

Computer Systems and Telematics

# Evaluation distanzbasierter Lokalisierung in Gebäuden

Dissertation zur Erlangung des akademischen Grades eines  
Doktors der Naturwissenschaften (Dr. rer. nat.) im Fachbereich  
Mathematik und Informatik der Freien Universität Berlin

vorgelegt von

**Simon Schmitt**

Datum der Disputation: 04.11.2019

Erstgutachter: Prof. Dr.-Ing. Jochen Schiller, Freie Universität Berlin

Zweitgutachter: Prof. Dr. Marcel Kyas, Reykjavik University

---

Institut für Informatik, Freie Universität Berlin, Deutschland

18. Dezember 2019



Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, sind als solche gekennzeichnet. Die Zeichnungen oder Abbildungen sind von mir selbst erstellt worden oder mit entsprechenden Quellennachweisen versehen. Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner Prüfungsbehörde eingereicht worden.

Berlin, den 18. Dezember 2019

---

(Simon Schmitt)





---

# Danksagung

An erster Stelle geht mein Dank an meine beiden Betreuer Jochen Schiller und Marcel Kyas, die mir diese Dissertation und den internationalen Austausch ermöglichten. Ich danke ganz besonders Marcel Kyas und Stephan Adler für unzähliges Ideenaustauschen. Ohne beide wäre das Gros meiner Veröffentlichungen nicht möglich gewesen. Weiterhin schätze ich den Austausch und die Arbeit mit Katinka Wolter und Thomas Willemsen. Wesentlich zum Erfolg beigetragen haben auch meine beiden langjährigen studentischen Mitarbeiter Larissa Zech und Niels Groth. Beide zeigten bis in die Nacht hinein riesigen Einsatz und standen mir in unzähligen Diskussionen beim Ideenjonglieren bei. Last but not least geht auch ein sehr hoher Dank für eine unermüdete Motivierung und viele offenen Ohren an alle weiteren Menschen die mich auf diesem Weg begleiteten.



---

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>ix</b>
<b>Tabellenverzeichnis</b>	<b>xiii</b>
<b>Algorithmenverzeichnis</b>	<b>xv</b>
<b>Abkürzungsverzeichnis</b>	<b>xvi</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Wissenschaftlicher Beitrag . . . . .	5
<b>2 Evaluation von Lokalisierungssystemen</b>	<b>9</b>
2.1 Experimentelle Evaluation in der Wissenschaft . . . . .	9
2.1.1 Motivation . . . . .	10
2.1.2 Methodik . . . . .	12
2.1.3 Ergebnisse . . . . .	17
2.1.4 Genauigkeit der Studie . . . . .	30
2.1.5 Zusammenfassung . . . . .	33
2.2 Einführung eines automatisierten Referenzsystems . . . . .	34
2.2.1 Motivation . . . . .	35
2.2.2 Automatisches Referenzsystem . . . . .	37
2.2.3 Evaluation . . . . .	41
2.2.4 Anwendungsfall: Vergleich von Algorithmen . . . . .	42
2.2.5 Zusammenfassung . . . . .	43
<b>3 Virtuelles Testbed für distanzbasierte Lokalisierung</b>	<b>45</b>
3.1 Motivation . . . . .	47
3.2 Aufbau des virtuellen Testbeds . . . . .	49
3.2.1 Vollständige Abdeckung . . . . .	49
3.2.2 Datenformat und Datenbank . . . . .	51
3.2.3 Benutzerschnittstelle . . . . .	53
3.3 Vorstellung der Datensätze . . . . .	55
3.3.1 Konfiguration des Referenzsystems . . . . .	55
3.3.2 Modular Sensor Board . . . . .	55
3.3.3 Raumszenario . . . . .	57

3.3.4	Gebäudeszenario . . . . .	58
3.4	Evaluation . . . . .	61
3.4.1	Machbarkeitsnachweis . . . . .	61
3.4.2	Reale Büroumgebung . . . . .	65
3.5	Zusammenfassung . . . . .	68
<b>4</b>	<b>Räumliche Fehlerverteilung bei distanzbasierter Lokalisierung</b>	<b>70</b>
4.1	Einleitung . . . . .	71
4.2	Pfadverlust und Mehrwege-Effekte . . . . .	72
4.2.1	Motivation . . . . .	72
4.2.2	Weitgehend NLOS-freie Ankerpositionen . . . . .	75
4.2.3	Ankerpositionen mit deutlichen NLOS-Effekten . . . . .	79
4.2.4	Zusammenfassung . . . . .	81
4.3	Betrachtung von Lokisierungsalgorithmen . . . . .	82
4.3.1	Motivation . . . . .	82
4.3.2	Ausgewählte Lokisierungsalgorithmen . . . . .	84
4.3.3	Methodik . . . . .	88
4.3.4	Innerhalb der konvexen Hülle . . . . .	89
4.3.5	Außerhalb der konvexen Hülle . . . . .	90
4.3.6	Zusammenfassung . . . . .	92
4.4	Vergleich mit Simulationsergebnissen . . . . .	94
4.4.1	Motivation und Methodik . . . . .	94
4.4.2	Raumszenario . . . . .	95
4.4.3	Gebäudeszenario . . . . .	99
4.4.4	Zusammenfassung . . . . .	101
4.5	Zusammenfassung . . . . .	101
<b>5</b>	<b>Erzeugung von Routing Graphen</b>	<b>103</b>
5.1	Motivation . . . . .	103
5.2	Allgemeine Anforderungen . . . . .	105
5.3	Conditional Erosion Algorithm (CEA) . . . . .	107
5.3.1	Iterative Erosion freien Raumes . . . . .	108
5.3.2	Formulierung als Gleichungssystem . . . . .	111
5.3.3	Weitere Reduzierung der Kernel . . . . .	113
5.3.4	Liste weißer Pixel . . . . .	115
5.3.5	Implementierung und vollständiger Algorithmus . . . . .	117
5.4	Ergebnisse und Diskussion . . . . .	120
5.5	Zusammenfassung . . . . .	128
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>130</b>
	<b>Literaturverzeichnis</b>	<b>132</b>
	<b>Zusammenfassung</b>	<b>151</b>
	<b>Abstract</b>	<b>153</b>

---

# Abbildungsverzeichnis

2.1	TurtleBot: Roomba 531 mit einer montierten Microsoft Kinect Tiefenbildkamera. Zur Lokalisierung ist zusätzlich ein handelsüblicher Laptop nötig. . . . .	38
2.2	TurtleBot in einem Flur. Die Tiefeninformationen der Kinect (grün) werden mit bekannten Hindernissen in der Karte abgeglichen. Eine Menge an möglichen und wahrscheinlichen Positionen ist durch die violetten Pfeile gegeben. Die Einbuchtung im unteren Teil der Karte stellt eine (als geschlossen eingetragene) Tür dar. . . . .	40
2.3	Die kumulative Fehlerverteilung des Referenzsystem im getesteten Flur. Die x-Achse zeigt den Positionsfehler in Metern. . . . .	42
2.4	Positionsschätzungen im 1. OG des Freie Universität Berlin (FU)-Gebäudes. Die Startposition des gefahrenen Pfades ist mit $S$ , die Endposition mit $E$ gekennzeichnet. . . . .	43
3.1	Komponenten und Informationsfluss des virtuellen Testbeds. . . . .	50
3.2	Datenbankrelation des virtuellen Testbeds. Übernommen aus der Dissertation von Hillebrandt und Will [4]. . . . .	53
3.3	Durchschnittlicher räumlicher Distanzfehler des System Under Test (SUT) (über alle aktivierten Anker). Die Zellgröße beträgt 0,1 m. . . . .	54
3.4	Eine Personal Tracking and Transmission Unit (PTTU) mit Gehäuse und angeschlossener GPS-Antenne. . . . .	57
3.5	Seminarraum in der ersten Etage des FU-Gebäudes mit installierten Ankerpositionen (blau). . . . .	58
3.6	Zweite Etage des FU-Gebäudes mit installierten Ankerpositionen (grün). . . . .	59
3.7	Dichte der gesammelten Daten. Die Zellgröße beträgt 0,3 m. . . . .	60
3.8	Räumlicher Distanzfehler des SUT. Die Zellgröße beträgt 0,3 m. . . . .	61

3.9	Datenaufzeichnungen und extrahierter virtueller Lauf der Machbarkeitsstudie. Die Einheit der Achsen ist Meter. . . . .	62
3.10	Fehlerverteilungen des realen und virtuellen Laufs. . . . .	64
3.11	Durchschnittlicher Fehler pro Anker. . . . .	64
3.12	Reale Büroumgebung mit realem Pfad. . . . .	66
3.13	Fehler pro Anker in der realen Büroumgebung. . . . .	67
4.1	Räumliche Fehlerverteilung verschiedener Ankerplatzierungen mit wenig Non Line of Sight (NLOS)-Anteil. . . . .	76
4.2	Erfolgreiche Distanzmessungen zu Anker 3. . . . .	78
4.3	Räumliche Fehlerverteilung verschiedener Ankerplatzierungen mit hohem oder unerwartetem NLOS-Anteil. . . . .	80
4.4	Räumliche Fehlerverteilung verschiedener Lokalisierungsalgorithmen im ersten Szenario. Für einen guten Vergleich der besseren Algorithmen sind höhere Fehler (bspw. Abb. 4.4a) farblich nicht mehr unterscheidbar. . . . .	91
4.5	Räumliche Fehlerverteilung verschiedener Lokalisierungsalgorithmen im zweiten Szenario. Für einen guten Vergleich der besseren Algorithmen sind höhere Fehler (bspw. Abb. 4.4a) farblich nicht mehr unterscheidbar. . . . .	93
4.6	Durch Simulation erwartete räumliche Positionsfehlerverteilung. Fehler in [dm]. . . . .	96
4.7	Räumliche Fehlerverteilung im Raumszenario für einen Anker (unten, mitte). Ein Pixel entspricht $5\text{ cm} \times 5\text{ cm}$ . . . . .	97
4.8	Räumliche Fehlerverteilungen im Raumszenario. Ein Pixel entspricht $15\text{ cm} \times 15\text{ cm}$ . . . . .	98
4.9	Durch Simulation erwartete räumliche Positionsfehlerverteilung für verschiedene Lokalisierungsalgorithmen. Fehler in [dm]. . . . .	100
5.1	Ausgewählte Kernel (Beispiele). . . . .	109
5.2	Beispielkernel, welche mindestens ein Kriterium verletzen. Kernel 5.2a verletzt Krit. 1), 5.2b verletzt Krit. 2), Kernel 5.2c verletzt Krit. 3) und Kernel 5.2d verletzt Krit. 4). . . . .	109
5.3	Repräsentanten mit drei Zuständen: weiß, schwarz und irrelevant (grau). Gruppen 5.3a bis 5.3c erlauben jeweils vier Rotationen. Gruppen 5.3d und 5.3e erlauben zusätzlich Reflexionen, folglich acht Möglichkeiten. Die Gruppe 5.3f besteht nur aus einem Kernel, welcher nicht rotiert oder gespiegelt werden kann. . . . .	110

---

5.4	Repräsentanten der vereinfachten Gruppen mit drei Zuständen: weiß, schwarz und irrelevant (grau). Gruppen 5.4a und 5.4c erlauben jeweils vier Rotationen, Gruppe 5.4b erlaubt zusätzlich Reflexionen, also acht Möglichkeiten. . . . .	112
5.5	Ausschnitt aus einem erodierten Gebäudeplan. Der blaue Graph zeigt die Erosion mit 17 Kernen. In rot sind die zusätzlich übrig bleibenden Pixel bei der Erosion mit nur acht Kernen dargestellt. . . . .	114
5.6	Relative Häufigkeit der Treffer pro Kernel. Die Nummerierung entspricht der Reihenfolge in den Formeln 5.9 und 5.10. . . . .	114
5.7	Weiter reduzierte Kernmenge, welche nur zwei Rotationen erlaubt. . .	115
5.8	Gebäudepläne mit resultierendem Routing-Graphen. . . . .	121
5.9	Ein erodierter Gebäudeplan. Der Routing-Graph (blau) wurde mit acht Threads parallel berechnet. Die Grenzen zwischen den Threads sind in rot angegeben. . . . .	124
5.10	Durchschnittliche relative Häufigkeit von Schwärzungen beim ersten, zweiten usw. Versuch die Kernel auf einen bestimmten Pixel anzuwenden. Wurde ein Pixel bei der ersten (zweiten, dritten, . . .) Überprüfung geschwärzt, war nur eine (zwei, drei, . . .) Kernelanwendung nötig. . .	125
5.11	Ein erodierter Gebäudeplan (blau) bei Anwendung von nur vier Kernen. Wände wachsen hier von rechts unten nach links oben. Die verwendeten Kernel sind in Abb. 5.7 gezeigt. . . . .	126
5.12	Erodierter Gebäudeplan. Der rote Graph repräsentiert das Ergebnis ohne rotierte Karte, also in intuitiver Orientierung wie hier abgebildet. Der blaue Graph zeigt das Ergebnis, wenn die selbe Karte um $45^\circ$ im Uhrzeigersinn gedreht dem Algorithmus als Eingabe übergeben wird. Das Ergebnis wurde für eine sinnhafte Präsentation zurückgedreht. . .	127
5.13	Erodierter Gebäudeplan. Der Graph zeigt das Ergebnis, wenn die Karte um $45^\circ$ im Uhrzeigersinn gedreht dem Algorithmus als Eingabe übergeben wird. Das Ergebnis wurde für eine sinnhafte Präsentation zurückgedreht. . . . .	127





---

# Tabellenverzeichnis

2.1	Systemkategorie [Anzahl (%)] . . . . .	18
2.2	Evaluationskategorie [Anzahl (%)] . . . . .	18
2.3	Evaluationskategorie (nur <i>Inertial</i> ) [Anzahl (%)] . . . . .	19
2.4	Evaluationskategorie (außer <i>Inertial</i> ) [Anzahl (%)] . . . . .	19
2.5	Referenzkategorien [Anzahl (%)] . . . . .	20
2.6	Metrikkategorien [Anzahl (%)] . . . . .	22
2.7	Metrikkategorien (nur <i>Inertial</i> ) [Anzahl (%)] . . . . .	23
2.8	Metrikkategorien (nur <i>Map-Matching</i> ) [Anzahl (%)] . . . . .	24
2.9	Metrikkategorien (nur <i>RSS</i> ) [Anzahl (%)] . . . . .	25
2.10	Metrikkategorien (nur <i>TOF</i> ) [Anzahl (%)] . . . . .	25
2.11	Metrikkategorien (nur <i>Simple-/Complex-Simulation</i> ) [Anzahl (%)] . . . . .	26
2.12	Metrikkategorien (außer <i>No-Evaluation</i> ) [Anzahl (%)] . . . . .	27
2.13	Metrikkategorien (nur <i>Discrete-Point</i> ) [Anzahl (%)] . . . . .	27
2.14	Metrikkategorien (nur <i>Office-Walk</i> ) [Anzahl (%)] . . . . .	28
2.15	Metrikkategorien (nur <i>Multiple</i> ) [Anzahl (%)] . . . . .	28
2.16	Baselinekategorie [Anzahl (%)] . . . . .	29
3.1	Parameter des realen und virtuellen Laufs . . . . .	63
3.2	Resultate verschiedener Lokalisierungsalgorithmen . . . . .	65
3.3	Parameter des realen und virtuellen Laufs . . . . .	66
4.1	Lokalisierungsfehler ausgewählter Algorithmen in beiden Szenarien . . . . .	92
4.2	Durchschnittliche Positionsfehler im Raumszenario . . . . .	98
5.1	CEA-Laufzeiten in [ms] . . . . .	123



---

# Algorithmenverzeichnis

1	Einfache Version des Conditional Erosion Algorithm (CEA). . . . .	110
2	Vollständiger Conditional Erosion Algorithm (CEA). . . . .	118
3	Parallelisierung der Erosion. . . . .	120

---

# Abkürzungsverzeichnis

<b>AMCL</b>	Adaptive Monte Carlo Localization
<b>AML</b>	Adapted Multi-Lateration
<b>AoA</b>	Angle of Arrival
<b>CAD</b>	Computer-Aided Design
<b>CDF</b>	Cumulative Distribution Function
<b>CSS</b>	Chirp Spread Spectrum
<b>DNF</b>	disjunktive Normalform
<b>DoA</b>	Direction of Arrival
<b>FM</b>	Frequenzmodulation
<b>FU</b>	Freie Universität Berlin
<b>GNSS</b>	Global Navigation Satellite System
<b>GPS</b>	Global Positioning System
<b>GSM</b>	Global System for Mobile Communications
<b>HCU</b>	HafenCity Universität
<b>HTTP</b>	Hypertext Transfer Protocol
<b>ICLA</b>	Iterative Clustering-based Localization Algorithm
<b>IMU</b>	Inertial Measurement Unit
<b>INS</b>	Inertial Navigation System
<b>IPIN</b>	International Conference on Indoor Positioning and Indoor Navigation
<b>ITU</b>	International Telecommunications Union

---

<b>KS-Test</b>	Kolmogorov-Smirnov-Test
<b>LLS</b>	Linear Least Squares
<b>LOS</b>	Line of Sight
<b>MAE</b>	Mean Absolute Error
<b>MSE</b>	Mean Squared Error
<b>NetCDF</b>	Network Common Data Form
<b>NLLS</b>	Non-linear Least Squares
<b>NLOS</b>	Non Line of Sight
<b>PDR</b>	Pedestrian Dead Reckoning
<b>PoA</b>	Phase of Arrival
<b>PTTU</b>	Personal Tracking and Transmission Unit
<b>REST</b>	Representational State Transfer
<b>RF</b>	Radio Frequency
<b>RFID</b>	Radio Frequency Identification
<b>RLSM</b>	Robust Least-Squared Multilateration
<b>RMSE</b>	Root Mean Squared Error
<b>ROS</b>	Robot Operating System
<b>RS</b>	Referenzsystem
<b>RSS</b>	Received Signal Strength
<b>RSSI</b>	Received Signal Strength Indicator
<b>RTof</b>	Roundtrip Time of Flight
<b>RTT</b>	Round Trip Time
<b>SDS-TWR</b>	Symmetrical Double-Sided Two Way Ranging
<b>SNR</b>	Signal to Noise Ratio
<b>SUT</b>	System Under Test
<b>TDoA</b>	Time Difference of Arrival
<b>ToA</b>	Time of Arrival

<b>ToF</b> . . . . .	Time of Flight
<b>TWIST</b> . . . . .	TKN Wireless Indoor Sensor Network Testbed
<b>UWB</b> . . . . .	Ultra Wide Band
<b>VirTIL</b> . . . . .	Virtual Testbed for Indoor Localization
<b>WLAN</b> . . . . .	Wireless Local Area Network



---

# KAPITEL 1

---

## Einleitung

### 1.1 Motivation

Die Bestimmung des eigenen Ortes ist ein tagtäglicher, meist vollkommen unbewusster Prozess eines jeden Menschen. Auch die bewusste absolute oder relative Ortsbestimmung anhand von Informations- und Leitsystemen, Schildern und Pfeilen, analogen oder digitalen Grundrissplänen, Übersichtskarten und durch satellitengestützte Systeme ist allgegenwärtig. Aktuell entstehen neue Märkte zur Lokalisierung von Personen, Waren, Maschinen, Werkzeugen usw. innerhalb von Gebäuden. Die grundlegenden Methoden und Techniken existieren seit langem und werden zum Beispiel von Navigationsgeräten im Auto durch Funkdistanzmessungen und inertialbasierte Systeme eingesetzt. Da Signale von Satellitensystemen innerhalb von Gebäudestrukturen physikalisch bedingt nicht ausreichend empfangen werden können und derartige Systeme mit nicht zufriedenstellenden Positionsfehlern von mehreren Metern aufwarten, bedarf es hier anderer Lösungen. Das Anbringen spezieller oder das Nutzen vorhandener Funksender (bspw. Wireless Local Area Network (WLAN)-Router) innerhalb von Gebäuden ist nur in bestimmten Fällen ausreichend. Erfahrungsgemäß ist das der Fall, wenn genügend Ressourcen wie finanzielle Mittel, Zeit für Einmessung und Wartung sowie Redundanz eingesetzt werden. Die Gründe sind oft technischer Natur: Dazu zählen ungenaue Messmethoden, die Anwendung schlechter oder unpassender Algorithmen, der Einfluss unterschiedlicher Materialzusammensetzung der Wände, Gegenstände oder des Mediums auf die Signalausbreitung oder eine nicht optimale Installation, bspw. wenn Reflexionseigenschaften von Wänden vernachlässigt werden.



