten. Versuch mit 13 Referenzbilder von 6 Objekten und 21 Testbildern.

A.5 Inhalt der CD

Auf der CD befindet sich diese Arbeit als PDF sowie der entwickelte Prototyp. Um den Prototypen zu nutzen muss die OpenCV Bibliothek, das Android SDK in der Version 2.3.3 sowie das NDK installiert sein. Die CD enthält folgende Ordner:

- **Auswertung:** Enthält die Tabellen für die Auswertung der Lokalisierung und der visuellen Erkennung.
- **Bilder:** Enthält alle Abbildung in dieser Arbeit.
- **Referenzobjekte:** Enthält die Bilder der Referenzobjekte.
- **Testobjekte:** Enthält die Bilder der Testobjekte.
- **Programmcode:** Enthält den entwickelten Prototypen ObjektTracker.

Literatur

- [ABB⁺01] AZUMA, Ronald ; BAILLOT, Yohan ; BEHRINGER, Reinhold ; FEINER, Steven ; JULIER, Simon ; MACINTYRE, Blair: Recent Advances in Augmented Reality. In: *IEEE Computer Graphics and Applications* 21 (2001), Nr. 6, S. 34–47
- [Azu97] AZUMA, Ronald: A Survey of Augmented Reality. In: *Presence* 6 (1997), Nr. 4, S. 355–385
- [BHK⁺09] BEUTH, Klaus ; HANEBUTH, Richard ; KURZ, Günter ; LÜDERS, Christian ; BREIDE, Stephan: *Nachrichtentechnik - Elektronik 7*. Würzburg : Vogel Verlag Und Druck, 2009. – ISBN 978-3-834-33108-3
- [Bil12] BILKE, Andreas: *Ortung und Navigation mit mobilen Geräten in Gebäuden*, Hochschule für Technik und Wirtschaft, Masterarbeit, 2012
- [Bos11] BOSCH, Karl: *Elementare Einführung in die Wahrscheinlichkeitsrechnung: Mit 82 Beispielen und 73 Übungsaufgaben mit vollständigem Lösungsweg*. 11., akt. Aufl. 2011. Vieweg+Teubner Verlag, 2011. – ISBN 9783834818614
- [BP00] BAHL, Paramvir ; PADMANABHAN, Venkata N.: RADAR: An In-Building RF-Based User Location and Tracking System. In: *INFOCOM*, 2000, S. 775–784
- [Bro86] BROCKHAUS: *Brockhaus Enzyklopadie: In vierundzwanzig Banden (German Edition)*. 19. völlig neu bearbeitete Aufl. F.A. Brockhaus, 1986. – ISBN 9783765311000
- [BTG06] BAY, Herbert ; TUYTELAARS, Tinne ; GOOL, Luc V.: Surf: Speeded up robust features. In: *In ECCV*, 2006, S. 404–417
- [Bun10] BUNK, Christian: *Augmented Reality*, Hochschule für Technik und Wirtschaft, Seminararbeit, 2010