Dies bedeutet, dass teilweise stark fehlerbehaftete Werte zur Positionsermittlung zur Verfügung stehen. Zwar existieren Methoden, die genau jene oft charakteristischen Eigenschaften zur Schätzung hinzuziehen, da bspw. das Erdmagnetfeld charakteristisch verzerrt ist. Allerdings bedarf es dann einer stetigen Aktualisierung von Datenbanken, die genau jene Charakteristiken speichern. Ein hoch gestecktes Ziel der Wissenschaft ist es daher, eine Positionsermittlung unabhängig von vorhandener Infrastruktur, unabhängig der konkreten Zusammensetzung von Wänden etc. zu erforschen.

Es existieren einige Arbeiten, die das Forschungsfeld der Lokalisierung unter technischen und praktischen Aspekten sehr gut beschreiben, wie bspw. die Bücher von Zekavat und Buehrer [1], Bensky [2], die Habilitationsschrift von Mautz [3] und die Dissertation von Hillebrandt und Will speziell für distanzbasierte Lokalisierung [4], daher wird in dieser Arbeit auf eine ausführliche Darstellung der verfügbaren Techniken verzichtet. Neben dem Schätzen von ungefähren Aufenthaltsbereichen (bspw. Gebäudeteil, Raum etc.) [5–7] sieht man gegenwärtig hauptsächlich drei oft miteinander kombinierte Strategien, das Lokalisierungsproblem für konkrete Positionen zu lösen – Fingerprinting, Inertial Navigation Systeme (INS) und Multilateration:

**Fingerprinting** Erstens können aktiv oder passiv Umgebungsparameter gemessen und diese zur Ortsbestimmung ggf. gegen eine bekannte Datenbasis abgeglichen werden. Es wird hierbei eine lediglich geringe Änderung der Parameter über die Zeit angenommen. Hierzu gehören Systeme, die Received Signal Strength (RSS) [8–12], Ultraschall [13], Licht [14], magnetische Felder [15–20], Frequenzmodulation (FM) [8, 9, 21–29], Global System for Mobile Communications (GSM) [24, 30, 31] oder WLAN [10–12, 18, 19, 21, 25, 26, 28, 29, 32–34] als Fingerprint der Umgebung nutzen. Hier kann eine zusätzliche Infrastruktur nötig werden oder bestehende Systeme, wie bspw. WLAN-Router, können genutzt werden.

**INS** Zweitens kann die aktuelle Position durch ein INS mit Beschleunigungssensoren, Gyroskopen und Barometern kontinuierlich verfolgt werden [19, 20, 32, 35–41]. Dieser Ansatz der Koppelnavigation bedarf keiner externen Hardware und ist damit überall kostengünstig zu implementieren, allerdings werden gut kalibrierte Sensoren benötigt. Und auch dann existiert ein systemimmanenter Drift, welcher nur durch zusätzliche Informationsquellen rückgesetzt werden kann.

**Multilateration** Zur dritten Kategorie zählen Systeme, welche die Position über Distanzen zu bekannten Orten/Geräten in der Umgebung bestimmen. Hier muss

derzeit meist ein spezielles Sensorsystem (sog. Anker) installiert und ggf. kalibriert werden. Es kann die Signalstärke (Received Signal Strength Indicator (RSSI)), die Signallaufzeit (Time of Flight (ToF) bzw. Time of Arrival (ToA) oder Time Difference of Arrival (TDoA), sowie Roundtrip Time of Flight (RTof)) oder die Signalphase (Phase of Arrival (PoA)) gemessen werden. Dazu gehören Systeme, die Ultraschall [42, 43], Radio Frequency (RF)/RSS [4, 40–48], Licht [14, 49] oder Radio Frequency Identification (RFID) [7] nutzen.

Eine der Multilateration verwandte Art der Ortsbestimmung ist die Multiangulation. Bekannt sind diese Ansätze unter den Namen Angle of Arrival (AoA) oder auch Direction of Arrival (DoA). Dabei werden Winkel zu mehreren bekannten Orten mithilfe von Sätzen aus der Trigonometrie zur Schätzung genutzt [50–55]. Andere Verfahren nutzen bspw. Radiotomographie [56–59]. Hier werden Transceiver an den Rändern eines Einsatzgebietes platziert und Signalveränderungen zwischen ebendiesen paarweise ausgewertet. Eine weitere Möglichkeit sind nachbarschaftsbasierte Ansätze [60–62]. Hier wird lediglich die Nähe (bspw. anhand einer Funksignals oder anderen messbaren Parametern) als Positionsmetrik genutzt.

In der Literatur findet sich eine Unterscheidung zwischen infrastrukturbasierten und infrastrukturlosen Ansätzen, obwohl eine Definition nicht immer direkt gegeben ist [63]. Während rein inertialbasierte Systeme klar als infrastrukturlos bezeichnet werden, betrachtet man Systeme, die bereits vorhandene Infrastruktur verwenden, ebenso als infrastrukturlos, und Systeme, welche neuartige Geräte in der Umgebung installieren müssen, als infrastrukturbasiert. Beispielsweise im Kontext von Katastrophen- und Rettungseinsätzen kann sich nicht zwangsläufig auf vorhandene Strukturen bzw. eine Stromversorgung verlassen werden [41, 64], wodurch sich diese Definition ändern muss. In oben vorgestellter Literatur findet man neben den inertialbasierten Ansätzen ein System von Tarzia et al., welches das akustische Hintergrundrauschen analysiert und mit einer zuvor erstellten Datenbank abgleicht [13]. Ein weiteres System von Chung et al. gleicht Messungen des natürlichen Magnetfelds, welches durch Eisenelemente in Gebäuden charakteristisch verzerrt wird, mit einer ebenfalls zuvor erstellten Datenbank ab [17]. Beide Vertreter benötigen damit zwar keine Infrastruktur per se, sind jedoch für genannte Szenarien nicht einsetzbar, da sich aufgezeichnete Charakteristiken nicht mehr unbedingt wiederfinden. Eine ganzheitliche infrastrukturlose Lösung ohne Erstellung eines Modells des Gebäudes ist daher nicht absehbar.

Eine genaue Ortsbestimmung innerhalb von Gebäuden hat das Potential, die Art

und Weise von Navigation in Gebäuden ähnlich der Einführung des Global Positioning Systems (GPS) zu verändern. Daher werden aktuell diverse Wettbewerbe zur Lokalisierung in Gebäuden im internationalen Forschungsumfeld durchgeführt, welche alle drei o.g. Lokalisierungsstrategien abdecken. Die Erfahrungen aus dem Lokalisierungswettbewerb der *International Conference on Information Processing in Sensor Networks (IPSN)* im Jahr 2014 mit 22 teilnehmenden internationalen Teams lassen die noch heute gültigen Schlüsse zu [63]:

**Genauigkeit** Gegenwärtige Strategien zur Lokalisierung von Menschen oder Objekten in Gebäuden erlauben eine Genauigkeit bis hin zu 0,72m. Applikationen, welche eine raum- oder metergenaue Ortung benötigen, wie bspw. die Navigation von Menschen, sind damit leicht umsetzbar. Das Lokalisierungsproblem ist damit jedoch noch nicht gelöst. Eine Reihe an weiteren Anwendungen, wie bspw. personalisierte Preise, Produktwerbungen in Geschäften oder ggf. auch die Interaktion mit virtuellen Objekten bei Nutzung von 3D-Brillen, benötigen eine weitaus höhere Genauigkeit, um bspw. im Industriekontext einen signifikanten Mehrwert zu generieren.

**Installation** Beim Wettbewerb benötigten die Teilnehmer für ca. 300 m<sup>2</sup> durchschnittlich sieben Stunden, um ihr System zu installieren und zu kalibrieren. Es ist anzunehmen, dass die Teilnehmer die gegebene Zeit möglichst bis zum Ende sinnvoll nutzen wollten. Dennoch charakterisiert dies den enormen zeitlichen und personellen Aufwand, der für eine solche Fläche nötig sein kann. Es wurde kein Wert auf eine Integration in die bestehende Umgebung (bspw. ein Verstecken der Geräte oder deren Design) gelegt. Auch war der Prozess recht aufdringlich; es wurde keine Rücksicht auf andere Parameter genommen. So ist es unvorstellbar, dass ein solcher Installationsaufwand in einem geöffneten Kaufhaus durchgeführt werden könnte. Eine schnelle Installation wird damit zum Erfolgsfaktor für ein Lokalisierungssystem.

**Umgebungsveränderungen** Einige Studien zeigen den Einfluss des Menschen und der Einrichtung auf den Lokalisierungsprozess. Allerdings werden die meisten Labortests in kleinen, statischen Umgebungen durchgeführt. Auch werden weitere Umgebungsbedingungen wie Temperatur, Zusammensetzung des Mediums, temporäre Änderungen des Kanals etc. unzureichend untersucht. Eine Evaluation sollte unter sich verändernden Bedingungen mehrmals wiederholt werden.

**Evaluation** Weiterhin scheint keine Technologie zu existieren, welche über alle Eva-

luationspunkte hinweg einen ähnlichen Lokalisierungsfehler bietet. Alle Systeme zeigen große Veränderungen in der Genauigkeit über die Evaluationspunkte hinweg. Um eine allgemeingültige Evaluation durchzuführen, müsste das System in breiter Fläche evaluiert werden, um die Stabilität und damit Verlässlichkeit der Technologie zu bewerten. Eine manuelle Evaluation, wie sie bisher größtenteils durchgeführt wird, scheint hier zu ermüdend und zeitraubend; eine automatische Evaluation der Systeme mit Robotern ist hier vielversprechender und erlaubt eine Skalierung der Dichte der Messpunkte. Auch können so kontinuierliche Pfade untersucht werden, was tatsächliche Anwendungen eher widerspiegelt.

**Bewertung** Die Effektivität einer Lösung zu bewerten, ist nicht trivial. Idealerweise sollten Technologien nach vielen Genauigkeitsmetriken und auch in weiteren Aspekten, wie bspw. der Installationszeit, untersucht werden.

Trotz oder gerade wegen der Vielzahl an vorhandenen Möglichkeiten, deren spezifischen Vor- und Nachteilen sowie unterschiedlicher Evaluationsmethoden hat sich bisher keine dominante Methode oder generische Architektur gezeigt [63]. Unter o.g. gutartigen Bedingungen lässt sich das Lokalisierungsproblem in Gebäuden als weitgehend gelöst betrachten, sofern Durchschnittsfehler von wenigen Dezimetern für die Applikation ausreichen. Sobald eine dieser notwendigen Aspekte nicht erfüllt werden kann, bspw. beim Lokalisieren von Rettungskräften, leidet das Ergebnis teilweise erheblich und bedarf weiterer Forschungs- und Entwicklungsarbeit.

## 1.2 Wissenschaftlicher Beitrag

Diese Dissertation trägt in obigem Kontext zu den folgenden Thematiken bei. Die wesentlichen wissenschaftlichen Neuheiten werden kurz vorgestellt.

### Evaluation von Lokalisierungssystemen

Als Motivation für darauf folgende Arbeiten diskutiert Kapitel 2 die Dokumentationsstiefe der Evaluation von Veröffentlichungen der *IEEE Intl. Conf. on Indoor Positioning and Indoor Navigation (IPIN)* aus den Jahren 2010 bis 2014 [65]. Der Mangel an gut dokumentierten Evaluationen motiviert die Arbeit an einem günstigen, effektiven Referenzsystem zur Evaluation von Lokalisierungstechnologien [66]. Diese fokussiert weitgehend die Evaluation von Systemen denen Distanzmessungen zu be-

kannten Orten zu Grunde liegen. Eingeführte Konzepte können jedoch teilweise auf weitere Technologien ausgeweitet werden.

*Wissenschaftliche Neuheit:* Roboter als Referenzsystem

- Aufarbeitung der Evaluations-/Dokumentationstiefe in der Wissenschaft
- Einführung eines günstigen Referenzsystems zur Sammlung von Groundtruth-Daten (wirkliche Position in Raum und Zeit)
- Ermöglichung numerischer Vergleiche zur Evaluation von Lokalisierungssystemen

Virtuelles Testbed für distanzbasierte Lokalisierung

In Anlehnung an Erfahrungen zu benannten Wettbewerben und deren Schlüsse präsentiert und evaluiert Kapitel 3 eine neuartige Form von Testbeds. Zur Vermeidung langwieriger und kostenintensiver Experimente hält eine Datenbank genügend unabhängige funkbasierte Messungen vor, sodass virtuelle Läufe mit beliebigen Bewegungsmustern durch Echtwelt-Daten generiert werden können [67, 68]. Außerdem werden Unterschiede zwischen Testumgebungen und Simulationen gleicher Szenarien aufgezeigt [69]. Auch hier lassen sich eingeführte Konzepte meist leicht auf weitere Technologien erweitern. Als wichtiger Vertreter von Lokalisierungstechnologien können INS dagegen prinzipbedingt nicht ohne Weiteres in einem solchen Testbed repräsentiert werden.

*Wissenschaftliche Neuheit:* Virtual Testbed for Indoor Localization (VirTIL)

- Automatische, engmaschige Flächenabdeckung durch mobile Roboter
- Sammlung versch. Daten, bspw. Distanzen, aus realen Experimenten
- Definition eines generischen Speicherformats für Lokalisierungsdaten
- Simple Extraktion von beliebigen Bewegungsmustern oder Teilbereichen
- Erlaubt die Evaluation eines Lokalisierungssystems in spezifischen Szenarien

## Räumliche Fehlerverteilung bei distanzbasierter Lokalisierung

Kapitel 4 stellt Beiträge zur räumlichen Distanzmess- und Positionsfehleranalyse eines funkbasierten Systems vor, welche als wertvolle Vergleiche mehrerer Lokalisierungsalgorithmen unter Verwendung authentischer Daten dienen [69–71].

*Wissenschaftliche Neuheit:* Räumlicher Einblick in physikalische Effekte und deren Auswirkungen auf die Lokalisierung

- Vermeidung nachweislich nicht allgemeingültiger Aussagen nur anhand von Simulationen
- Räumliche Distanz- und Positionsfehleranalyse
- Untersuchung neuer, zuvor unbekannter und/oder schwer nachweisbarer physikalischer Effekte

## Erzeugung von Routing Graphen

Inertialbasierte Lokalisierungssysteme leiden unter einem systemimmanenten Drift. Eine Lösung dazu sind sog. Routing-Graphen, anhand welcher bspw. Inertialdaten abgeglichen bzw. zurückgesetzt werden. Lösungen derartiger Graphen werden in Kapitel 5 vorgestellt. Die Extraktion aus Gebäudeplänen in Form von Rastergrafiken erfolgt durch eine bedingte Dilatation von Wänden in Anlehnung an Bildverarbeitungsmechanismen [72, 73]. Auch lässt sich ein solcher Graph zukünftig für einzelne Funkdistanzmessungen verwenden. Dies ist insbesondere in Katastropheneinsätzen von Interesse, da aufgrund von Umgebungsbedingungen nicht immer genügend Signale zur Multilateration empfangen werden können.

*Wissenschaftliche Neuheit:* Graphenextraktion

- Einführung einer automatischen Methode zur Graphenextraktion
- Ähnliche Algorithmen (bspw. Labyrinthsegmentierung) zeigen eine bedeutend schlechtere Laufzeit

## Über diese Arbeit

Die Arbeit geht kapitelweise auf verschiedene Aspekte der Lokalisierung ein. Um dem Leser weite Lesesprünge zu ersparen, werden Hintergründe zu Kapiteln in ebendiesen dargelegt. Daher werden alle Kapitel in weitgehend abgeschlossener Form präsentiert und relevante eigene Publikationen zum Thema zu Beginn jedes Kapitels aufgeführt.

Alle Publikationen wurden gemeinsam mit anderen Autoren erstellt. Die Reihenfolge der Autorennennung bei den als *eigene Arbeiten* deklarierten Veröffentlichungen (zu Beginn der Kapitel) richtet sich dabei ausschließlich nach praktischen Erwägungen. Es lässt sich keine Haupt- oder Nebenautorenschaft daraus ableiten. So wird bei Konferenzbeiträgen immer der Autor als erstes genannt, der den Beitrag auf der Konferenz vorstellte. Dies hat sich anhand praktischer Abwägungen, wie bspw. einer zeitlichen Verfügbarkeit, ergeben. Das bedeutet, dass die zugrundeliegenden Publikationen von den jeweiligen Autoren gemeinsam und gleichwertig erstellt wurden.

Dieser Arbeit liegen Forschungsbeiträge aus den Jahren 2012 bis 2017 zugrunde, die allesamt einem Peer-Review-Verfahren unterzogen wurden und auf der entsprechenden Konferenz vorgestellt sowie im entsprechenden Druckwerk veröffentlicht wurden.

---

## KAPITEL 2

---

# Evaluation von Lokalisierungssystemen

Experimente stellen in der Forschung zur Lokalisierung von Menschen oder Objekten ein wichtiges Instrument zur Evaluation neuer Methoden, Systeme und Algorithmen dar. Es kann zwischen simulierten und sog. Real-Welt-Experimenten (auch Feldtests genannt) unterschieden werden. Während erstere leicht zu entwickeln und bewusst oder unbewusst auf das zu Untersuchende zugeschnitten sein können, erlauben Real-Welt-Experimente in der Lokalisierung meist bessere Erkenntnisse in Bezug auf die tatsächliche Leistung. Das wesentliche Problem hinter Real-Welt-Experimenten ist der Bedarf an Hardware, dem vom zu untersuchenden Lokalisierungsalgorithmus unabhängigen Wissen über die Hardware oder schlicht finanzieller Mittel, ein solches zu realisieren. Zudem ist ein Referenzsystem (RS) nötig, mit welchem die wirkliche Position des SUT im Raum zu einer bestimmten Zeit (im folgenden als Groundtruth bezeichnet) messbar wird. Für einen messbaren wissenschaftlichen Fortschritt ist es daher unerlässlich, eine Vergleichbarkeit neuer Lösungen mit bisherigen Alternativen herzustellen. Dieses Kapitel analysiert dazu zunächst die Dokumentation experimenteller Evaluation auf einer einschlägigen Konferenz. Anschließend wird ein neuartiges RS vorgestellt, mit welchem einfach und günstig Real-Welt-Experimente durchgeführt, und somit Lokalisierungslösungen evaluiert werden können.