- [Bun11] BUNK, Christian: *Augmented Reality Techniken für den Einsatz in mobilen Multimediaguides*, Hochschule für Technik und Wirtschaft, Forschungsprojekt 1, 2011
- [Bun12] BUNK, Christian: *Visuelle und elektronische Objektidentifikation zur Nutzung in Augmented Reality Anwendungen*, Hochschule für Technik und Wirtschaft, Forschungsprojekt 2, 2012
- [CF08] CAWOOD, Stephen ; FIALA, Mark: *Augmented Reality: A Practical Guide*. 1. Pragmatic Bookshelf, 2008. – ISBN 9781934356036
- [Dev12] DEVLIN, Keith: *Die Berechnung des Glücks: Eine Reise in die Geschichte der Mathematik*. Deutscher Taschenbuch Verlag, 2012. – ISBN 9783423347044
- [DGA00] DOUCET, Arnaud ; GODSILL, Simon ; ANDRIEU, Christophe: On Sequential Monte Carlo Sampling Methods for Bayesian Filtering. In: *STATISTICS AND COMPUTING* 10 (2000), Nr. 3, S. 197–208
- [DH04] DODEL, Hans ; HÄUPLER, Dieter: *Satellitennavigation*. 1. Aufl. Heidelberg : Hüthig, 2004. – ISBN 978-3-826-65036-9
- [DKV09] DOBRINSKI, Paul ; KRAKAU, Gunter ; VOGEL, Anselm: *Physik für Ingenieure (German Edition)*. 12, akt. Aufl. 2010. Vieweg+Teubner Verlag, 2009. – ISBN 9783834805805
- [dtv99] DTV: *dtv - Lexikon in 20 Bänden. Neuausgabe. (German Edition)*. Dtv, 1999. – ISBN 9783423059985
- [Fab80] FABECK, Wolf von: *Kreiselgeräte*. 1. Vogel Verlag Und Druck, 1980. – ISBN 9783802306129
- [FHL⁺03] FOX, Dieter ; HIGHTOWER, Jeffrey ; LIAO, Lin ; SCHULZ, Dirk ; BORRIELLO, Gaetano: Bayesian filtering for location estimation. In: *IEEE Pervasive Computing* 2 (2003), S. 24–33
- [Fin08] FINKENZELLER, Klaus: *RFID-Handbuch: : Grundlagen und praktische Anwendungen von Transpondern, kontaktlosen Chipkarten und NFC*. Hanser Fachbuchverlag, 2008. – ISBN 9783446412002
- [Fur] FURNESS, Thomas A.: *The Virtual Retinal Display: A New Technology for Virtual Reality und Augmented Vision in Medicine*

- [Ger06] GERTHSEN, Christian: *Gerthsen Physik (Springer-Lehrbuch) (German Edition)*. 23., überarb. Aufl. Springer, 2006. – ISBN 9783540254218
- [GK09] GESSLER, Ralf ; KRAUSE, Thomas: *Wireless-Netzwerke für den Nahbereich: Eingebettete Funkssysteme: Vergleich von standardisierten und proprietären Verfahren (German Edition)*. 2009. Vieweg+Teubner Verlag, 2009. – ISBN 9783834802477
- [GL02] GIAGLIS, George M. ; LEKAKOS, George: A Taxonomy of Indoor and Outdoor Positioning Techniques for Mobile Location Services. In: *ACM SIGecom Exchanges* 3 (2002)
- [GSS93] GORDON, N. J. ; SALMOND, D. J. ; SMITH, A. F. M.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In: *Radar and Signal Processing, IEE Proceedings F* 140 (1993), April, Nr. 2, S. 107–113. – ISSN 0956–375X
- [HB01] HIGHTOWER, Jeffrey ; BORRIELLO, Gaetano: A Survey and Taxonomy of Location Systems for Ubiquitous Computing / IEEE Computer. 2001. – Forschungsbericht. – 57–66 S.
- [Hey05] HEYMANN, Sebastian: *Implementierung und Evaluierung von Video Feature Tracking auf moderner Grafikkhardware*, Bauhaus Universität Weimar, Diplomarbeit, 2005
- [HHS⁺99] HARTER, Andy ; HOPPER, Andy ; STEGGLES, Pete ; WARD, Andy ; WEBSTER, Paul: The Anatomy of a Context-Aware Application. In: *WIRELESS NETWORKS, VOL 8* (1999), S. 187–197
- [HS11] HEDDERICH, Jürgen ; SACHS, Lothar: *Angewandte Statistik: Methodensammlung mit R*. 14. Aufl. 2012. Springer Berlin Heidelberg, 2011. – ISBN 9783642244001
- [Jäh05] JÄHNE, Bernd: *Digitale Bildverarbeitung*. 6., überarb. u. erw. Aufl. Springer Berlin Heidelberg, 2005. – ISBN 9783540249993
- [KHKE05] KING, Thomas ; HAENSELMANN, Thomas ; KOPF, Stephan ; EFFELBERG, Wolfgang: Positionierung mit Wireless-LAN und Bluetooth / University of Mannheim Fakultät für Mathematik und Informatik –. 68159 Mannheim, Germany, Dezember 2005 (TR-2005-015). – Forschungsbericht

- [Lag11] *OpenCV 2 Computer Vision Application Programming Cookbook*. Packt Publishing, 2011. – ISBN 9781849513241
- [Low99] LOWE, David G.: *Object Recognition from Local Scale-Invariant Features*. 1999
- [Low03] LOWE, David G.: *Distinctive Image Features from Scale-Invariant Keypoints*. 2003
- [Man09] MANSFELD, Werner: *Satellitenortung Und Navigation - Grundlagen, Wirkungsweise Und Anwendung Globaler Satellitennavigationssysteme*. 3. überarb. u. akt. Aufl. 2010. Vieweg+Teubner Verlag, 2009. – ISBN 978-3-834-80611-6
- [MGNR12] MÜLLER-GRONBACH, Thomas ; NOVAK, Erich ; RITTER, Klaus: *Monte Carlo-Algorithmen*. 2012. Aufl. Berlin, Heidelberg : Springer, 2012. – ISBN 978-3-540-89140-6
- [MRS08] MANNING, Christopher D. ; RAGHAVAN, Prabhakar ; SCHÜTZE, Hinrich: *Introduction to Information Retrieval*. Bd. 1. Cambridge University Press, 2008 <http://nlp.stanford.edu/IR-book/pdf/irbookprint.pdf>. – ISBN 9780521865715. – [Online Ausgabe; Stand 16. Mai 2012]
- [NLLP03] NI, Lionel M. ; LIU, Yunhao ; LAU, Yiu C. ; PATIL, Abhishek P.: LAND-MARC: Indoor Location Sensing Using Active RFID. In: *IEEE International Conference on Pervasive Computing and Communications (2003)*, S. 407–415. ISBN 0-7695-1893-1
- [Obs09] OBST, Marcus: *Untersuchung zur kooperativen Fahrzeuglokalisierung in dezentralen Sensornetzen*, Technische Universität Chemnitz, Diplomarbeit, 2009
- [OXM02] OWEN, Charles B. ; XIAO, Fan ; MIDDLELIN, Paul: What is the Best Fiducial? In: *The First IEEE International Augmented Reality Toolkit Workshop*, 2002, S. 98–105
- [PCB00] PRIYANTHA, Nissanka B. ; CHAKRABORTY, Anit ; BALAKRISHNAN, Hari: The Cricket location-support system. In: *MOBICOM*, 2000, S. 32–43

- [Pre07] PRESTLE, Benjamin: *Konzeption und Implementierung eines Prototypen zur Identifizierung stationärer Objekte anhand digitaler Photos*, Fachhochschule für Technik und Wirtschaft Berlin, Diplomarbeit, 2007
- [RAG04] RISTIC, Branko ; ARULAMPALAM, Sanjeev ; GORDON, Neil: *Beyond the Kalman Filter: Particle Filters for Tracking Applications (Artech House Radar Library)*. Artech Print on Demand, 2004. – ISBN 9781580536318
- [Rot05] ROTH, Jörg: *Mobile Computing: Grundlagen, Technik, Konzepte*. 2., aktualis. A. Dpunkt Verlag, 2005. – ISBN 9783898643665
- [Sau10] SAUTER, Martin: *Grundkurs Mobile Kommunikationssysteme: UMTS, HSD-PA und LTE, GSM, GPRS und Wireless LAN (German Edition)*. 4., überarb. und erw. Aufl. 2011. Vieweg+Teubner Verlag, 2010. – ISBN 9783834814074
- [SCH03] SCHILLER: *Mobilkommunikation*. 2. Pearson Studium, 2003. – ISBN 9783827370600
- [SSTa08] STRANG, Thomas ; SCHUBERT, Frank ; THÖLERT, Steffen ; AL., Rainer O.: *Lokalisierungsverfahren*. Deutsches Zentrum für Luft- und Raumfahrt (DLR) - Institut für Kommunikation und Navigation, 2008
- [TM08] TUYTELAARS, Tinne ; MIKOLAJCZYK, Krystian: Local Invariant Feature Detectors: A Survey. In: *FnT Comp. Graphics and Vision*, 2008, S. 177–280
- [WBW11] WAHRIG-BURFEIND, Renate ; WAHRIG, Gerhard: *Brockhaus WAHRIG Deutsches Wörterbuch*. 9. Brockhaus in der Wissenmedia, 2011. – ISBN 9783577075954
- [Wen07] WENDEL, Jan: *Integrierte Navigationssysteme*. Oldenbourg Wissensch.Vlg, 2007. – ISBN 9783486595154. – [eBook; Stand 23. Mai 2012]
- [WFG92] WANT, Roy ; FALCAO, Veronica ; GIBBONS, Jon: The Active Badge Location System. In: *ACM Transactions on Information Systems* 10 (1992), S. 91–102
- [Wid10] WIDYAWAN: *Indoor Localisation: State of the Art and Novel Algorithms: Particle Filter, Fingerprinting and Data Fusion for Opportunistic Indoor Localisation*. LAP LAMBERT Academic Publishing, 2010. – ISBN 9783838344294