### 2.1 Experimentelle Evaluation in der Wissenschaft

Dieses Kapitel basiert auf folgender eigener Veröffentlichung:

- S. Adler, S. Schmitt, K. Wolter und M. Kyas, “A survey of experimental eva-



luation in indoor localization research”, in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Okt. 2015. DOI: 10.1109/IPIN.2015.7346749

### 2.1.1 Motivation

Die systematische Aufbereitung von Evaluationen gängiger Systeme in wissenschaftlichen Veröffentlichungen erlaubt eine Untersuchung des gegenwärtigen Standes der Wissenschaften im Bereich der Lokalisierung in Gebäuden. Die Motivation zu dieser Untersuchung liegt in einer Teilnahme an der fünften International Conference on Indoor Positioning and Indoor Navigation (IPIN). Diese vermittelte den Eindruck über eine sichtbare Verbesserung der Methodik bei der Evaluation von Lokalisierungssystemen in den letzten Jahren. Folgende Untersuchung soll zeigen, ob der Eindruck durch eine systematische Analyse bisheriger Veröffentlichungen auf dieser Konferenz bestätigt werden kann. Dazu werden die Evaluationsmethoden und deren Vergleichbarkeit der ersten fünf Konferenzen dieser Reihe analysiert [74–78]. Die Untersuchung soll zeigen, ob es eine technologische Tendenz gibt, wie Technologien evaluiert werden und ob die Veröffentlichungen auf wissenschaftlichen, empirisch adäquaten Methoden beruhen. Als Fazit werden methodische Verbesserungen bei der Evaluation vorgeschlagen.

Studien deuten darauf hin, dass sich der Mangel an empirischen oder experimentellen Anteilen in Veröffentlichungen über Jahre hinweg nicht signifikant veränderte. Tichy et al. [79] überprüfen Veröffentlichungen in der Informatik im Jahr 1993 daraufhin, ob Wissenschaftler ihre Ergebnisse durch experimentelle Evaluation untermauern. Die Ergebnisse zeigen eine unbefriedigende Rate: 40% ihrer Probe führten gar keine Evaluation durch, auch wenn es nach Tichy et al. nötig gewesen wäre. Die Autoren motivieren für mehr Experimente in allen Bereichen der Informatik [80]. Die Ergebnisse von Tichy et al. wurden später durch Wainer et al. für Publikationen im Jahr 2005 bestätigt [81, 82]. Auch neuere Studien unterstreichen die Relevanz von Experimenten und deren Mangel in verschiedenen Disziplinen der Informatik, wie bspw. Andujar et al. [83] sowie Tedre und Moisseinen [84].

Dieses Kapitel ist auch ein Beitrag zur Debatte über Reproduzierbarkeit von Ergebnissen in diesem Bereich der Wissenschaft. Reproduzierbarkeit ist ein Eckpfeiler jeder Wissenschaft, insbesondere in der Physik. Eine Theorie sollte durch die Wis-

senschaftsgemeinde und schließlich durch die Gesellschaft akzeptiert werden, genau dann wenn sie durch unabhängige Untersuchungen unter gleichen oder vergleichbaren Rahmenbedingungen, bspw. durch Reproduktion der Ergebnisse, unabhängig bestätigt wurde [85]. Diese Untersuchung schätzt jedoch nicht die Reproduzierbarkeit der Ergebnisse und wiederholt auch kein Experiment. Auch wurde kein Kontakt zu den Autoren hergestellt, um bspw. Code oder zusätzliche Informationen zu erhalten. Daher kann keine Aussage zur Reproduzierbarkeit einer einzelnen Veröffentlichung gemacht werden. Ein sehr gutes Beispiel zur Herstellung weitgehender Nachvollziehbarkeit und damit Reproduzierbarkeit zeigen die Autoren Buckheit und Donoho [86] sowie später Collberg et al. [87].

Da experimentelle Daten statistisch analysiert werden, sollten die Vorschläge von Gentleman und Lang gelten [88]. Eine Publikation sollte zumindest die Daten und die Methoden referenzieren, die zur Erstellung aller statistischer Daten genutzt wurden, wie Zahlen und Abbildungen. Für Simulationen bedeutet dies auch die Referenzierung des Codes, der Einstellungen und Parameterbelegungen.

Physikalische Experimente sind im Allgemeinen schwieriger zu reproduzieren. Nach Collberg et al. müssten alle Veröffentlichungen ausgeschlossen werden, welche spezielle Hardware nutzen [87], was im Untersuchungsbereich nahezu allen Veröffentlichungen entspricht. Jedoch sollten auch diese in größtmöglichem Detail erklärt werden, sodass andere mit Zugriff auf diese oder vergleichbare Hardware die Experimente wiederholen können.

Jedoch kann durchaus eine Aussage über Wiederholung innerhalb von Experimenten gemacht werden. Dazu werden die Veröffentlichungen gezählt, welche über die Wiederholung ihrer Experimente berichten und konkrete Zahlen nennen. Ein Experiment zu wiederholen, erhöht das Vertrauen in die Beobachtung und verbessert die Schärfe der Messungen. Letztere ist proportional zur reziproken Quadratwurzel der Anzahl an Wiederholungen. Wiederholbarkeit sollte die folgenden Bedingungen erfüllen: Das Experiment sollte mit den selben experimentellen Mitteln, vom selben Beobachter mit den selben Messinstrumenten unter den selben oder nahezu gleichen Bedingungen (siehe unkontrollierbare Veränderungen, bspw. verstrichene Zeit) am selben Ort durchgeführt werden. Das Experiment sollte über einen kurzen Zeitraum mit dem selben Ziel wiederholt werden [89–91].

Zur Fokussierung auf eine statistische *Vergleichbarkeit* von Ergebnissen schätzt die Untersuchung, ob die gleichen experimentellen Ergebnisse reproduziert wurden (oder

nicht). Da die Veröffentlichungen gewöhnlich ein statistisches Maß (bspw. Durchschnitt, Cumulative Distribution Function (CDF) oder andere Messstatistiken) angeben, sollten die Ergebnisse für statistisch vergleichende Tests benutzbar sein. Das selbe Verständnis von Vergleichbarkeit wird genutzt, um zu zeigen, ob eine Methode einer anderen effektiv überlegen ist.

### 2.1.2 Methodik

Die durch IEEE Xplore veröffentlichten Tagungsbände der ersten fünf IPIN-Konferenzen enthalten 626 Veröffentlichungen (2010: 129, 2011: 95, 2012: 157, 2013: 144, 2014: 101). Die Liste der Veröffentlichungen wird pro Jahr randomisiert. Anschließend werden ca. 30% der Beiträge in einem Jahr analysiert, um gleichermaßen eine mögliche Befangenheit zu vermeiden und den benötigten Aufwand überschaubar zu halten.

Um eine emotionale Diskussion über die Einordnungen zu vermeiden, werden keine einzelnen Publikationen explizit als Positiv- oder Negativbeispiel genannt. Die Ergebnisse können jedoch ohne Weiteres durch eigene Untersuchungen der Veröffentlichungen bestätigt und erweitert werden.

Die Menge an Publikationen wird in fünf übergeordneten Kategorien untersucht:

- Typen von Lokalisierungstechniken (Systemkategorien)
- Unterscheidung der Evaluationsmethode und -szenario (Evaluationskategorien)
- Methoden zur Groundtruth-Bestimmung (Referenzkategorien)
- Vollständigkeit von Metriken (Metrikkategorien)
- Herstellung und Vergleich mit einer Ausgangssituation (Baselinekategorie)

#### Systemkategorien

Die Systemkategorie beschreibt die grundsätzliche Technologie des SUT. Falls eine Publikation ein multimodales System beschreibt, wird dies als *Multimodal* markiert. Es wird auch als Mitglied aller übrigen Kategorien gezählt, die auf das SUT passen. Meistens handelt es sich dabei um INS, welche mit RSS und/oder ToF zur Rekalibrierung bzw. zum Zurücksetzen des Drifts kombiniert werden. Andere multimodale Ansätze beschreiben Systeme für Außenlokalisierung, welche mit einem Sensor eines

Global Navigation Satellite Systems (GNSS) ausgestattet sind und über ein weiteres (Sub-)System zur Innenraumlokalisierung als Rückgriff verfügen. Falls ein Sensor nur für die Bestimmung der Groundtruth bestimmt ist, wird dieser nicht als Teil des SUT gezählt und taucht damit auch nicht in der Systemkategorie auf.

Es gibt Publikationen, welche nicht in eine der unten angegebenen Kategorien passen. Dabei handelt es sich hauptsächlich um Beiträge, die nicht direkt ein System oder einen Algorithmus beschreiben, bspw. Studien zu einem bestimmten Thema. Diese werden als *Unrelated* markiert.

Die Systemkategorien lauten:

**Inertial** Systeme, die Pedestrian Dead Reckoning (PDR) oder andere Verfahren, welche eine Inertial Measurement Unit (IMU) nutzen, implementieren.

**Map-Matching** Systeme, die jede Art von a priori generierte oder aufgezeichnete Karten nutzen, einschließlich umgebungscharakteristische Daten, wie bspw. RSSI.

**RSS** Systeme, die RSS für eine Distanzbestimmung nutzen.

**TOF** Alle Ansätze, die ToF-Verfahren zur Distanzbestimmung nutzen.

**Sound** Schallwellenbasierte Systeme, bspw. Ultraschall oder andere Tonquellen mit bekannten Positionen.

**Other** Alle übrigen Lokalisierungsansätze, welche bspw. Lichtquellen, das Magnetfeld oder Kameras zur visuellen Schätzung nutzen.

**Multimodal** Systeme, die unterschiedliche Techniken (mehrere Sensoren) zur Positionsschätzung nutzen.

**Unrelated** Veröffentlichungen, die kein Lokalisierungsverfahren beschreiben.

### Evaluationskategorien

Die Evaluationskategorien beschreiben die verwendete Methode zur Evaluation. Falls mehrere Methoden genutzt werden, wird die Veröffentlichung als *Multimodal* markiert. Sie wird wie zuvor auch in jeder übrigen passenden Kategorie gezählt. In den meisten Fällen wird eine Simulation mit realen Experimenten kombiniert. Da sich die Fehlermodelle für inertialbasierte Systeme sehr von distanzbasierten Ansätzen unterscheiden, sind die Evaluationsmethoden ebenfalls unterschiedlich. Daher werden diese

auch getrennt betrachtet.

Die Evaluationskategorien lauten:

**No-Evaluation** Veröffentlichungen, die zwar einen Lokisierungsalgorithmus oder ein -system, jedoch keinerlei Evaluation vorstellen, weder in Simulation noch in realen Experimenten.

**Simple-Simulation** Simulationen, die einfache Signalausbreitungsmodelle nutzen, welche nicht auf hindernisbezogene Effekte wie NLOS eingehen oder von gaußverteilten Fehlerverteilungen des Messsystems ausgehen.

**Complex-Simulation** Simulationen, die hindernisbezogene Effekte im Fehlermodell beachten oder experimentell gesammelte Daten als Fehlerquelle nutzen.

**Discrete-Point** Experimente mit Evaluation an diskreten Referenzpunkten. Diese sind bspw. in einer eindimensionalen Struktur angeordnet (Pfad oder gerade Linie) oder nicht-systematisch in der Umgebung verteilt.

**Grid-Like** Einfache Setups, meist in einem leeren Raum ohne Hindernisse. Die Referenzpunkte sind in einem Art Raster angeordnet.

**Office-Walk** Ein Lauf durch eine Büroumgebung auf einem bekannten Pfad.

**Outdoor** Experimente, die ausschließlich außerhalb von Gebäuden stattfinden.

**Real-World** Experimente, die das SUT in unterschiedlichen Gebäudearten, nicht unter Laborbedingungen, bspw. in bevölkerten Gebäuden oder über mehrere Tage, evaluieren.

**Multimodal** Veröffentlichungen, die mehr als eine Evaluationsmethode nutzen.

**Unrelated** Veröffentlichungen, die keine Lokalisierung vorstellen und damit keiner Evaluation bedürfen.

### Referenzkategorien

Die folgenden Kategorien beschreiben die Art des RS oder der Referenzmethode. Veröffentlichungen, die keine Informationen über die Art der Groundtruth-Bestimmung geben, werden als *Undefined* markiert.

Die Referenzkategorien lauten:

**Landmarks** Einzelne Referenzpositionen mit variierender Genauigkeit der Ground-

truth-Bestimmung.

**Path** Ein Pfad, bspw. konsekutive Referenzpunkte auf dem Fußboden, oder durch Landmarken ungefähr bestimmte Orte.

**Optical** Ein optisches Trackingsystem, bspw. an Decken hängende Kameras.

**Infrastructure** Ein RS, welches oft signifikanten Aufwand zur Installation bedarf, bspw. aber auch Roboter.

**GNSS** Durch GNSS bestimmte Groundtruth-Daten außerhalb von Gebäuden.

**Undefined** Die Nutzung von Groundtruth-Daten oder deren Bestimmung wird in der Veröffentlichung nicht genannt, oder ist nicht nachvollziehbar beschrieben.

### Metrikkategorien

Jede Systemevaluation, die die Genauigkeit bewertet, sollte quantitativ und in einer bestimmten Einheit sein. Auch sollte klar sein, wie diese Quantität gemessen wird bzw. welche Parameter Einfluss auf die Messungen haben. Schließlich sollte auch die Anzahl der Messungen, deren Genauigkeit und Präzision vorgestellt werden [91]. Diese Terminologie folgt ISO 5275-1: 1994 [90].

Die Metrikkategorien lauten:

**No-Metric** Es wird keine Metrik genannt.

**Trueness** Die Nähe und damit Messrichtigkeit zwischen dem durchschnittlichen Wert und einem glaubhaften Referenzwert, meist als Bias, arithmetisches Mittel oder als Mean Absolute Error (MAE) der Messungen ausgedrückt.

**Precision** Die Nähe zwischen unabhängigen Messungen am gleichen Ort. Meist als Varianz, Standardabweichung oder Quantil genannt.

**Accuracy** Es wird ein Maß über die miteinander kombinierte Messrichtigkeit und Präzision genannt, bspw. Mean Squared Error (MSE) oder Root Mean Squared Error (RMSE).

**Distribution** Die Datenverteilung wird, als Histogramm, CDF o. Ä., genannt.

**Sample-Size** Die Anzahl der Experimente bzw. Referenzpunkte wird genannt.

Normalerweise ist die exakte Leistungsfähigkeit eines Systems nicht bekannt, sie wird lediglich durch gesammelte Messungen geschätzt. Die Genauigkeit eines Systems ist

die Nähe zwischen experimentell bestimmten Ergebnissen sowie Referenzdaten und besteht aus den zwei Komponenten Messrichtigkeit und Präzision. Die Relation zwischen Genauigkeit, Messrichtigkeit und Präzision kann wie folgt ausgedrückt werden:

$$\text{MSE} = \text{MAE}^2 + \sigma^2 \quad (2.1)$$

Eine Erhöhung der Stichprobe  $N$ , bspw. durch Wiederholung der Messungen, beeinflusst die Varianz und damit die Genauigkeit einer Schätzung. Wenn eine statistische Unabhängigkeit zwischen Messungen angenommen wird, hängt die Stichprobenvarianz der Messungen von der Anzahl der Proben ab und verringert sich proportional zu  $N^{-1/2}$ . Dies bedeutet, dass sich die Schätzung des MAE und die Varianz mit größerer Stichprobe verbessert.

Sofern Ergebnisse nur graphisch dargestellt werden, werden nur Veröffentlichungen mit eindeutig extrahierbaren Ergebnissen in der entsprechenden Kategorie gezählt. Wenn lediglich ein Pfad dargestellt wird, gibt dies keine Möglichkeit zur eindeutigen Extraktion, da bspw. die Zeitstempel der Messpunkte fehlen, was den Pfad besser erscheinen lassen könnte, als er numerisch ist.

Betrachtet werden auch zwei abgeleitete Kategorien in dieser Studie:

**Statistics** Es sind alle Maße gegeben, welche zum Vergleich zweier Experimente oder Simulationen benötigt werden. Solche Veröffentlichungen werden in mind. zwei der Kategorien *Trueness*, *Precision* und/oder *Accuracy* gezählt, sodass die Gleichung 2.1 voll bestimmt ist.

**Complete** Alle Bedingungen für die *Statistics*-Kategorie sind gegeben, wobei zusätzlich die Stichprobengröße angegeben wird.

Die Varianz wird benötigt, um zu bestimmen, ob die Messungen eines Experiments nahe an zuvor veröffentlichten Messungen liegen und das ursprüngliche Experiment damit als reproduziert gewertet werden kann. Der RMSE kann hierfür nicht genutzt werden [92].

### Baselinekategorie

Diese Kategorie gibt eine Antwort auf die Frage, ob die Autoren einen Vergleich mit zuvor publizierten Ergebnissen oder Techniken geben (daher die Bezeichnung *baseline* (engl.) für den Ausgangswert bzw. das -szenario). Dafür sollten die reproduzier-

ten Ergebnisse mit zuvor genutzten Methoden evaluiert werden, sodass ein Vergleich ermöglicht wird. Die vorgeschlagene Methode wird dann auf den selben oder vergleichbaren Daten evaluiert. Zwar ist diese Methode gängige Praxis in den meisten Wissenschaften, jedoch ist die zeitliche Differenz zwischen der ursprünglichen und der neuen Publikation wichtig. Idealerweise ist diese relativ klein, bspw. ein Jahr, sodass tatsächlicher Fortschritt sichtbar wird.

Veröffentlichungen werden nicht in dieser Kategorie gezählt, sofern altbekannte Methoden wie Linear Least Squares (LLS) als jüngste Baseline genutzt werden. LLS ist ein gut bekannter Lokalisierungsalgorithmus, welcher sehr leicht verbessert werden kann. Falls die Publikation folglich mit einer Verbesserung gegenüber dergleichen motiviert, stellt dies keine tatsächliche Verbesserung im Sinne wissenschaftlichen Fortschritts dar. Zusätzlich wäre eine gründliche Demonstration der Verbesserung wünschenswert, bspw. durch Argumentation über die Wahrscheinlichkeit, dass die neue Methode tatsächlich von der Baseline abweichende Ergebnisse liefert.

### 2.1.3 Ergebnisse

Aus den 626 Veröffentlichungen der Konferenzreihe werden 183 gezogen und 45 davon als *Unrelated* markiert. Die folgenden Nennungen von relativen und absoluten Zahlen beziehen sich somit auf 138 relevante Veröffentlichungen, sofern nicht anders angemerkt. Tabellen zeigen die absoluten Zahlen der Veröffentlichungen in einer zugewiesenen Kategorie und die relative Anzahl in Prozent in Klammern dahinter.

#### Systemkategorien

Tabelle 2.1 zeigt die Lokalisierungsmethoden. Zwischen 68 und 85 % aller Veröffentlichungen pro Jahr passen in die zuvor definierten Kategorien. Die übrigen werden als *Unrelated* markiert. Alle Kategorien außer *Sound* (nur 9 %) teilen ähnlich große Anteile an allen Beiträgen (20 bis 28 %).

Die distanzbasierten Ansätze, welche *TOF* und/oder *RSS* nutzen, summieren sich über die Jahre auf 41 %. Distanzbasierte Veröffentlichungen verringern sich von 62 % in 2010 auf 25 % in 2011 und 2012, und erhöhen sich wieder auf 49 % in 2013 und 35 % in 2014. Dies wird mit einem Anstieg in der *Unrelated*- und *Map-Matching*-Kategorie in 2012 und 2013 begleitet. Es lässt sich kein Trend hin zu einer favorisierten



Tabelle 2.1: Systemkategorie [Anzahl (%)]

	2010	2011	2012	2013	2014	Summe
<i>Inertial</i>	7 (21)	7 (35)	8 (25)	10 (29)	7 (41)	39 (28)
nur <i>Inertial</i>	2 (6)	4 (20)	4 (13)	4 (11)	3 (18)	17 (12)
<i>Map-Matching</i>	4 (12)	6 (30)	9 (28)	6 (17)	5 (29)	30 (22)
nur <i>Map-Matching</i>	3 (9)	5 (25)	7 (22)	4 (11)	3 (18)	22 (16)
<i>RSS</i>	7 (21)	3 (15)	4 (13)	10 (29)	3 (18)	27 (20)
nur <i>RSS</i>	3 (9)	3 (15)	3 (9)	4 (11)	1 (6)	14 (10)
<i>TOF</i>	16 (47)	2 (10)	4 (13)	10 (29)	3 (18)	35 (25)
nur <i>TOF</i>	10 (29)	1 (5)	1 (3)	7 (20)	0 (0)	19 (14)
<i>Sound</i>	5 (15)	0 (0)	5 (16)	2 (6)	1 (6)	13 (9)
nur <i>Sound</i>	1 (3)	0 (0)	5 (16)	2 (6)	1 (6)	9 (7)
<i>Other</i>	8 (24)	5 (25)	8 (25)	7 (20)	4 (24)	32 (23)
nur <i>Other</i>	6 (18)	4 (20)	6 (19)	5 (14)	3 (18)	24 (17)
Multiple	9 (26)	3 (15)	6 (19)	9 (26)	6 (35)	33 (24)
<i>Related</i>	34 (100)	20 (100)	32 (100)	35 (100)	17 (100)	138 (100)

Tabelle 2.2: Evaluationskategorie [Anzahl (%)]

	2010	2011	2012	2013	2014	Summe
<i>No-Evaluation</i>	3 (9)	1 (5)	1 (3)	1 (3)	0 (0)	6 (4)
<i>Simple-Simulation</i>	10 (29)	4 (20)	1 (3)	4 (11)	4 (24)	23 (17)
nur <i>Simple-Simulation</i>	7 (21)	3 (15)	1 (3)	3 (9)	3 (18)	17 (12)
<i>Complex-Simulation</i>	1 (3)	2 (10)	2 (6)	7 (20)	2 (12)	14 (10)
nur <i>Complex-Simulation</i>	1 (3)	1 (5)	1 (3)	5 (14)	2 (12)	10 (7)
<i>Discrete-Point</i>	5 (15)	3 (15)	9 (28)	9 (26)	3 (18)	29 (21)
nur <i>Discrete-Point</i>	4 (12)	2 (10)	8 (25)	7 (20)	3 (18)	24 (17)
<i>Grid-Like</i>	5 (15)	2 (10)	5 (16)	4 (11)	1 (6)	17 (12)
nur <i>Grid-Like</i>	4 (12)	2 (10)	5 (16)	3 (9)	1 (6)	15 (11)
<i>Office-Walk</i>	9 (26)	8 (40)	11 (34)	10 (29)	6 (35)	44 (32)
nur <i>Office-Walk</i>	8 (24)	7 (35)	9 (28)	9 (26)	5 (29)	38 (28)
<i>Outdoor</i>	1 (3)	1 (5)	5 (16)	2 (6)	0 (0)	9 (7)
nur <i>Outdoor</i>	0 (0)	1 (5)	3 (9)	1 (3)	0 (0)	5 (4)
<i>Real-World</i>	4 (12)	1 (5)	2 (6)	2 (6)	1 (6)	10 (7)
nur <i>Real-World</i>	4 (12)	1 (5)	1 (3)	2 (6)	1 (6)	9 (7)
Multiple	3 (9)	2 (10)	3 (9)	4 (11)	1 (6)	13 (9)

Technologie erkennen.

76 % der Veröffentlichungen fokussieren lediglich einen Sensortyp, wogegen der Rest einen multimodalen Ansatz verfolgt. 67 % aller multimodalen Systeme sind der IMU-Kategorie zugeordnet, mit einer Rekalibrierung mithilfe eines weiteren Systems. Sie werden meist mit *RSS*-, *TOF*-, *Map-Matching*- und *Other*-Systemen kombiniert, in absteigender Reihenfolge. Es finden sich keine Systeme, die *Inertial* und *Sound* kombinieren.

Tabelle 2.3: Evaluationskategorie (nur *Inertial*) [Anzahl (%)]

	2010	2011	2012	2013	2014	Summe
<i>No-Evaluation</i>	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
<i>Simple-Simulation</i>	1 (14)	1 (14)	0 (0)	0 (0)	0 (0)	2 (5)
nur <i>Simple-Simulation</i>	0 (0)	1 (14)	0 (0)	0 (0)	0 (0)	1 (3)
<i>Complex-Simulation</i>	0 (0)	1 (14)	1 (13)	2 (20)	1 (14)	5 (13)
nur <i>Complex-Simulation</i>	0 (0)	1 (14)	0 (0)	1 (10)	1 (14)	3 (8)
<i>Discrete-Point</i>	0 (0)	0 (0)	1 (13)	0 (0)	0 (0)	1 (3)
nur <i>Discrete-Point</i>	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
<i>Grid-Like</i>	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
nur <i>Grid-Like</i>	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
<i>Office-Walk</i>	4 (57)	4 (57)	6 (75)	7 (70)	4 (57)	25 (64)
nur <i>Office-Walk</i>	3 (43)	4 (57)	5 (63)	6 (60)	4 (57)	22 (56)
<i>Outdoor</i>	1 (14)	1 (14)	2 (25)	1 (10)	0 (0)	5 (13)
nur <i>Outdoor</i>	0 (0)	1 (14)	1 (13)	1 (10)	0 (0)	3 (8)
<i>Real-World</i>	3 (43)	0 (0)	1 (13)	1 (10)	1 (14)	6 (15)
nur <i>Real-World</i>	3 (43)	0 (0)	0 (0)	1 (10)	1 (14)	5 (13)
Multiple	1 (14)	0 (0)	2 (25)	1 (10)	0 (0)	4 (10)

Tabelle 2.4: Evaluationskategorie (außer *Inertial*) [Anzahl (%)]

	2010	2011	2012	2013	2014	Summe
<i>No-Evaluation</i>	3 (11)	1 (8)	1 (4)	1 (4)	0 (0)	6 (6)
<i>Simple-Simulation</i>	9 (33)	3 (23)	1 (4)	4 (16)	4 (40)	21 (21)
nur <i>Simple-Simulation</i>	7 (26)	2 (15)	1 (4)	3 (12)	3 (30)	16 (16)
<i>Complex-Simulation</i>	1 (4)	1 (8)	1 (4)	5 (20)	1 (10)	9 (9)
nur <i>Complex-Simulation</i>	1 (4)	0 (0)	1 (4)	4 (16)	1 (10)	7 (7)
<i>Discrete-Point</i>	5 (19)	3 (23)	8 (33)	9 (36)	3 (30)	28 (28)
nur <i>Discrete-Point</i>	4 (15)	2 (15)	8 (33)	7 (28)	3 (30)	24 (24)
<i>Grid-Like</i>	5 (19)	2 (15)	5 (21)	4 (16)	1 (10)	17 (17)
nur <i>Grid-Like</i>	4 (15)	2 (15)	5 (21)	3 (12)	1 (10)	15 (15)
<i>Office-Walk</i>	5 (19)	4 (31)	5 (21)	3 (12)	2 (20)	19 (19)
nur <i>Office-Walk</i>	5 (19)	3 (23)	4 (17)	3 (12)	1 (10)	16 (16)
<i>Outdoor</i>	0 (0)	0 (0)	3 (13)	1 (4)	0 (0)	4 (4)
nur <i>Outdoor</i>	0 (0)	0 (0)	2 (8)	0 (0)	0 (0)	2 (2)
<i>Real-World</i>	1 (4)	1 (8)	1 (4)	1 (4)	0 (0)	4 (4)
nur <i>Real-World</i>	1 (4)	1 (8)	1 (4)	1 (4)	0 (0)	4 (4)
Multiple	2 (7)	2 (15)	1 (4)	3 (12)	1 (10)	9 (9)

Tabelle 2.5: Referenzkategorien [Anzahl (%)]

	2010	2011	2012	2013	2014	Summe
<i>Landmarks</i>	10 (29)	4 (20)	15 (47)	14 (40)	5 (29)	48 (35)
<i>Path</i>	4 (12)	6 (30)	7 (22)	6 (17)	2 (12)	25 (18)
<i>Optical</i>	2 (6)	1 (5)	1 (3)	2 (6)	2 (12)	8 (6)
<i>Infrastructure</i>	2 (6)	1 (5)	1 (3)	2 (6)	0 (0)	6 (4)
<i>GNSS</i>	0 (0)	0 (0)	4 (13)	1 (3)	0 (0)	5 (4)
<i>Undefined</i>	8 (24)	2 (10)	3 (9)	1 (3)	4 (24)	18 (13)

## Evaluationskategorien

Tabelle 2.2 zeigt die Verteilung der Evaluationskategorien. 96 % aller Veröffentlichungen zeigen eine Art der Evaluation, wohingegen 4 % gar keine Evaluation vorstellen. Die drei meist genutzten Evaluationsformen sind *Office-Walk* (32 %), alle Arten von *Simulation* (27 %), *Discrete-Point* (21 %), gefolgt von *Grid-Like*-Experimenten (12 %). Lediglich 7 % aller Veröffentlichungen erfüllen die zuvor definierten *Real-World*-Anforderungen. *Simple-Simulation* ging von 29 % in 2010 auf 3 % in 2012 zurück und stieg auf 24 % in 2014, während *Complex-Simulation* am stärksten in 2013 mit 20 % vertreten war. 20 % aller Veröffentlichungen führen ausschließlich eine Art von Simulation durch, wohingegen 68 % ausschließlich physikalische Experimente zeigen. 7 % führen sowohl Simulationen als auch physikalische Experimente durch.

Da sich inertialbasierte Systeme oft in der *Office-Walk*-Kategorie befinden, werden inertialbasierte und nicht-inertialbasierte Systeme im Folgenden getrennt betrachtet. Tabelle 2.3 zeigt die Prozentangaben für alle inertialbasierten Systeme. Das Gros aller Evaluationen benutzt vordefinierte Pfade in Büroumgebungen (*Office-Walk*, 64 %). Es scheint auch interessant, dass mehr Veröffentlichungen in die *Real-World*- (15 %) und *Outdoor*-Kategorie (13 %) passen. Für 2010 werden 43 % aller Veröffentlichungen als *Real-World*-Experiment kategorisiert. Dies ist signifikant höher als in den anderen Jahren (zwischen 0 und 14 %). Bei den nicht-inertialbasierten Systemen, vorgestellt in Tab. 2.4, ist *Discrete-Point*-Evaluation als dominanteste Methode erkennbar (28 %), gefolgt von *Simple-Simulation* (21 %), *Office-Walk* (19 %) und *Grid-Like* (17 %).

## Referenzkategorien

Tabelle 2.5 listet die Prozentangaben für Referenzmethoden. Es findet sich kein RS, welches nicht in die definierten Kategorien passt. Die meisten Veröffentlichungen beschreiben ihr Setup sehr gut, tendieren allerdings dazu, die Art und Weise der

Groundtruth-Bestimmung unvollständig zu beschreiben. Bei impliziter Klarheit über die genutzte Methode (meist *Landmarks*), wurde die Veröffentlichung nicht in *Undefined* kategorisiert. Es existieren allerdings Veröffentlichungen, welche die Methode der Groundtruth-Bestimmung gar nicht erwähnen. Meist zeigt eine Abbildung Lokalisierungsergebnisse hinterlegt mit dem zurückgelegten Weg als gerade Linie ohne weitere Informationen.

Die meisten Autoren wählen *Landmarks* als Groundtruth-Ressource (27%). Es finden sich Veröffentlichungen, welche explizit die Genauigkeit der manuellen Messungen zwischen Milli- und Zentimetern nennen, aber auch solche, die nur eine ungefähre Beschreibung der wahren Position geben (bspw. durch einen Punkt in einer Abbildung). Der Unterschied zu *Path*-basierten Referenzmethoden (15%) liegt darin, dass eine konsekutive Menge an Referenzpunkten als Groundtruth genutzt wird. Leider findet sich keine Veröffentlichung, die nicht nur auf die räumliche Fehlerdistanz, sondern explizit auch auf die zeitliche Distanz zwischen Referenzpunkt und Lokalisierungsergebnis eingeht.

Es findet sich wenig über *Optical*, *Infrastructure* und *GNSS* (jeweils 3 bis 4%) als Referenzmethode. *GNSS* als Groundtruth-Bestimmung ergibt lediglich bei Evaluationen außerhalb von Gebäuden auf einem längeren Pfad bei klarer Sichtverbindung zum GNSS Sinn. Dies fand sich nur bei 6% der Veröffentlichungen. Die anderen beiden Kategorien wären allerdings gut geeignet, um eine breite Menge an Systemen in einer leicht wiederholbaren Manier zu evaluieren. Der Grund für dieses Ergebnis liegt wahrscheinlich in den vermuteten Installationskosten einer solchen Lösung. Jedoch gibt es durchaus preiswerte Wege eine solche Evaluation aufzubauen, bspw. mit mobilen Robotern [66]. Es fand sich jedoch keine solche Lösung in der Stichprobe.

## Metrikkategorien

Tabelle 2.6 stellt die Verteilung bezüglich der Metrikkategorien vor. Insgesamt geben ungefähr 73% aller Veröffentlichungen eine Art von Metrik wieder. Davon zeigen 70% eine quantifizierbare Metrik, wobei drei Veröffentlichungen einen nicht-numerischen Weg wählen. Die anderen Veröffentlichungen stützen sich auf Anhaltspunkte, indem sie nur die Machbarkeit ihres Ansatzes zeigen.

Ein Maß der *Trueness* wird von 54% aller Veröffentlichungen und *Accuracy* von 38% vorgestellt. Insgesamt geben nur 9% aller Veröffentlichungen ein Maß für Genauigkeit

Tabelle 2.6: Metrikkategorien [Anzahl (%)]

	2010	2011	2012	2013	2014	Summe
<i>No-Metric</i>	17 (50)	3 (15)	8 (25)	7 (20)	6 (35)	41 (30)
<i>Trueness</i>	15 (44)	15 (75)	17 (53)	20 (57)	8 (47)	75 (54)
nur <i>Trueness</i>	10 (29)	7 (35)	6 (19)	7 (20)	4 (24)	34 (25)
<i>Precision</i>	5 (15)	8 (40)	12 (38)	14 (40)	4 (24)	43 (31)
nur <i>Precision</i>	0 (0)	0 (0)	1 (3)	1 (3)	0 (0)	2 (1)
<i>Accuracy</i>	7 (21)	9 (45)	14 (44)	16 (46)	7 (41)	53 (38)
nur <i>Accuracy</i>	2 (6)	1 (5)	3 (9)	3 (9)	3 (18)	12 (9)
<i>Distribution</i>	0 (0)	2 (10)	4 (13)	13 (37)	3 (18)	22 (16)
nur <i>Distribution</i>	0 (0)	1 (5)	3 (9)	4 (11)	0 (0)	8 (6)
<i>Sample-Size</i>	5 (15)	6 (30)	15 (47)	16 (46)	6 (35)	48 (35)
nur <i>Sample-Size</i>	1 (3)	0 (0)	1 (3)	0 (0)	1 (6)	3 (2)
<i>Trueness &amp; Precision</i>	5 (15)	8 (40)	11 (34)	13 (37)	4 (24)	41 (30)
<i>Complete</i>	2 (6)	4 (20)	9 (28)	10 (29)	3 (18)	28 (20)

an. Eine *CDF*, ein Histogramm oder die Rohdaten werden von 16 % der Veröffentlichungen gezeigt. Die Hälfte davon, 8 %, zeigt ausschließlich eine *CDF* ohne eine numerische Zusammenfassung. Zwei Veröffentlichungen (2 %) geben nur Varianzen an.

Genauigkeit ist die Kombination aus Messrichtigkeit und Präzision. Eine kombinierte Metrik ist für Vergleiche weniger nützlich, da der systematische Fehler nicht mehr von einer mangelnden Präzision unterscheidbar ist. Daher werden auch Veröffentlichungen betrachtet, welche gleichermaßen eine Schätzung der Messrichtigkeit und der Präzision angeben. Dies trifft auf 30 % der Veröffentlichungen zu.

Da die berichteten Zahlen eine Schätzung darstellen, welche auf der Stichprobengröße basiert, werden hier die Veröffentlichungen betrachtet, die gleichermaßen ein Maß der *Trueness*, *Precision* und die Stichprobengröße nennen. Dies trifft auf 20 % zu. Falls ein Vergleich solcher Ergebnisse zu einem zuvor publizierten Ergebnis angestellt wird, muss die Stichprobengröße in der Schätzung der Ähnlichkeit betrachtet werden, da diese oft eine Eingabe für die Berechnung darstellt.

Es finden sich drei Veröffentlichungen, welche zwar eine Stichprobengröße, jedoch keinerlei andere Metrik nennen. Diese präsentieren Rohdaten in einer Abbildung. Die Effektivität der vorgestellten Lösung soll durch die visuelle Darstellung erkannt werden. Allerdings werden die Rohdaten nicht veröffentlicht und können somit nicht verglichen werden.

*Metriken in Systemkategorien* Ist ein Grund für die Wahl der vorgestellten Metriken auf den IPIN-Konferenzen erkennbar? Warum nennt nur eine von fünf Veröffentlichungen Metriken, die einen vollständigen Vergleich ermöglichen? Warum nennt nahezu jede dritte Veröffentlichung keine Metriken? Da die Präsentation von Metriken variiert

Tabelle 2.7: Metrikkategorien (nur *Inertial*) [Anzahl (%)]

	2010	2011	2012	2013	2014	Summe
<i>No-Metric</i>	3 (43)	1 (14)	2 (25)	2 (20)	3 (43)	11 (28)
<i>Trueness</i>	4 (57)	6 (86)	4 (50)	5 (50)	3 (43)	22 (56)
nur <i>Trueness</i>	3 (43)	6 (86)	0 (0)	3 (30)	1 (14)	13 (33)
<i>Precision</i>	1 (14)	0 (0)	5 (63)	3 (30)	2 (29)	11 (28)
nur <i>Precision</i>	0 (0)	0 (0)	1 (13)	1 (10)	0 (0)	2 (5)
<i>Accuracy</i>	1 (14)	0 (0)	5 (63)	2 (20)	3 (43)	11 (28)
nur <i>Accuracy</i>	0 (0)	0 (0)	1 (13)	0 (0)	1 (14)	2 (5)
<i>Distribution</i>	0 (0)	0 (0)	0 (0)	4 (40)	2 (29)	6 (15)
nur <i>Distribution</i>	0 (0)	0 (0)	0 (0)	2 (20)	0 (0)	2 (5)
<i>Sample-Size</i>	1 (14)	2 (29)	5 (63)	4 (40)	3 (43)	15 (38)
nur <i>Sample-Size</i>	0 (0)	0 (0)	1 (13)	0 (0)	1 (14)	2 (5)
<i>Trueness &amp; Precision</i>	1 (14)	0 (0)	4 (50)	2 (20)	2 (29)	9 (23)
<i>Complete</i>	0 (0)	0 (0)	3 (38)	2 (20)	1 (14)	6 (15)

und die Nennung in Relation zu einem gewissen Standard manchmal nicht vollständig ist, wird im Folgenden die Verteilung der Methoden zwischen den unterschiedlichen Systemen betrachtet, auch um einer möglichen Bevorzugung bestimmter Methoden auf den Grund zu gehen.