- [WJH97] WARD, A. ; JONES, A. ; HOPPER, A.: A New Location Technique for the Active Office. In: *IEEE Personal Communications* 4 (1997), Oktober, Nr. 5, S. 42–47. – ISSN 10709916
- [Zap06] ZAPOTOCZKY, Johannes: *Hybride Funkortung: Adaptive Aggregation von RFID, WLAN und ZigBee*, Humboldt Universität zu Berlin, Diplomarbeit, 2006
- [ZLN07] ZHAO, Yiyang ; LIU, Yunhao ; NI, Lionel M.: VIRE: Active RFID-based Localization Using Virtual Reference Elimination. In: *ICPP*, 2007, S. 56
- [ZS10] ZEITLER, Jürgen ; SIMON, Günter: *Physik für Techniker*. 6. aktualisierte Auflage. München, Wien : Hanser Verlag, 2010. – ISBN 978–3–446–41048–0

Internetquellen

- [1] INFORMATIONSTECHNOLOGIE, ITWissen: D.: *Augmented Reality*. [http : //www.itwi s sen.info/definition/lexikon/augmented-reality-AR-Erweiterte-Realitaet.html](http://www.itwissen.info/definition/lexikon/augmented-reality-AR-Erweiterte-Realitaet.html), 2012. – [Online; Stand 30. April 2012]
- [2] *ARToolKit*. <http://www.hitl.washington.edu/artoolkit/>, 2012. – [Online; Stand 01. Mai 2012]
- [3] WIKIPEDIA: *QR-Code* — *Wikipedia, Die freie Enzyklopädie*. <http://de.wikipedia.org/w/index.php?title=QR-Code&oldid=102999797>, 2012. – [Online; Stand 21. Mai 2012]
- [4] WAVE, Denso: *QR Code Features*. <http://www.denso-wave.com/qrcode/qrcodefeature-e.html>, 2012. – [Online; Stand 21. Mai 2012]
- [5] *QCAR*. <https://ar.qualcomm.at/qdevnet/sdk>, 2012. – [Online; Stand 01. Mai 2012]
- [6] DANIEL WAGNER, Thomas P. ; SCHMALSTIEG, Dieter: *The Invisible Train*. http://studierstube.icg.tugraz.at/invisible_train/, 2004. – [Online; Stand 27. Juli 2012]
- [7] WIKIPEDIA: *Bildverarbeitung* — *Wikipedia, Die freie Enzyklopädie*. <http://de.wikipedia.org/w/index.php?title=Bildverarbeitung&oldid=103540468>, 2012. – [Online; Stand 29. Mai 2012]
- [8] *String*. <http://www.poweredbystring.com/>, 2012. – [Online; Stand 01. Mai 2012]
- [9] *Usnap*. <http://mobile.usnap.de/>, 2012. – [Online; Stand 20. August 2012]
- [10] WIKIPEDIA: *Lokalisierung* — *Wikipedia, Die freie Enzyklopädie*. <http://de.wikipedia.org/w/index.php?title=Lokalisierung&oldid=100610434>, 2012. – [Online; Stand 24. April 2012]