Da die hier präsentierten Daten eine relativ kleine Menge an Veröffentlichungen beschreiben, enthalten sie eine große Fehlerspanne. Zwar können die Zahlen dadurch nicht ganz akkurat sein, allerdings zeigen Sie dennoch das Vorhandensein von Problemen auf. Konsequenterweise wird kein Modell zu diesen Daten abstrahiert oder auf Korrelationen analysiert. Eine einzelne Veröffentlichung hat einen relativ hohen Einfluss auf die Prozentangaben. Daher werden hier keine als *Sound* kategorisierten Systeme betrachtet. Tabelle 2.7 fasst die Präsentation von Metriken in inertialbasierten Veröffentlichungen zusammen. Da sich 39 Veröffentlichungen in der Kategorie *Inertial* finden, ist die Fehlerspanne mit 0,15 für ein Konfidenzlevel von 95% beschränkt.

Außer für 2010 und 2014 sind die genutzten Metriken ungefähr ähnlich der Gesamtstatistik in Tab. 2.6. Der hohe Prozentunterschied in 2010 und 2014 kommt durch eine einzige Veröffentlichung zustande.

Eine erste Beobachtung ist, dass inertialbasierte Veröffentlichungen wahrscheinlich genauso Metriken wie die Gesamtheit nennen. Es mag etwas unwahrscheinlicher sein, dass inertialbasierte Veröffentlichungen eine Messrichtigkeit und Präzision liefern. In der Stichprobe nennen mehr Autoren die Stichprobengröße als die Präzision.

Es lässt sich vermuten, dass diese Metriken oft nicht genannt werden, da Experimente mit laufenden Testpersonen schwierig zu wiederholen sind. Daher ist die Präzision

Tabelle 2.8: Metrikkategorien (nur *Map-Matching*) [Anzahl (%)]

	2010	2011	2012	2013	2014	Summe
<i>No-Metric</i>	1 (25)	1 (17)	0 (0)	2 (33)	0 (0)	4 (13)
<i>Trueness</i>	3 (75)	5 (83)	7 (78)	2 (33)	4 (80)	21 (70)
nur <i>Trueness</i>	1 (25)	2 (33)	3 (33)	0 (0)	3 (60)	9 (30)
<i>Precision</i>	2 (50)	3 (50)	4 (44)	2 (33)	1 (20)	12 (40)
nur <i>Precision</i>	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
<i>Accuracy</i>	2 (50)	3 (50)	4 (44)	2 (33)	2 (40)	13 (43)
nur <i>Accuracy</i>	0 (0)	0 (0)	0 (0)	0 (0)	1 (20)	1 (3)
<i>Distribution</i>	0 (0)	0 (0)	2 (22)	3 (50)	1 (20)	6 (20)
nur <i>Distribution</i>	0 (0)	0 (0)	2 (22)	2 (33)	0 (0)	4 (13)
<i>Sample-Size</i>	0 (0)	3 (50)	5 (56)	1 (17)	1 (20)	10 (33)
nur <i>Sample-Size</i>	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
<i>Trueness &amp; Precision</i>	2 (50)	3 (50)	4 (44)	2 (33)	1 (20)	12 (40)
<i>Complete</i>	0 (0)	3 (50)	3 (33)	1 (17)	1 (20)	8 (27)

schwer schätzbar. Bei Veröffentlichungen, welche eine Stichprobengröße nennen, ist diese eher sehr niedrig. Daher ergibt es hier tatsächlich Sinn, diese Zahlen nicht zu nennen, da ihre Signifikanz gering ist.

Trotz der erwarteten großen Fehlerspanne, lassen sich die Zahlen erklären. Dennoch sollte die Nennung der Anzahl von Versuchen allgegenwärtig sein. Hier dient diese als Begründung für das Nichterwähnen der Präzision.

Tabelle 2.8 fasst die Präsentation von Metriken in *Map-Matching*-Veröffentlichungen zusammen. Hier finden sich 30 Veröffentlichungen.

Die meisten Veröffentlichungen in der Kategorie *Map-Matching* nennen Metriken der Leistung ihres Systems. 40 % aller Veröffentlichungen nennen die Messrichtigkeit und die Präzision separat. Daher kann die Genauigkeit korrekt bestimmt werden. Die Größe der Probe ist in nur 33 % der Veröffentlichungen genannt. Daher ist es schwierig hier eine qualitative Aussage zu treffen. In vielen Fällen wird die Evaluation in der Kategorie *Map Matching* gut gemacht.

Tabelle 2.9 fasst die Metriken der *RSS*-Kategorie zusammen. Hier befinden sich 27 Veröffentlichungen in der Stichprobe. Die Daten sind nur schwer interpretierbar: drei Veröffentlichungen in 2011, drei in 2012 und drei in 2014. *RSS* war in 2010 (sieben Veröffentlichungen) und 2013 (zehn Veröffentlichungen) zur Distanzbestimmung populär.

Im Jahr 2010 nannte fast die Hälfte der Veröffentlichungen keine Metriken, im Jahr 2013 nannten dagegen fast alle eine. Jedoch wurden die Daten meist nicht in einem gut extrahierbaren Format präsentiert. Oft finden sich Abbildungen in Form

Tabelle 2.9: Metrikkategorien (nur *RSS*) [Anzahl (%)]

	2010	2011	2012	2013	2014	Summe
<i>No-Metric</i>	3 (43)	0 (0)	2 (50)	1 (10)	1 (33)	7 (26)
<i>Trueness</i>	4 (57)	1 (33)	2 (50)	5 (50)	0 (0)	12 (44)
nur <i>Trueness</i>	3 (43)	0 (0)	1 (25)	3 (30)	0 (0)	7 (26)
<i>Precision</i>	1 (14)	1 (33)	1 (25)	3 (30)	0 (0)	6 (22)
nur <i>Precision</i>	0 (0)	0 (0)	0 (0)	1 (10)	0 (0)	1 (4)
<i>Accuracy</i>	1 (14)	2 (67)	1 (25)	4 (40)	2 (67)	10 (37)
nur <i>Accuracy</i>	0 (0)	1 (33)	0 (0)	2 (20)	2 (67)	5 (19)
<i>Distribution</i>	0 (0)	1 (33)	0 (0)	5 (50)	0 (0)	6 (22)
nur <i>Distribution</i>	0 (0)	1 (33)	0 (0)	1 (10)	0 (0)	2 (7)
<i>Sample-Size</i>	0 (0)	0 (0)	1 (25)	4 (40)	1 (33)	6 (22)
nur <i>Sample-Size</i>	0 (0)	0 (0)	0 (0)	0 (0)	1 (33)	1 (4)
<i>Trueness &amp; Precision</i>	1 (14)	1 (33)	1 (25)	2 (20)	0 (0)	5 (19)
<i>Complete</i>	0 (0)	0 (0)	1 (25)	2 (20)	0 (0)	3 (11)

Tabelle 2.10: Metrikkategorien (nur *TOF*) [Anzahl (%)]

	2010	2011	2012	2013	2014	Summe
<i>No-Metric</i>	8 (50)	1 (50)	2 (50)	1 (10)	2 (67)	14 (40)
<i>Trueness</i>	7 (44)	1 (50)	1 (25)	7 (70)	1 (33)	17 (49)
nur <i>Trueness</i>	5 (31)	0 (0)	0 (0)	0 (0)	0 (0)	5 (14)
<i>Precision</i>	2 (13)	1 (50)	1 (25)	7 (70)	1 (33)	12 (34)
nur <i>Precision</i>	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
<i>Accuracy</i>	3 (19)	1 (50)	2 (50)	8 (80)	1 (33)	15 (43)
nur <i>Accuracy</i>	1 (6)	0 (0)	1 (25)	1 (10)	0 (0)	3 (9)
<i>Distribution</i>	0 (0)	0 (0)	0 (0)	4 (40)	1 (33)	5 (14)
nur <i>Distribution</i>	0 (0)	0 (0)	0 (0)	1 (10)	0 (0)	1 (3)
<i>Sample-Size</i>	3 (19)	0 (0)	0 (0)	6 (60)	1 (33)	10 (29)
nur <i>Sample-Size</i>	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
<i>Trueness &amp; Precision</i>	2 (13)	1 (50)	1 (25)	7 (70)	1 (33)	12 (34)
<i>Complete</i>	2 (13)	0 (0)	0 (0)	5 (50)	1 (33)	8 (23)

einer CDF (fünf Veröffentlichungen in 2013) und Abbildungen über die Messrichtigkeit (fünf Veröffentlichungen in 2010). Wahrscheinlich nennen Autoren hier keine Präzision, da RSS-basierte Systeme meist höchst unpräzise sind, Experimente viel Zeit verschlingen können und Messungen daher schwierig zu wiederholen sind.

Tabelle 2.10 fasst die Metriken der *TOF*-Kategorie zusammen. Hier befinden sich 35 Veröffentlichungen in der Stichprobe. In den Jahren 2011, 2012 und 2014 sind ToF-basierte Systeme nicht gut repräsentiert, daher werden diese Daten hier nicht interpretiert. In 2010 war allerdings die Nennung der Metriken nicht adäquat. Die Hälfte der Veröffentlichungen nennt gar keine Metrik, keine Veröffentlichung nennt Informationen über die Verteilung. Nur zwei Veröffentlichungen erlauben eine nennenswerte Evaluation bzw. einen Vergleich der Metriken. Dies verbesserte sich in 2013: Hier nennen fast alle Veröffentlichungen eine Metrik, wobei davon die Hälfte eine nennenswerte Evaluation bzw. einen Vergleich erlaubt.

Insgesamt findet sich keine Korrelation zwischen Metriken und verschiedenen Systeme-



Tabelle 2.11: Metrikkategorien (nur *Simple-/Complex-Simulation*) [Anzahl (%)]

	2010	2011	2012	2013	2014	Summe
<i>No-Metric</i>	6 (55)	1 (17)	0 (0)	1 (9)	3 (50)	11 (30)
<i>Trueness</i>	4 (36)	4 (67)	2 (67)	6 (55)	1 (17)	17 (46)
nur <i>Trueness</i>	3 (27)	1 (17)	1 (33)	1 (9)	1 (17)	7 (19)
<i>Precision</i>	1 (9)	3 (50)	1 (33)	6 (55)	0 (0)	11 (30)
nur <i>Precision</i>	0 (0)	0 (0)	0 (0)	1 (9)	0 (0)	1 (3)
<i>Accuracy</i>	2 (18)	4 (67)	2 (67)	7 (64)	2 (33)	17 (46)
nur <i>Accuracy</i>	1 (9)	1 (17)	1 (33)	2 (18)	2 (33)	7 (19)
<i>Distribution</i>	0 (0)	0 (0)	0 (0)	6 (55)	0 (0)	6 (16)
nur <i>Distribution</i>	0 (0)	0 (0)	0 (0)	1 (9)	0 (0)	1 (3)
<i>Sample-Size</i>	1 (9)	1 (17)	1 (33)	7 (64)	1 (17)	11 (30)
nur <i>Sample-Size</i>	1 (9)	0 (0)	0 (0)	0 (0)	0 (0)	1 (3)
<i>Trueness &amp; Precision</i>	1 (9)	3 (50)	1 (33)	5 (45)	0 (0)	10 (27)
<i>Complete</i>	0 (0)	1 (17)	1 (33)	5 (45)	0 (0)	7 (19)

men. Dies könnte teilweise an der relativ niedrigen Stichprobengröße der einzelnen Kategorien liegen. Es lässt sich allerdings vermuten, dass hier tatsächlich keine Korrelation vorliegt. Nur im Fall der inertialbasierten Systeme könnte eine Abhängigkeit bestehen.

*Metriken in Evaluationskategorien* Da die Methode, Metriken zu präsentieren, variiert und sich keine Korrelation mit Systemen zeigte, wird im Folgenden diskutiert, ob bestimmte Metriken bei verschiedenen Evaluationskategorien bevorzugt werden. Auch hier ist die Anzahl der relevanten Veröffentlichungen recht klein. Dies macht es schwer, etwas Definitives zu schlussfolgern. Daher wird auch hier keine Korrelationen berechnet. Da in den Kategorien *Grid-Like*, *Outdoor* und *Real-World* nur wenige Experimente vorgestellt werden, werden diese im Folgenden ausgeschlossen.

Tabelle 2.11 fasst die Nennung von Metriken in den *Simple-* und *Complex-Simulation-*Kategorien zusammen. Es finden sich 37 Veröffentlichungen, die Simulationen zeigen. Davon passen 23 Veröffentlichungen in die Kategorie *Simple-Simulation* und 14 Veröffentlichungen in *Complex-Simulation*. Zehn Veröffentlichungen davon nutzen eine Simulation und führen gleichzeitig Experimente durch.

Zunächst ist es überraschend, dass viele Veröffentlichungen, welche Simulationen nutzen, gar keine Metrik nennen. Gerade hier sollte dies recht einfach zu bewerkstelligen sein. Auch die Stichprobengröße wird selten mitgeteilt in 2010, 2011, 2012 und 2014. Da Simulationen äußerst einfach zu wiederholen sind und darüber hinaus sowieso mit unterschiedlichen Startbedingungen simuliert werden sollte, ist dies äußerst verwunderlich.

Tabelle 2.12 fasst die Nennung von Metriken, welche Experimente zur Evaluation

Tabelle 2.12: Metrikkategorien (außer *No-Evaluation*) [Anzahl (%)]

	2010	2011	2012	2013	2014	Summe
<i>No-Metric</i>	9 (39)	1 (7)	7 (24)	5 (19)	2 (18)	24 (23)
<i>Trueness</i>	13 (57)	13 (87)	16 (55)	16 (62)	7 (64)	65 (63)
nur <i>Trueness</i>	8 (35)	6 (40)	5 (17)	6 (23)	3 (27)	28 (27)
<i>Precision</i>	5 (22)	7 (47)	12 (41)	11 (42)	4 (36)	39 (38)
nur <i>Precision</i>	0 (0)	0 (0)	1 (3)	1 (4)	0 (0)	2 (2)
<i>Accuracy</i>	6 (26)	7 (47)	13 (45)	11 (42)	6 (55)	43 (41)
nur <i>Accuracy</i>	1 (4)	0 (0)	2 (7)	1 (4)	2 (18)	6 (6)
<i>Distribution</i>	0 (0)	2 (13)	4 (14)	9 (35)	3 (27)	18 (17)
nur <i>Distribution</i>	0 (0)	1 (7)	3 (10)	3 (12)	0 (0)	7 (7)
<i>Sample-Size</i>	4 (17)	6 (40)	15 (52)	11 (42)	4 (36)	40 (38)
nur <i>Sample-Size</i>	0 (0)	0 (0)	1 (3)	0 (0)	0 (0)	1 (1)
<i>Trueness &amp; Precision</i>	5 (22)	7 (47)	11 (38)	10 (38)	4 (36)	37 (36)
<i>Complete</i>	2 (9)	4 (27)	9 (31)	7 (27)	3 (27)	25 (24)

Tabelle 2.13: Metrikkategorien (nur *Discrete-Point*) [Anzahl (%)]

	2010	2011	2012	2013	2014	Summe
<i>No-Metric</i>	1 (20)	0 (0)	1 (11)	2 (22)	1 (33)	5 (17)
<i>Trueness</i>	3 (60)	3 (100)	7 (78)	7 (78)	2 (67)	22 (76)
nur <i>Trueness</i>	2 (40)	0 (0)	3 (33)	2 (22)	1 (33)	8 (28)
<i>Precision</i>	1 (20)	3 (100)	4 (44)	5 (56)	1 (33)	14 (48)
nur <i>Precision</i>	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
<i>Accuracy</i>	2 (40)	3 (100)	5 (56)	5 (56)	1 (33)	16 (55)
nur <i>Accuracy</i>	1 (20)	0 (0)	1 (11)	0 (0)	0 (0)	2 (7)
<i>Distribution</i>	0 (0)	1 (33)	1 (11)	2 (22)	1 (33)	5 (17)
nur <i>Distribution</i>	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
<i>Sample-Size</i>	0 (0)	2 (67)	6 (67)	5 (56)	1 (33)	14 (48)
nur <i>Sample-Size</i>	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
<i>Trueness &amp; Precision</i>	1 (20)	3 (100)	4 (44)	5 (56)	1 (33)	14 (48)
<i>Complete</i>	0 (0)	2 (67)	4 (44)	4 (44)	1 (33)	11 (38)

durchführen, zusammen. Es finden sich 104 Veröffentlichungen mit einer Art von Experiment. Davon nutzen zehn Veröffentlichungen sowohl Simulation als auch reale Experimente.

Ungefähr zwei Drittel nennt die Messrichtigkeit ihres Systems und ungefähr ein Drittel zusätzlich die Präzision. Ungefähr ein Drittel nennt eine Metrik zur Genauigkeit, entweder direkt oder durch Nennung der Messrichtigkeit und Präzision. Ungefähr ein Viertel aller Experimente nennt gar keine Metrik. Dies ist immer noch innerhalb der Fehlerspanne der Zahlen im Kontext von Simulation. Jedoch ist es sehr wahrscheinlich, dass eine experimentelle Arbeit generell ein Maß der Messrichtigkeit und wahrscheinlich auch ein Maß der Präzision nennt.

Tabelle 2.13 stellt die Nennung der Metriken in der *Discrete-Point*-Kategorie vor. Es finden sich hier 29 Veröffentlichungen. Insgesamt wird experimentelle Arbeit mit Metriken vorgestellt. Ungefähr drei Viertel nennt eine Messrichtigkeit, wobei ein Sechstel gar keine Metriken nennt.