- [11] WIKIPEDIA: *Time of arrival* — *Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Time_of_arrival&oldid=489154971, 2012. – [Online; Stand 26. April 2012]
- [12] INFORMATIONSTECHNOLOGIE, ITWissen: D.: *TDOA (time difference of arrival)*. <http://www.itwissen.info/definition/lexikon/TDOA-time-difference-of-arrival.html>, 2012. – [Online; Stand 26. April 2012]
- [13] KÖLLER, Jürgen: *Hyperbel*. <http://www.mathematische-basteleien.de/hyperbel.htm>, 2007. – [Online; Stand 08. Juni 2012]
- [14] WIKIPEDIA: *Freiraumdämpfung* — *Wikipedia, Die freie Enzyklopädie*. <http://de.wikipedia.org/w/index.php?title=Freiraumd%C3%A4mpfung&oldid=100138606>, 2012. – [Online; Stand 15. Mai 2012]
- [15] WIKIPEDIA: *Leistungspegel* — *Wikipedia, Die freie Enzyklopädie*. <http://de.wikipedia.org/w/index.php?title=Leistungspegel&oldid=99492493>, 2012. – [Online; Stand 15. Mai 2012]
- [16] DÖING, Sönke: *Bild einer Funkzelle*. <http://www.fh-wedel.de/~si/seminare/ss04/Ausarbeitung/8.Doering/gsmnetz.gif>, 2012. – [Online; Stand 6. Juli 2012]
- [17] WIKIPEDIA: *Globales Navigationssatellitensystem* — *Wikipedia, Die freie Enzyklopädie*. http://de.wikipedia.org/w/index.php?title=Globales_Navigationssatellitensystem&oldid=101591393, 2012. – [Online; Stand 4. Mai 2012]
- [18] KOMPFF, Martin: *GPS - Entfernungsberechnung*. <http://www.kompf.de/gps/distcalc.html>, 2012. – [Online; Stand 18. Juli 2012]
- [19] WIKIPEDIA: *Zufallsvariable* — *Wikipedia, Die freie Enzyklopädie*. <http://de.wikipedia.org/w/index.php?title=Zufallsvariable&oldid=103389061>, 2012. – [Online; Stand 31. Mai 2012]
- [20] WIKIPEDIA: *Lumen (Einheit)* — *Wikipedia, Die freie Enzyklopädie*. [http://de.wikipedia.org/w/index.php?title=Lumen_\(Einheit\)&oldid=103377948](http://de.wikipedia.org/w/index.php?title=Lumen_(Einheit)&oldid=103377948), 2012. – [Online; Stand 20. Mai 2012]
- [21] INFORMATIONSTECHNOLOGIE, ITWissen: D.: *Magnetfeldsensor*. <http://www.itwissen.info/definition/lexikon/>

- Magnetfeldsensor-magnetic-field-sensor.html, 2012. – [Online; Stand 21. Mai 2012]
- [22] WIKIPEDIA: *Luftdruck* — *Wikipedia, Die freie Enzyklopädie*. <http://de.wikipedia.org/w/index.php?title=Luftdruck&oldid=103302176>, 2012. – [Online; Stand 20. Mai 2012]
- [23] WIKIPEDIA: *Druck (Physik)* — *Wikipedia, Die freie Enzyklopädie*. [http://de.wikipedia.org/w/index.php?title=Druck_\(Physik\)&oldid=103368748](http://de.wikipedia.org/w/index.php?title=Druck_(Physik)&oldid=103368748), 2012. – [Online; Stand 22. Mai 2012]
- [24] GOOGLE: *Sensor Event*. <http://developer.android.com/reference/android/hardware/SensorEvent.html>, 2012. – [Online; Stand 20. Mai 2012]
- [25] WIKIPEDIA: *Kreiselinstrument* — *Wikipedia, Die freie Enzyklopädie*. <http://de.wikipedia.org/w/index.php?title=Kreiselinstrument&oldid=103519293>, 2012. – [Online; Stand 24. Mai 2012]
- [26] WIKIPEDIA: *Kosinus-Ähnlichkeit* — *Wikipedia, Die freie Enzyklopädie*. <http://de.wikipedia.org/w/index.php?title=Kosinus-%C3%84hnlichkeit&oldid=103172714>, 2012. – [Online; Stand 16. Mai 2012]
- [27] OPENCV: *OpenCV*. <http://opencv.org/>, 2012. – [Online; Stand 27. Juli 2012]
- [28] OFTERDINGER, Adrian: *Java Native Interface*. http://www.sigs.de/publications/js/2005/05/ofterdinger_JS_05_05.pdf, 2012. – [Online; Stand 19. Juni 2012]
- [29] OPENCV: *OpenCV - Basic Structures*. http://opencv.willowgarage.com/documentation/cpp/basic_structures.html, 2012. – [Online; Stand 20. Juni 2012]
- [30] WIKIPEDIA: *Wireless Local Area Network* — *Wikipedia, Die freie Enzyklopädie*. http://de.wikipedia.org/w/index.php?title=Wireless_Local_Area_Network&oldid=102187921, 2012. – [Online; Stand 27. April 2012]