Tabelle 2.14: Metrikkategorien (nur *Office-Walk*) [Anzahl (%)]

	2010	2011	2012	2013	2014	Summe
<i>No-Metric</i>	3 (33)	1 (13)	2 (18)	1 (10)	1 (17)	8 (18)
<i>Trueness</i>	6 (67)	7 (88)	7 (64)	6 (60)	4 (67)	30 (68)
nur <i>Trueness</i>	5 (56)	4 (50)	1 (9)	3 (30)	1 (17)	14 (32)
<i>Precision</i>	1 (11)	3 (38)	7 (64)	4 (40)	3 (50)	18 (41)
nur <i>Precision</i>	0 (0)	0 (0)	1 (9)	1 (10)	0 (0)	2 (5)
<i>Accuracy</i>	1 (11)	3 (38)	7 (64)	3 (30)	4 (67)	18 (41)
nur <i>Accuracy</i>	0 (0)	0 (0)	1 (9)	0 (0)	1 (17)	2 (5)
<i>Distribution</i>	0 (0)	0 (0)	0 (0)	5 (50)	1 (17)	6 (14)
nur <i>Distribution</i>	0 (0)	0 (0)	0 (0)	2 (20)	0 (0)	2 (5)
<i>Sample-Size</i>	2 (22)	3 (38)	6 (55)	3 (30)	2 (33)	16 (36)
nur <i>Sample-Size</i>	0 (0)	0 (0)	1 (9)	0 (0)	0 (0)	1 (2)
<i>Trueness &amp; Precision</i>	1 (11)	3 (38)	6 (55)	3 (30)	3 (50)	16 (36)
<i>Complete</i>	0 (0)	2 (25)	4 (36)	1 (10)	2 (33)	9 (20)

Tabelle 2.15: Metrikkategorien (nur *Multiple*) [Anzahl (%)]

	2010	2011	2012	2013	2014	Summe
<i>No-Metric</i>	1 (33)	0 (0)	1 (33)	0 (0)	0 (0)	2 (15)
<i>Trueness</i>	2 (67)	2 (100)	1 (33)	3 (75)	0 (0)	8 (62)
nur <i>Trueness</i>	1 (33)	0 (0)	0 (0)	0 (0)	0 (0)	1 (8)
<i>Precision</i>	1 (33)	2 (100)	1 (33)	4 (100)	0 (0)	8 (62)
nur <i>Precision</i>	0 (0)	0 (0)	0 (0)	1 (25)	0 (0)	1 (8)
<i>Accuracy</i>	1 (33)	2 (100)	2 (67)	3 (75)	1 (100)	9 (69)
nur <i>Accuracy</i>	0 (0)	0 (0)	1 (33)	0 (0)	1 (100)	2 (15)
<i>Distribution</i>	0 (0)	0 (0)	0 (0)	2 (50)	0 (0)	2 (15)
nur <i>Distribution</i>	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
<i>Sample-Size</i>	0 (0)	1 (50)	1 (33)	3 (75)	0 (0)	5 (38)
nur <i>Sample-Size</i>	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
<i>Trueness &amp; Precision</i>	1 (33)	2 (100)	1 (33)	3 (75)	0 (0)	7 (54)
<i>Complete</i>	0 (0)	1 (50)	1 (33)	3 (75)	0 (0)	5 (38)

Tabelle 2.14 zeigt die Nennung von Metriken in der *Office-Walk*-Kategorie. Es finden sich 44 Veröffentlichungen. Ungefähr zwei Drittel nennt eine Messrichtigkeit, wobei auch hier ein Sechstel gar keine Metriken nennt.

Tabelle 2.15 zeigt die Nennung von Metriken in der *Multiple*-Kategorie. Zwar finden sich hier nur 13 Veröffentlichungen, dennoch werden sie hier vorgestellt, da die sichtbare Verteilung sehr nah an der Verteilung der Experimente liegt.

*Zusammenfassung* Man sollte die Qualität einer Veröffentlichung nicht ausschließlich anhand hier benutzter Kriterien schätzen. Es ist berechtigt, keine oder nicht alle Metriken zu nennen. Zum Beispiel ist die Nennung der Varianz nicht wichtig, wenn die Stichprobe recht klein ist. Manche Autoren stützen ihre Aussagen auf nur drei Wiederholungen eines Experiments.

Manche Experimente werden zur Demonstration der Machbarkeit des jeweiligen Systems genutzt und nicht zur Schätzung der Leistung oder zum Vergleich mit anderen

Tabelle 2.16: Baselinekategorie [Anzahl (%)]

	2010	2011	2012	2013	2014	Summe
Total	0 (0)	2 (10)	2 (6)	5 (14)	1 (6)	10 (7)
<i>Inertial</i>	0 (0)	0 (0)	1 (3)	1 (3)	0 (0)	2 (1)
<i>Map-Matching</i>	0 (0)	2 (10)	1 (3)	0 (0)	0 (0)	3 (2)
<i>RSS</i>	0 (0)	0 (0)	0 (0)	2 (6)	1 (6)	3 (2)
<i>TOF</i>	0 (0)	0 (0)	0 (0)	1 (3)	0 (0)	1 (1)
<i>Sound</i>	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
<i>Other</i>	0 (0)	0 (0)	0 (0)	2 (6)	0 (0)	2 (1)

Systemen. Auf der anderen Seite ist eine Leistungsanalyse nötig, sobald die Grundprinzipien eines Systems verstanden sind. Andernfalls ist der Beitrag zum aktuellen Stand der Forschung nicht nachvollziehbar. Wenn zum Beispiel ein neuer Lokalisierungsalgorithmus vorgestellt wird, sollte dessen Leistung mit bisherigen Verfahren verglichen werden, um den Nutzen des neuen Verfahrens einschätzen zu können.

Dennoch ist zu betonen, dass nur ein Drittel der Veröffentlichungen eine Schätzung der Messrichtigkeit und der Präzision gibt und nur ungefähr ein Fünftel alle nötigen Metriken zum statistischen Vergleich mit anderen Ergebnissen nennt. Dies ist überraschend, da es durch das bekannte Setup und die gesammelten Daten recht einfach sein sollte, diese Metriken zu berechnen oder zumindest zu erklären, warum sie nicht relevant sind.

### Baselinekategorie

Tabelle 2.16 stellt die Verteilung bezüglich der Baselinekategorie vor. Nur wenige Veröffentlichungen stellen eine externe Baseline zur Evaluation vor (7.1% bzw. 13 aller Veröffentlichungen inkl. *Unrelated*). Davon gehören zehn Veröffentlichungen nicht der *Unrelated*-Kategorie an: zweimal *Inertial*, dreimal *Map-Matching*, dreimal *RSS*, einmal *TOF*, keinmal *Sound* und zweimal *Other* (eine Veröffentlichung kombiniert ein INS mit RSS). Da sich nur wenige Veröffentlichungen in der Baselinekategorie finden, können keine signifikanten Aussagen zur Wahl der Baseline getroffen werden.

Die niedrigen Zahlen sind zunächst überraschend. Folgende Erklärungen für diese Beobachtung sind denkbar:

1. Viele Vorschläge sind neu, ohne vergleichbare Baseline. Der Zweck der Evaluation liegt in der Demonstration der Methode bzw. ihrer Effektivität.
2. Andere Methoden, die als Baseline in Frage kommen, sind unklar definiert. Für einen Vergleich muss die zuvor vorgeschlagene Methode in der eigenen Umge-

bung mit vorhandener Ausrüstung umgesetzt und sichergestellt werden, dass nur die Leistung der neuen Methode verglichen wird und kein anderer Parameter der Umgebungen. Dies fordert eine Wiederherstellung der Bedingungen, was bei unvollständiger Dokumentation nicht möglich ist.

3. Der Aufwand wird als zu groß eingeschätzt. Für eine dem wissenschaftlichen Standard entsprechende Evaluation sollte jedes Experiment für eine bessere Präzision mehrmals wiederholt werden. Experimente inkl. Auf- und Abbau können mehrere Stunden in Anspruch nehmen. Die Wiederherstellung einer Baseline verdoppelt leicht den Aufwand.

Die Zählweise dieser Arbeit betrachtet keine internen Baselines. Derartige Veröffentlichungen zeigen bspw. ungefilterte/unverarbeitete Ergebnisse als Baseline; gefilterte/verarbeitete Ergebnisse zeigen dann die Verbesserung.

Relativ wenige Veröffentlichungen quantifizieren den Nutzen ihrer vorgestellten Methode. Es wird nicht plausibel dargelegt, ob die Verbesserung mit Rauschen oder durch eine tatsächlich statistisch signifikante Verbesserung erklärt werden kann. Dies gilt vor allem für Veröffentlichungen, die eine Kombination von Methoden vorstellen, wobei jede zusätzliche Methode weniger Nutzen bringt. Es lässt sich hier nicht schlussfolgern, ob der Zusatz tatsächlich einen Effekt nach sich zieht, wenn keine Varianz angegeben ist oder bei CDF-Vergleichen kein Kolmogorov-Smirnov-Test (KS-Test) durchgeführt wurde. Letzteres würde helfen, den Unterschied zweier CDF zu verstehen und zu quantifizieren. Der Mangel an Baseline-Evaluationen macht es schwierig, den Fortschritt über die Jahre zu quantifizieren.

#### 2.1.4 Genauigkeit der Studie

Der Aufwand zur Kategorisierung von Veröffentlichungen ist relativ hoch, daher wurden nicht alle Veröffentlichungen der IPIN-Konferenzreihe kategorisiert. Auch wurden keine anderen großen Konferenzen, wie bspw. UPIN-LBS, WPNC oder andere mit Schwerpunkt auf Lokalisierung innerhalb von Gebäuden, betrachtet. Daher macht diese Studie lediglich Aussagen über die Publikations- und Auswahlhistorie der IPIN-Reihe.

## Systematische Fehler

Die größten Quellen von systematischen Fehlern sind falsche Klassifikation und ein Selektionsbias.

*Fehler durch falsche Klassifikation* Jeder Veröffentlichung wurde durch mindestens eine Person gelesen und klassifiziert. Meist machen die Autoren eine starke Aussage, die eine gute Klassifizierung nach den definierten Kategorien erlaubt. Jedoch könnte die Person die Veröffentlichung dennoch falsch klassifiziert haben. Gründe dafür könnten Uneindeutigkeiten der Definition einer Klasse sein. So ist es bspw. nicht ganz klar, wo die Grenze zwischen *Simple-* und *Complex-Simulation* liegt.

Manche Veröffentlichungen passen nicht eindeutig in eine der Kategorien und wurden daher als *Unrelated* markiert. Falls bspw. ein Experiment beschrieben wurde, ohne numerische Metriken zu nennen, wird die Veröffentlichung entsprechend in der *Undefined-Metrikkategorie* gezählt. Dies tritt bspw. ein, wenn Messungen in Diagrammen gezeigt werden, und davon kein klarer Wert abgeleitet werden kann.

Da nicht jede Veröffentlichung auch durch eine zweite Person klassifiziert wurde, kann keine genaue Schätzung über eine falsche Klassifikation angegeben werden. Unklare Klassifikationen wurden allerdings von den Autoren der Studie untereinander besprochen, sodass der Fehler hier nicht ins Gewicht fallen sollte.

*Fehler durch Selektionsbias* Die zweite Quelle systematischer Fehler ist der Selektionsbias. Die Veröffentlichungen könnten nach einem bestimmten Stil oder einer bestimmten Qualität ausgewählt sein.

Die Veröffentlichungen wurden für diese Studie wie folgt ausgewählt: ein selbstgeschriebenes Programm lud die Listen aller Veröffentlichungen der IPIN-Konferenzen in IEEE Xplore pro Jahr herunter, randomisierte die Listen pro Jahr und wählte zufällig mind. 30% der Veröffentlichungen pro Jahr aus. In diesen Schritten fand keine manuelle Einflussnahme statt. Die Veröffentlichungen der IPIN 2014 waren zum Zeitpunkt der Studie noch nicht über IEEE Xplore veröffentlicht. Daher wurde hier ein ähnliches Programm geschrieben, das die zu kategorisierende Menge direkt aus dem Tagungsband der Konferenz extrahierte.

Dadurch sollte der Auswahlprozess für zu kategorisierende Veröffentlichungen möglichst unvoreingenommen und die Auswahl repräsentativ für auf dieser Konferenzreihe

vorgestellte Veröffentlichungen sein.

### Statistische Fehler

Statistische Fehler bestehen aus zufälligen falschen Klassifikationen und einer schlechten Genauigkeit der Klassifizierung. Da hier kein Maß für den Klassifizierungsfehler existiert, ist dieser nicht schätzbar. Durch vorherige Absprachen und Konsultationen während der Klassifizierung zwischen den Autoren sollte eine weitgehend eindeutige Klassifizierung sichergestellt werden.

Die Fehlerspanne der Einordnungen kann geschätzt werden. Mit einer Konfidenz von 95% liegt diese bei 0,03 über alle Konferenzen hinweg. Bei den einzelnen Jahren finden sich Fehlerspannen von 0,05 bis 0,07 bei gleicher Konfidenz. Die Anteile pro Jahr hätten derart gewählt werden können, dass die Fehlerspanne jeweils identisch ist. Dies würde unterschiedliche Anteile pro Jahr implizieren und hätte die Menge der Kategorisierungen insgesamt erhöht und daher die Fehlerspanne reduziert. Diese Arbeit wählt die ökonomischere Lösung und gibt somit eine Fehlerspanne bis zu 0,07 mit Konfidenz 95% an.

Pro Kategorie ist die Fehlerspanne höher. Diese Arbeit bietet eine Schätzung der Veröffentlichungen in einer Kategorie, wobei das Stichprobenergebnis womöglich zu niedrig ist. Die Arbeit versucht, Aussagen über Kategorien zu vermeiden, bei denen die Fehlerspanne bei Konfidenz 95% über 0,15 liegt. Es wäre unmöglich, eine sichere Aussage zu treffen, falls die Fehlerspanne auch eine gegensätzliche Erklärung erlauben würde. Dies tritt wahrscheinlich bei einer Veröffentlichungsanzahl unter 30 ein.

### Gesamtfehler

Diese Arbeit sollte ein relativ genaues Bild über alle Publikationen der IPIN zeigen. Es lässt sich vermuten, dass die Fehlerspanne der falschen Klassifikation durch den Leser in der selben Größenordnung wie der statistische Fehler liegt. Die Daten pro Jahr und Kategorie sind weniger genau, wobei dies die insgesamten Schlussfolgerungen nicht beeinflusst.

### 2.1.5 Zusammenfassung

Insgesamt führen 95 % aller nicht *Unrelated*-Veröffentlichungen der untersuchten IPIN-Konferenzen zumindest eine Art der Evaluation durch. Physikalische Experimente spielen mit 77 % eine wichtige Rolle. Es lässt sich eine Tendenz hin zu komplexeren Simulationen und eine gleichbleibende Verteilung zwischen verschiedenen physikalischen Evaluationensmethoden erkennen. Die Relevanz von ausschließlichen *Outdoor*-Veröffentlichungen (3 %) bei einer Konferenz für die Lokalisierung innerhalb von Gebäuden lässt sich bezweifeln.

Während sich Autoren auf prinzipiell gute Evaluationen stützen, ist deren Beschreibung durchwachsen. Viele Autoren beschreiben die Methode der Groundtruth-Bestimmung mangelhaft. Obwohl oft nichts explizit beschrieben wird, lassen sich manuelle Messungen mit Zollstöcken u. ä. zur Bestimmung des Distanzfehlers vermuten.

Die Qualität von Experimenten wurde nicht evaluiert. Auch kann nicht unterstellt werden, dass Veröffentlichungen, die nicht oder nicht gut in die definierten Kategorien passen, von geringer Qualität sind. Eine Veröffentlichung könnte, aufgrund der Intention der Autoren, nicht zur definierten Klassifizierung passen, obwohl es ein wertvoller Beitrag ist, bspw. Konzept- und Machbarkeitsstudien. Auch passen manche Veröffentlichungen nicht in die Kategorien, weil anders gemessen oder evaluiert wird. Jedoch ließen sich während der Analyse keine weiteren Kategorien erkennen, sodass die wichtigsten abgebildet sein dürften.

Zum wissenschaftlichen Standard eines Experiments gehört die Formulierung einer Hypothese zur Leistungsfähigkeit eines Systems, eine Beschreibung, wie diese getestet wird, und einer Diskussion der Schritte, wie ein möglicher Bias ausgeschlossen wird. Zum Beispiel ist gut bekannt, dass PDR unter einem Drift leidet, sodass der beobachtete Fehler mit der Zeit größer wird. Es lassen sich allerdings selten Aussagen finden, die den beobachteten Fehler in Relation zur Distanz oder Zeit setzen. Nur 35 % der Veröffentlichungen nennen die Anzahl der Wiederholungen eines Experiments. Auch wird der Leser nicht über Daten aufgeklärt, die die Hypothese widerlegen könnten. Es wurde keine quantitative Schätzung des systematischen Fehlers in der experimentellen Evaluation gefunden. Zum Beispiel sollte bei GPS-basierten Referenzsystemen in Städten die Präzision des Signals verfolgt werden. Wenn diese bspw. bei 10 m liegt, und ein Positionsfehler beim SUT von 5 m gemessen wird, sagt das Experiment wenig über die Leistung des SUT aus. Ein Mangel an offener Kommunikation der Experi-



mentdetails macht es hier schwierig, die Ergebnisse unabhängig zu bewerten.

Experimente sind zentraler Bestandteil der Wissenschaft. Autoren sollten, im Sinne von Vergleichbarkeit, Reproduzierbarkeit und offener Kommunikation, über ihre Methode der Groundtruth-Bestimmung schreiben. Dies betrifft nicht nur die wahre Position, sondern auch die Zeit der Messung, welche oft nicht erwähnt wird. Die wissenschaftliche Gemeinde sollte mehr physikalische Experimente fordern. Es ist überaus wichtig, eine Lösung unter unterschiedlichen Bedingungen zu evaluieren. Zur Wiederholung lässt sich die Nutzung von automatisierten Referenzsystemen empfehlen, bspw. Roboter oder optische Systeme. Simulationen könnten durch die Veröffentlichung von Datensets verbessert werden. Hilfreich wären auch standardisierte Benchmarks zur Evaluation von Lokalisierungssystemen innerhalb von Gebäuden.

## 2.2 Einführung eines automatisierten Referenzsystems

Zum Vergleich verschiedener Lokalisierungslösungen wird im Folgenden ein automatisiertes RS vorgestellt. Dieses Kapitel basiert auf folgender eigener Veröffentlichung:

- S. Schmitt, H. Will, B. Aschenbrenner, T. Hillebrandt und M. Kyas, “A reference system for indoor localization testbeds”, in *Indoor Positioning and Indoor Navigation (IPIN)*, *International Conference on*, Nov. 2012. DOI: 10.1109/IPIN.2012.6418865

Diese Arbeit wurde teilweise bereits zuvor in weiteren eigenen Arbeiten vorgestellt:

- S. Schmitt, “Design und Implementierung eines mobilen Referenzsystems für die Indoorlokalisierung unter Verwendung des Robot Operating Systems”, Bachelorarbeit, Freie Universität Berlin, Jan. 2012
- S. Schmitt, “Aufbau und Evaluation eines virtuellen Testbeds zur Indoor Lokalisierung in drahtlosen Sensornetzwerken”, Masterarbeit, Freie Universität Berlin, Nov. 2013

Durch Nutzung des RS entstand darüber hinaus folgende eigene Veröffentlichung:

- S. Schmitt, S. Adler und M. Kyas, “The effects of human body shadowing in RF-based indoor localization”, in *Indoor Positioning and Indoor Navigation (IPIN)*, *International Conference on*, Okt. 2014, S. 307–313. DOI: 10.1109/IPIN.2014.7275497

### 2.2.1 Motivation

Während weit verbreitete Simulationen von Lokalisierungssystemen und -algorithmen für den Einsatz in Gebäuden eine signifikante Einsicht in das Verhalten in bestimmten Szenarien erlauben, ist eine detailgetreue Modellierung der Umgebung und deren Bedingungen sehr schwierig. Eine regelmäßige Analyse unter Bedingungen realer Umgebungen ist essentiell, um neue Algorithmen zu optimieren. Experimente in realen Umgebungen sind durchaus schwierig zu bewerkstelligen, da ein RS, welches eine deutlich präzisere Ortsbestimmung als das SUT erlaubt, für die Bestimmung von Groundtruth-Informationen nötig ist. Mit einem solchen System könnten verschiedenartige Lokalisierungstechniken evaluiert werden, bspw. ToF-Messungen. Ohne ein weitgehend automatisiertes RS müsste die Groundtruth händisch bestimmt werden. Dies ist entsprechend schwierig und benötigt einiges an Zeiteinsatz. Neben der Position enthält die Groundtruth mindestens auch die Zeit, zu der der Messwert aufgezeichnet bzw. die Position geschätzt wurde – je nachdem welche Ebene betrachtet wird. Soll ein kontinuierlich bewegtes Objekt verfolgt werden, wird die Automatisierung des Prozesses umso wichtiger, um Ressourcen einzusparen.

Für eine einfache Umsetzung eines automatisierten RS werden im Folgenden mobile Roboter genutzt. Um ein anderes Lokalisierungssystem zu evaluieren, sollte ein solches RS in der Lage sein, die Komponenten des SUT zu tragen. Während der Fahrt durch das Gebäude dürfen sich beide Systeme in ihrer Lokalisierung nicht beeinflussen oder voneinander abhängen. Zum Beispiel könnten nahestehende Personen, die das RS kontrollieren, die Messungen des SUT systematisch beeinflussen. Daher sollte das RS weitgehend autonom oder entsprechend ferngesteuert werden können. Die dafür genutzten Funkfrequenzen dürfen dementsprechend nicht mit dem SUT interferieren. Um eine einfache Nutzung zu gewährleisten, sollte das RS keine zusätzliche Infrastruktur, wie bspw. installierte Kameras oder Funkbaken, benötigen. Das RS sollte seine Position in einem festgelegten Koordinatensystem schätzen. Der Fehler dieser Schätzung muss zwingend eine Größenordnung kleiner als der erwartete Fehler des SUT sein, um verschiedene Lokalisierungssysteme und -algorithmen gut miteinander vergleichen zu können.

Die Literatur zeigt, dass Simulation als Evaluierungstechnik zur Evaluation von Lokalisierungssystemen und -algorithmen innerhalb von Gebäuden äußerst dominant ist. In realen Experimenten mit drahtlosen Sensornetzwerken werden meist statische Messungen genutzt, um das SUT zu evaluieren. Piras und Cina messen bspw. an

zuvor definierten Punkten entlang eines Pfades [96]. Ein Wagen, welcher das Lokalisierungsequipment trägt, wird händisch den Pfad entlang geschoben. Bal, Xue und Shen et al. bauen ein Testbed zur Lokalisierung und Verfolgung in einem drahtlosen Sensornetzwerk [97]. Die Autoren fokussieren auf Industrieumgebungen. Die mobilen Komponenten werden auf bewegliche Teile von Maschinen, auf Tischen, auf metallischen Regalen und auf Industrierobotern angebracht. Allerdings wird die Groundtruth nicht automatisch bestimmt.

Diese Ansätze unterstützen kein mobiles und automatisiertes Testen. Jedoch lässt sich dies zu den Anforderungen an den Evaluationsprozess zählen, wenn mobile Komponenten zum SUT zählen. Die meisten vorgestellten automatisierten Systeme nutzen vorbereitete Infrastruktur in der Umgebung. So nutzen Johnson, Stack und Fish et al. mobile Roboter mit einem eingebauten Funkgerät zur Evaluation von Lokalisierungsalgorithmen [98]. Die Autoren nutzen Emulab zur Kontrolle eines drahtlosen Netzwerks aus mehreren Knoten. Spezielle Software wird zur Kontrolle und Verfolgung der Roboter genutzt. An der Decke angebrachte Kameras lokalisieren die Roboter, mit einem durchschnittlichen Fehler von 0,28 cm. Jedoch wird dies nur nach guter Vorbereitung und einer Feinkalibrierung des Systems in einer abgesteckten Umgebung erreicht. Es werden sechs Kameras für sechs Roboter in einem Areal von 60 m<sup>2</sup> installiert. Prorok, Arfire und Bahr et al. nutzen neben anderen Sensoren auch an der Decke installierte Kameras, um mehrere Roboter in einem abgegrenztem Areal zu verfolgen [99]. Die Genauigkeit des Systems liegt bei ca. 1-3 cm, was ebenfalls nur nach einem präzisen Aufbau und einer zuvor stattfindenden Kalibrierung gelingt. Rydell und Emilsson zeigen ebenfalls ein kamerabasiertes System, wobei die vom zu lokalisierenden Objekt getragene Kamera zuvor installierte visuelle Landmarken in der Umgebung für eine Lokalisierung nutzt [100]. Segura, Hashemi und Sisterna et al. zeigen ein Lokalisierungssystem, welches auf verteilten Funkmodulen und Ultra Wide Band (UWB) basiert [101]. Die Module erlauben eine Selbstlokalisierung eines Roboters mit einem Fehler unter 20 cm.

Der Nachteil dieser Implementierungen liegt in der benötigten Infrastruktur. Um ein mobiles RS in üblichen Gebäuden, ohne zeitintensive Vorbereitung und nicht in räumlich stark eingegrenzten Arealen, zu nutzen, wird ein vollständig unabhängiges System benötigt. Ein dem hier vorgestellten sehr ähnlichen System stellen Lemic et al. 2014 vor [102, 103]. Dies baut in wesentlichen Teilen auf dieser Arbeit auf und erweitert diese um künstlich erzeugte Interferenz, zur Evaluation des SUT unter kontrollierten Funkbedingungen. Im Folgenden wird ein präzises, kostengünstiges und mobiles RS

vorgestellt, welches keinerlei Infrastruktur benötigt und eine robuste Lokalisierung in verschiedenartigen Gebäudestrukturen erlaubt.

### 2.2.2 Automatisches Referenzsystem

Zu den wesentlichen Komponenten des RS gehören die ausgewählte Hardware, die Softwarebasis und die Generierung der Groundtruth.

#### Hardware

Es wird ein kostengünstiger TurtleBot-Roboter [104] genutzt, dargestellt in Abb. 2.1. Dieser besteht aus einem Roomba 531 Staubsaugerroboter der Firma iRobot, einer Microsoft Kinect Tiefenbildkamera, einem separaten Akku für die Kamera, und einem Aufbau zur Montage von zusätzlichen Komponenten wie dem SUT. Es handelt sich um ein nicht-holonomes Fahrwerk<sup>1</sup> mit zwei unabhängigen Antriebsrädern und einem Stützrad. Da Drehungen auch ohne lineare Geschwindigkeit des Roboters möglich sind, ist er als quasi-holonom anzusehen. Zur Durchführung der Lokalisierung des RS wird ein handelsüblicher Laptop über die Basis montiert. Dadurch ist der Roboter in der Lage, über jegliche flachen Böden, wie bspw. glatte Holzböden oder auch Teppiche, und über Türschwellen mit einer ungefähren maximalen Höhe von 0,5 cm autonom zu fahren. Der Roboter erlaubt eine Maximalgeschwindigkeit von 0,5 m/s. Die Maße des Roboters von ungefähr 33 cm im Durchmesser und 40 cm Höhe sind ausreichend, um die meisten SUT zu montieren.

Die Kinect erlaubt gegenüber leistungsfähigeren Laserscannern eine kostengünstige Möglichkeit, um Distanzen auf Sicht zwischen 50 cm und bis zu ungefähr 5 m zu schätzen [105]. Der gemessene Blickwinkel beträgt ca. 58°. Während die Genauigkeit der Kinect relativ hoch ist, ist der Distanzfehler einer Messung proportional zum Quadrat der Distanz. Bei einer Distanz von 2,5 m liegt der durchschnittliche Distanzfehler bei 4,5 cm, wobei bereits 5 % der größten Ausreißer entfernt wurden. Auf einer Distanz von 5 m liegt der Fehler bereits bei signifikanten 19 cm. Zur Unterstützung des Lokalisierungsprozesses wird eine zweite Kinect auf dem Roboter installiert, welche nach hinten gerichtet ist. Dies garantiert, dass das RS immer ausreichend viele Objekte in der Umgebung in Sicht hat, insbesondere in Räumen, die größer als 5 m sind.

---

<sup>1</sup>Eine Rollbewegung ist nur senkrecht zur Radachse möglich.

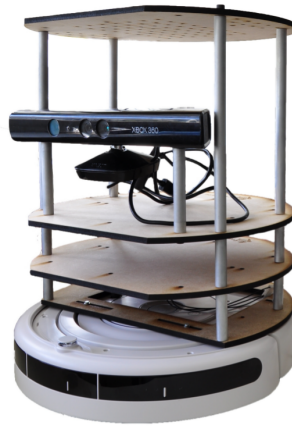


Abbildung 2.1: TurtleBot: Roomba 531 mit einer montierten Microsoft Kinect Tiefenkamera. Zur Lokalisierung ist zusätzlich ein handelsüblicher Laptop nötig.

Dies hilft auch bei vorübergehenden Personen, die das Sichtfeld auf die Umgebung temporär verdecken. Bei größeren Räumen mit Wänden außerhalb der maximalen Reichweite beider Kameras können entsprechend platzierte und in der Karte eingetragene Einrichtungsgegenstände zur Lokalisierung genutzt werden. Zwar würde ein (deutlich teurerer) Laserscanner die Lokalisierungsgenauigkeit gegenüber bekannten Hindernissen deutlich erhöhen [63, 102], allerdings ist die global betrachtete Genauigkeit auch von der Güte der notwendigen Karte abhängig. Die Arbeit geht von einem Kartenfehler von unter 10 cm aus. Da die Lokalisierungsgenauigkeit mithilfe der Kinect in der selben Größenordnung liegt, erfüllt dies die Anforderungen ausreichend. Zu beachten ist weiterhin, dass durch den Positionsfehler des RS und durch Eigenheiten der Sensorik (Mindestabstand der Tiefenbildkamera) eine kollisionsfreie und automatische Fahrt nahe einer Wand mit Entfernung unter ungefähr 30 cm nicht möglich ist. Falls Daten hier generiert werden sollen, kann das RS dazu jedoch manuell gesteuert werden, sofern die Tiefenbildkameras genügend Informationen zur Lokalisierung erhalten.

Zur Definition des Koordinatensystems und zur Lokalisierung des Roboters werden zuvor erstellte oder aufgezeichnete Karten der Umgebung genutzt. Dies können präparierte Gebäudepläne, bspw. in Form von Flucht- oder Architektenplänen, sein. Ist keine solche vorhanden oder der manuelle Aufwand zur Erstellung zu hoch, kann das im Folgenden vorgestellte System diese prinzipiell automatisch erstellen. Bei ungünstiger Geometrie des Gebäudes kann dabei jedoch ein Drift entstehen, da der systembedingte Fehler der Odometrie maßgeblich die Qualität des Resultats beeinflusst. Ein erkennbarer Drift in der Karte verhindert eine einfache Relation zwischen

Karte und Distanzmessung bzw. der Ortsbestimmung des SUT.

Um den Lokalisierungsfehler des Systems klein zu halten, müssen genügend in der Karte eingetragene Objekte in seinem Sichtfeld vorhanden sein. Es wird eine Nutzung in üblichen, büroähnlichen Gebäuden mit Flurbreiten zwischen 1 und 3 m mit ebenfalls büroähnlichen Räumen und mit nicht mehr als 10 m Raumdurchmesser intendiert.

### Robot Operating System

Das Robot Operating System (ROS) ist ein quelloffenes Framework zur Steuerung verschiedenartiger Roboter [106, 107]. Es bietet meist benötigte Treiber verbreiteter Hardware (u.a. Kinect und Roomba) und verbindet relevante Datenquellen mithilfe asynchroner Nachrichtenkanäle. Die Bestandteile sind generisch aufgebaut, sodass neue Funktionalitäten meist mit vorhandenen Modulen aufgebaut werden können. Eine Lokalisierungskomponente (Adaptive Monte Carlo Localization (AMCL) [108]) schätzt die Position des Roboters anhand der Kinect-Daten in einer 2D-Karte, indem Tiefeninformationen und Odometriedaten mit ebendieser abgeglichen werden. Abbildung 2.2 zeigt einen Kartenausschnitt, den Roboter an seiner geschätzten Position und die mit der Karte abgeglichenen Tiefeninformationen. Zur Initialisierung benötigt AMCL eine händisch eingegebene Startposition. Eine weitere Komponente (Gmapping) erlaubt die Erstellung von Karten durch Odometrie und Kinect. Wie erwähnt, führt dies allerdings bei größeren Umgebungen zu einem Drift in der Karte, sodass eingetragene und tatsächliche Winkel sowie Entfernungen weit auseinander liegen können. Für kleinere Umgebungen mit wenigen Räumen kann dies jedoch genutzt werden. Zur Automatisierung wird eine Komponente namens Move-Base genutzt. Hier werden Wege zwischen dem aktuellen Ort und einer vorgegebenen Zielposition autonom geplant, abgefahren und dabei Hindernisse umfahren. Soll auch der Pfad vorgegeben werden, ist ebenso eine manuelle Steuerung möglich.

### Groundtruth-Bestimmung und Wiederholbarkeit

Das SUT wird an den auf dem Roboter befindlichen Laptop angeschlossen. Dadurch kann eine Software auf dem Laptop die gemessenen Daten oder die geschätzte Position des SUT abgreifen. Die geschätzte Position des Roboters und ein entsprechender

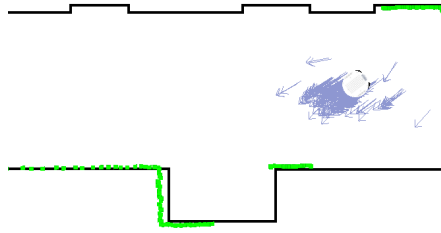


Abbildung 2.2: TurtleBot in einem Flur. Die Tiefeninformationen der Kinect (grün) werden mit bekannten Hindernissen in der Karte abgeglichen. Eine Menge an möglichen und wahrscheinlichen Positionen ist durch die violetten Pfeile gegeben. Die Einbuchtung im unteren Teil der Karte stellt eine (als geschlossen eingetragene) Tür dar.

Zeitstempel dazu werden direkt aus AMCL abgegriffen. Die beiden Informationsquellen generieren zeitlich voneinander unabhängig Daten, allerdings stellt der Laptop als Verbindungsschnittstelle eine gemeinsame Uhr, sodass keine weitere Zeitsynchronisierung nötig wird. Generierte Daten beider Quellen werden in jeweils einer chronologischen Warteschlange vorgehalten. Nachdem ein Groundtruth-Datum erzeugt wurde, werden alle vorhandenen Distanzmessungen des SUT iteriert. Für alle Messungen, für die sowohl ein Groundtruth-Datum vor, als auch unmittelbar nach ihren Zeitstempeln vorliegt, wird eine künstliche Groundtruth-Position linear interpoliert. Dazu wird angenommen, dass der Roboter seine Geschwindigkeit nicht abrupt und seine Position in relevanten Zeiteinheiten nicht unverhältnismäßig verändert. Da der Roboter in der Datenaufzeichnung maximal 10 cm/s fährt und Positionsänderungen mit ungefähr 10 Hz abgegriffen werden, ist der maximale Abstand zweier gesammelter Groundtruth-Daten normalerweise auf 1 cm limitiert. Sollte die Frequenz der Groundtruth-Generierung systembedingt geringfügig nicht eingehalten werden können, begünstigt die Interpolation die Datenaufzeichnung. Allerdings wird die Interpolation durch den Lokalisierungsfehler des RS beeinflusst. Da der erwartete Fehler des SUT signifikant größer als der kombinierte Fehler des RS und der Interpolation ist, kann von einer ausreichenden Generierung der Groundtruth ausgegangen werden. Bei erfolgreicher Interpolation und Bestimmung der Groundtruth für ein SUT-Datum werden entsprechend die Einträge aus den Warteschlangen entfernt.

Zur Wiederholung von Experimenten ist es sinnvoll, einen abgefahrenen Pfad mehrmals zu wiederholen. Die aufgezeichneten Positionen des RS können mithilfe von Move-Base nochmals abgefahren werden. Dazu generiert eine neue Softwarekomponente aus den aufgezeichneten Daten fortwährend Richtungs- und Geschwindigkeits-

informationen zur nächsten Pfadposition, welche das RS möglichst präzise auf dem bereits gefahrenen Pfad fahren lassen. Da die Lokalisierung des RS fehlerbehaftet ist, kann der Pfad folglich nur mit entsprechendem Fehler abgefahren werden. Um eine möglichst gleiche zeitliche Abfolge der abzufahrenden Positionen gegenüber der originalen Fahrt zu gewährleisten, ist es sinnvoll, die Richtungs- und Geschwindigkeitsinformationen während der Wiederholung dynamisch anzupassen. Dazu sollten allerdings die Umgebungsbedingungen (Hindernisse) unverändert bleiben.

### 2.2.3 Evaluation

Die Evaluation wird im Informatik-Gebäude der FU durchgeführt. Das Gebäude bietet einen ca. 60 m langen Flur und drei davon abzweigende kleinere Flure (mit 5-20 m Länge) mit ähnlichen Charakteristiken, wie zum Beispiel die Lokalisierung begünstigenden Einbuchtungen zu Büroräumen. Alle Fahrten finden in einem solchen Flur des 1. OG statt.

Die Lokalisierungsgenauigkeit des RS hängt wesentlich von der Umgebung ab. In Umgebungen mit vielen in der Karte eingetragenen Hindernissen im Sichtfeld des Roboters ist die Genauigkeit deutlich höher als in Umgebungen mit weniger bzw. gar keinen Hindernissen. Da das System in Fluren eines büroähnlichen Gebäudes eingesetzt werden soll, wird die Evaluation in einem solchen durchgeführt. Um eine Vorstellung über die Genauigkeit des RS zu erhalten, werden  $5 \times 9$  Positionen in einem Raster markiert. Der Roboter wird manuell entlang allen Positionen gesteuert, während die durch AMCL gemeldeten Positionsschätzungen aufgezeichnet werden. Dies wird zweimal wiederholt, sodass 135 Messungen entstehen. Das Ergebnis zeigt einen durchschnittlichen Positionsfehler von 6,7 cm, mit einer Standardabweichung von 4,0 cm. Der maximale Fehler lag bei 21,7 cm. In Abb. 2.3 ist zu sehen, dass 96,3 % der Fehler kleiner oder gleich 15 cm und 80,0 % kleiner oder gleich 10 cm sind. Dieser Lokalisierungsfehler ist für die meisten gängigen SUT ausreichend gering. Durch die sich wiederholenden Eigenschaften der Flure im Testgebäude gelten diese Werte für alle dortigen Flure.

Wie zuvor erwähnt, bietet das Gebäude signifikante Merkmale in den Fluren. Die sich in wenigen Metern wiederholenden Einbuchtungen zu Büroräumen bieten eine nahezu ideale Umgebung für den Lokalisierungsprozess des RS. Würde das System entlang eines merkmalslosen Flurs geführt werden, könnte die Höhe im Flur aufgrund



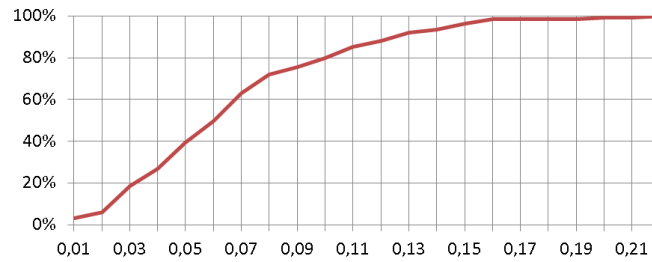


Abbildung 2.3: Die kumulative Fehlerverteilung des Referenzsystem im getesteten Flur. Die x-Achse zeigt den Positionsfehler in Metern.

der driftenden Odometrie der Basis nicht mehr akkurat abgeschätzt werden. Hier müssten folglich künstliche Hindernisse eingepflegt werden, um die Lokalisierung zu verbessern.

## 2.2.4 Anwendungsfall: Vergleich von Algorithmen

Um die Nutzungsmöglichkeiten des RS aufzuzeigen, wird ein einfacher Anwendungsfall betrachtet. Das SUT dient hier nur als Beispielsystem, daher wird dieses erst in Kapitel 3.3.2 genauer beschrieben.