Abkürzungsverzeichnis

AoA	Angle of Arrival.
BTS	Base Transceiver Station.
CoO	Cell of Origin.
DAO	Data Access Object.
DoG	Difference of Gaussian.
GPS	Global Positioning System.
IEEE	Institute of Electrical and Electronics Engineers.
JNI	Java Native Interface.
JSON	JavaScript Object Notation.
LANDMARC	Location Identification based on Dynamic Active RFID Calibration.
MAC	Media Access Control.
NDK	Native Development Kit.
NFC	Near Field Communication.
RFID	Radio Frequency Identification.
RSSI	Received Signal Strength Indicator.
SIFT	Scale-Invariant Feature Transform.

TDoA	Time Difference of Arrival.
ToA	Time of Arrival.
VIRE	Virtual Reference Elimination.
WLAN	Wireless Local Area Network.

Stichwortverzeichnis

Bluetooth wurde entwickelt, um über kurze Entfernungen (wenige Meter) in Funknetzwerken zu kommunizieren [GK09]. Ziel war es verschiedene Peripheriegeräte miteinander zu verbinden [SCH03]. Die Entwicklung von Bluetooth wurde von verschiedenen Firmen wie Ericsson, IBM, Intel, Nokia und Toshiba angestoßen, welche 1998 die Bluetooth Special Interest Group (BSIG) gründeten [BHK⁺09, S. 716ff]. Jedes Bluetooth Modul kann über eine 48 Bit Media Access Control (MAC) Adresse identifiziert werden. Der Name Bluetooth wurde zu Ehren des Königs Harald von Dänemark und Norwegen gewählt, welcher den Beinamen Blauzahn (engl. bluetooth) trug. .

Fingerprint (dt. Fingerabdruck) ist eine eindeutige Signatur. Im Kontext der Lokalisierung ist ein Fingerprint eine eindeutige unverwechselbare Eigenschaft an einer Position.

Leistungspegel ist eine logarithmische Größe die im Verhältnis zu einer Bezugsgröße steht [BHK⁺09, S. 38 f.]. Der Leistungspegel ist eine Größe ohne physikalische Einheit. Um jedoch die Rechenvorschrift anzudeuten mit der er entstanden ist, wird die Einheit Dezibel dB angehängen. Der Leistungspegel ist definiert durch die Ausgangsleistung $P_a = U_a * I_a$ und der Eingangsleistung $P_e = U_e * I_e$:

$$L_p = 10 \lg \frac{P_a}{P_e} \text{ dB}$$

I ist die Stromstärke und U die elektrische Spannung. Wird als Bezugsgröße $1mW$ verwendet, wird der Leistungspegel in dBm angegeben:

$$L_p = 10 \lg \frac{P_a}{1mW} \text{ dBm}$$

NFC ist eine drahtlose Funktechnik im Frequenzbereich 13.56 MHz. Mit NFC können Daten bis zu 10 cm übertragen werden. NFC ist zu RFID Systemen kompatibel..

WLAN [30] (dt. „drahtloses lokales Netzwerk“) ist ein lokales Funknetz, nach dem Standard des Institute of Electrical and Electronics Engineers (IEEE) 802.11. WLAN arbeitet im Frequenzband von 2.4 GHz. Synonym wird oft auch Wi-Fi verwendet.

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel verfasst habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Ort, Datum

Unterschrift