Abbildung 2.4 zeigt eine beispielhafte Datenaufzeichnung mit 17 im Gebäude, hauptsächlich in geschlossenen Büroräumen, installierten Ankern. Das RS wurde über ca. 100 m entlang des Flurs geleitet (schwarze Linie), während Mitarbeiter im Gebäude arbeiteten und gelegentlich über den Flur liefen. Zwei gut bekannte Lokisierungsalgorithmen werden miteinander verglichen: Multilateration mit Non-linear Least Squares (NLLS) [109, 110] und Min-Max [111, 112]. Die geschätzten Positionen sind ebenfalls in der Abbildung zu sehen. Mit den interpolierten Daten ist nun ein numerischer Vergleich der beiden Resultate möglich. Als Metriken werden hier das arithmetische Mittel für Positionsfehler (MAE für Distanzmessungen) und der RMSE genutzt. Das SUT erreichte eine Messgenauigkeit der Distanzen von durchschnittlich 2,85 m mit einem RMSE von 4,32 m. Allerdings finden sich in seltenen Fällen Distanzfehler bis hin zu 75 m. Der Lokalisierungsfehler von NLLS liegt im Anwendungsfall bei durchschnittlich 4,49 m und bei Min-Max bei 2,05 m. Daraus lässt sich schließen, dass das RS mit einem unter vergleichbaren Bedingungen gemessenen durchschnittlichen Positionsfehler von unter 7 cm klar für derartige Anwendungsfälle ausreicht.

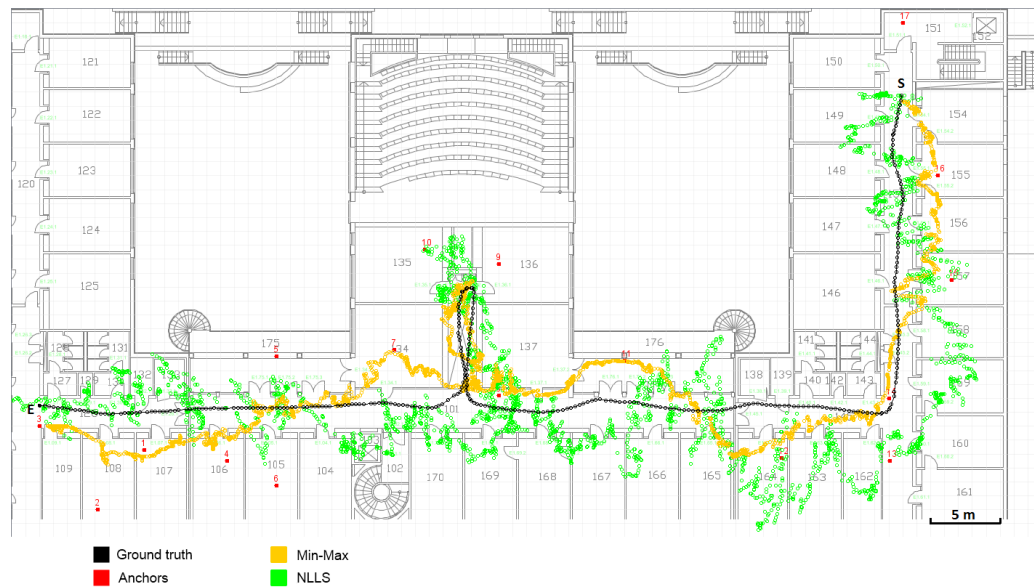


Abbildung 2.4: Positionsschätzungen im 1. OG des FU-Gebäudes. Die Startposition des gefahrenen Pfades ist mit *S*, die Endposition mit *E* gekennzeichnet.

### 2.2.5 Zusammenfassung

Es wurde ein präzises RS zur Evaluation von Lokalisierungssystemen für Gebäude präsentiert. Das System ist durch alltägliche Komponenten kostengünstig und durch quelloffene Software für Anwender leicht zu benutzen und ggf. zu erweitern. Durch die Natur des Ansatzes ist eine vorhandene und möglichst präzise Karte unbedingt erforderlich. Hier können je nach benötigter Genauigkeit vorhandene Gebäudepläne genutzt werden, die ggf. manuell editiert werden müssen, um nicht erwünschte oder nicht real vorhandene Hindernisse zu entfernen. Durch ROS-eigene Software oder manuelle Annotationen können bestehende Pläne auch erweitert oder detaillierter gemacht werden, um eine Lokalisierung auch in größeren Räumen zu erlauben. Eine neue Softwarekomponente vereinfacht die Zusammenführung von Groundtruth und SUT-Daten und erlaubt eine Experimentwiederholung durch erneutes Abfahren des Pfades.

Neben der verwendeten Karte ist die Genauigkeit des Systems abhängig von den Tiefeninformationen und der Odometrie. Zusätzliche Sensoren (bspw. Gyroskop, Laserscanner) können die Genauigkeit der Groundtruth und auch die Genauigkeit von automatisch erstellten Karten erhöhen. Das vorgestellte System lokalisiert sich selbst in Räumen kleiner oder gleich 10 m. Dies deckt sehr viele Szenarien ab. Allerdings

ist zu beachten, dass die Genauigkeit mit steigender Entfernung der sichtbaren Hindernisse abnimmt. Daher sollte der Schätzfehler des RS in neuen Szenarien separat bestimmt werden. Im Anwendungsszenario zeigt das RS einen Fehler von unter 7 cm, welcher einem Positionsfehler von bspw. 2,05 m (Min-Max) gegenübersteht. Ähnliche Lokalisierungssysteme und -algorithmen lassen sich folglich mit diesem System numerisch vergleichen.

Die vorgestellte Evaluierungsmethode ermöglicht eine kostengünstige Durchführung von physikalischen Experimenten mit hohem Messdatenanteil. Das System erlaubt durch die Automatisierung der Groundtruth Bestimmung eine leichtere Wiederholbarkeit als bspw. bei händischer Bestimmung. Bei Anwendung derartiger Methodik und mit Vorhandensein der resultierenden breiteren Datenbasis ist anzunehmen, dass Autoren statistische Größen über die Ergebnisse öfter und vollständiger nennen. Darüber hinaus kann die Methodik der Datensammlung in wissenschaftlichen Veröffentlichungen nachvollziehbar beschrieben werden, womit eine Reproduzierbarkeit der Ergebnisse unterstützt wird.

Zwar bietet das vorgestellte System eine einfache Wiederholbarkeit und eine weitgehend automatische Versuchsdurchführung, allerdings ist es dabei auf eine Etage, glatte Böden und bereits geöffnete oder sich automatisch öffnende Türen beschränkt. Auch können Glastüren meist nicht erkannt werden, wodurch hier eine Überwachung oder ausreichende Planung der Evaluation nötig wird. Die automatische Skalierung auf mehrere Etagen, andersartige Untergründe oder auch die Messung in unterschiedlicher Höhe bleiben anspruchsvoll und bisher ungelöst.

Die Messungen unterscheiden sich im Allgemeinen von einem Szenario, in dem ein Mensch das Sensorsystem bspw. am Körper durch ein Gebäude trägt. Hierbei würde das Signal in einem gewissen Bereich abgeschwächt werden (engl. *human body shadowing*, [113]). Da das SUT auf dem Roboter angebracht ist kann die Signalausbreitung als weitgehend omnidirektional betrachtet werden (sofern das SUT omnidirektionale Eigenschaften bietet). Das Problem der Abschattung beim hier verwendeten SUT wurde in einer eigenen Arbeit ebenfalls unter Verwendung des RS betrachtet [95].

---

## KAPITEL 3

---

# Virtuelles Testbed für distanzbasierte Lokalisierung

Im Folgenden wird eine neuartige Form von Testbeds vorgestellt, das Virtual Testbed for Indoor Localization (VirTIL). Dieses Kapitel basiert auf folgenden eigenen Veröffentlichungen:

- S. Schmitt, H. Will, T. Hillebrandt und M. Kyas, “A virtual indoor localization testbed for Wireless Sensor Networks”, in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 10th Annual IEEE Communications Society Conference on, Juni 2013, S. 239–241. DOI: 10.1109/SAHCN.2013.6644985
- S. Adler, S. Schmitt, H. Will, T. Hillebrandt und M. Kyas, “Virtual testbed for indoor localization”, in *Indoor Positioning and Indoor Navigation (IPIN)*, International Conference on, Okt. 2013. DOI: 10.1109/IPIN.2013.6817880

Diese Arbeit wurde teilweise bereits zuvor in einer weiteren eigenen Arbeit vorgestellt:

- S. Schmitt, “Aufbau und Evaluation eines virtuellen Testbeds zur Indoor Lokalisierung in drahtlosen Sensornetzwerken”, Masterarbeit, Freie Universität Berlin, Nov. 2013

Im Rahmen der Entwicklung von VirTIL entstanden darüber hinaus folgende eigene Arbeiten:

- D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, V. Handziski, S. Sen, F. Lemic, J. Busch, Z. Jiang, H. Zou, H. Jiang, A. Ashok, C. Xu, A. Ghose, N. Ahmed, C. Zhang, Z. Xiao, H. Wen, T. Schmid, D. Lee, M. Klepal, C.

- Beder, M. Nikodem, S. Szymczak, P. Hoffmann, L. Selavo, D. Giustiniano, V. Lenders, M. Rea, A. Marcaletti, C. Laoudias, D. Zeinalipour-Yazti, Y.-K. Tsai, T. E. Abrudan, A. Markham, A. Bestmann, R. Reimann, L. Li, C. Zhao, S. Adler, S. Schmitt, P. Lazik, N. Rajagopal, A. Rowe, V. Dentamaro, D. Colucci, P. Ambrosini, A. S. Ferraz, L. Q. L. Martins, P. Bello, A. Alvino, V. Sark, G. Pirkl und P. Hevesi, “A Realistic Evaluation and Comparison of Indoor Location Technologies: Experiences and Lessons Learned”, in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks (IPSN)*, Seattle, Washington: ACM, Apr. 2015, S. 178–189, ISBN: 978-1-4503-3475-4. DOI: 10.1145/2737095.2737726
- S. Adler, S. Schmitt, Y. Yang, Y. Zhao und M. Kyas, “FUBLoc - Accurate Range-based Indoor Localization and Tracking”, Microsoft Indoor Localization Competition Award, 13th ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN), 3. Platz (Infrastructure-based Approaches), Apr. 2014
  - F. Lemic, V. Handziski, A. Wolisz, T. Constambeys, C. Laoudias, D. Zeinalipour-Yazti, C. Panayiotou, S. Adler, S. Schmitt, H. Will, T. Hillebrandt, M. Kyas, Y. Yang und Y. Zhao, “Systematic Objective Evaluation of RF-based Indoor Localization Algorithms (EVARILOS Open Challenge: Track 3)”, Telecommunication Networks Group, Technische Universität Berlin, TKN Technical Report Series TKN-14-005, Okt. 2014. Adresse: [http://www.tkn.tu-berlin.de/fileadmin/fg112/Papers/2014/lemic14\\_open\\_challenge.pdf](http://www.tkn.tu-berlin.de/fileadmin/fg112/Papers/2014/lemic14_open_challenge.pdf)
  - S. Adler, S. Schmitt, H. Will, T. Hillebrandt und M. Kyas, “Geo-n Localization and 3CoM (3 Centers of Mass) Indoor Localization Algorithms”, Evaluation of RF-based Indoor Localization Algorithms, EVARILOS Open Challenge, 2. Platz, Mai 2014
  - F. Lemic, V. Handziski, A. Wolisz, T. Constambeys, C. Laoudias, S. Adler, S. Schmitt und Y. Yang, “Experimental evaluation of RF-based indoor localization algorithms under RF interference”, in *Localization and GNSS (ICL-GNSS), International Conference on*, Juni 2015. DOI: 10.1109/ICL-GNSS.2015.7217149

Die Grundzüge von VirTIL wurden bereits in der Dissertation von Hillebrandt und Will vorgestellt [4], welche zeitlich parallel zur Entwicklung des Testbeds abgeschlossen wurde. Die Beschreibung des Datenbankaufbaus wurde dieser Quelle entnommen und entsprechend markiert, da es sich um den selben Inhalt handelt.

## 3.1 Motivation

Eine Frage die jede Forschung an distanzbasierten und anderen Lokalisierungsmethoden beantworten muss lautet: Wie sollten entwickelte oder verbesserte Algorithmen oder Systeme getestet werden? Der Experimentaufbau benötigt oft einen hohen Einsatz von Zeit und anderen Ressourcen. Da Hardware zur funkbasierten Lokalisierung eher teuer und meist nicht aus dem Massenmarkt kommt, sind bisher nur wenige Arbeitsgruppen in der Lage, große Experimente durchzuführen. Standardtechnik, wie bspw. von Will et al. vorgestellt [118], oder die Simulation von funkbasierter Distanzmessung stellen zwar kostengünstige Möglichkeiten dar, allerdings hat dies auch große Distanzfehler und eine niedrige Messrate bzw. wenig Übereinstimmung mit der Realität zur Folge. Letzteres liegt daran, dass eine realitätsnahe Modellierung des Kanals und des Fehlermodells äußerst schwierig ist. Die Schwierigkeiten bei realen Experimenten und Simulation stellen ein Problem innerhalb der akademischen Gemeinschaft dar.

Reale Experimente können in großangelegten Feldtests wie bspw. dem Projekt L.I.N.K. des Fraunhofer IIS durchgeführt werden [119]. Hier wurde eine Testumgebung mit 1400 m<sup>2</sup> Hallen- und Laborfläche speziell für das Testen von Lokalisierungstechnologien aufgebaut. Das TKN Wireless Indoor Sensor Network Testbed (TWIST) nutzt dagegen das auch hier vorgestellte RS für einen der ersten Ansätze eine Standardisierung über Bewertungsmetriken zu schaffen [103, 120]. Ziel hierbei ist unter anderem die Berechnung eines Scores, welcher verschiedene Techniken anhand einer Metrik direkt vergleichbar macht. Innerhalb des statischen Testbeds ist eine Datenaufzeichnung in einem Bürogebäude mit durch den Benutzer steuerbaren Robotern möglich. Lemic et al. zeigen auch die Möglichkeit, eine webbasierte Analyse von Lokisierungsalgorithmen auf diesen Daten durchzuführen [121, 122]. Nader et al. stellen mit PerfLoc eine öffentliche Plattform, welche einen Vergleich von Algorithmen auf vorgegebenen Daten von Smartphones ermöglicht [123]. Auch andere Autoren stellen gesammelte Datensätze zur Verfügung, wie bspw. Angermann et al. [124].

Um den monetären und evaluationsorientierten Zielkonflikt anzugehen, ist ein einfach zu nutzendes Simulationssystem mit realitätsnahen Annahmen oder ein einfaches, kostengünstiges Setup zur Reproduzier- und Wiederholbarkeit von Experimenten nötig, welches sich im Kontext beliebiger Gebäude nutzen lässt. Das präsentierte System erlaubt sogenannte virtuelle Experimente in einem großen Datensatz experimentell

gesammelter Daten mit Groundtruth. Dies umgeht die Schwierigkeiten beim Modellieren von Fehlermodellen, die aufwändige Installation von (Referenz-)Hardware und auch das Problem der Reproduzierbarkeit. Der Ansatz eines solchen virtuellen Testbeds ist bereits aus anderen Bereichen, wie eingebetteten Systemen oder drahtlosen Sensornetzwerken, bekannt.

Die Qualität des Ergebnisses eines Lokalisierungsprozesses hängt auch bei sehr akkuraten Systemen wesentlich vom Ort ab [63]. Manche Orte können von nahezu allen Systemen gut, andere Orte können von nahezu allen Systemen nur mit großem Fehler geschätzt werden. Die Wahl der Evaluationspunkte ist daher essentiell und muss objektiv durchgeführt werden, um kein SUT zu benachteiligen. Das bedeutet, dass die Evaluation verschiedener Techniken, Systeme und Algorithmen an möglichst vielen Punkten durchgeführt werden muss. Bisherige Evaluationen und insbesondere auch Lokalisierungswettbewerbe nutzen einzelne Punkte zur Evaluation [63, 115, 117] und sind zudem aufgrund ihrer engen Rahmenbedingungen anfällig für Verzerrungen (bspw. Bevorzugung von WLAN-Lösungen durch realitätsferne Überabdeckung). Mithilfe von automatisierten RS ist es dagegen leicht möglich, ein dichtes Raster aus Evaluationspunkten zu betrachten. Dies gewährleistet eine ganzheitliche Bewertung eines SUT für konkrete Applikationskontexte. Diese Diskussion zeigt auch die Komplexität, möglichst allgemeingültige und dennoch in jedem Kontext anwendbare, vergleichbare Metriken für Systeme zu definieren.

Das im Folgenden vorgestellte virtuelle Testbed verfolgt daher zweierlei Ziele: Zum einen wird eine vergleichsweise große Datenbasis generiert. Diese bildet die Eigenschaften der Umgebung und des SUT in möglichst großer und feingranularer Fläche ab. Hierbei soll es sich um real gesammelte Daten aus einer bestimmten Umgebung handeln. Zwar existieren physikalische Modelle und durch Erfahrung unterstützte Vorhersagen über die Ausbreitung eines Signals und eine sinnvolle Ankerplatzierung, allerdings können Umgebungen in der Realität durchaus komplexer sein, als zunächst angenommen. Das Testbed soll hier über unerwartete physikalische Effekte und deren Einfluss auf die Lokalisierung aufklären. Während Lokisierungsalgorithmen bereits auf dieser Datenbasis evaluiert werden können, ist es zum anderen häufig von Interesse, einen dedizierten Lauf durch die Umgebung als Testdatensatz zu betrachten. So ist es vorstellbar, den Weg eines sich natürlich und damit nicht perfekt geradeaus sowie nicht in unmittelbarer Nähe zur Wand laufenden Menschen zu betrachten. Ein solcher Lauf kann daher aus der gesammelten Datenmenge extrahiert werden. Auch können die gewünschten Anker aus der bei Durchführung der Datensammlung instal-

lierten Ankermenge ausgewählt werden. Dadurch entsteht ein sogenanntes virtuelles Experiment, welches zwar nicht in der Konstellation real stattgefunden hat, jedoch ausschließlich reale Messungen enthält. Zudem ist durch die virtuelle Ankerselektion eine gezielte Untersuchung des Einflusses der Abdeckung möglich. Die folgenden Kapitel zeigen, dass diese Herangehensweise valide Aussagen über einen Algorithmus erlaubt.

Um diese Möglichkeiten zu bieten, benötigt ein solches System mehrere Komponenten. Der aufwändigste Teil ist die Sammlung einer großen Datenbasis. Diese sollte mindestens dichte Datenpunkte in einem Raster im verfügbaren Raum umfassen. Ein Datenpunkt umfasst dabei die Groundtruth-Informationen, wie Ort und Zeitpunkt, und das gemessene Datum. Im Falle dieser Arbeit ist dies eine Distanzmessung zu bekannten Orten (Ankern). Allerdings können hier nahezu beliebige andere SUT genutzt werden. Zur Bestimmung der Groundtruth wird ein Referenzsystem benötigt, um den tatsächlichen Ort einer Messung möglichst genau zu bestimmen. Kapitel 2.2 präsentierte zuvor ein solches System. Zusätzlich ist ein adäquates Mittel zur graphischen Datenanalyse und Extraktion nötig.

## 3.2 Aufbau des virtuellen Testbeds

Abbildung 3.1 zeigt den generellen Aufbau des virtuellen Testbeds. Das in Kapitel 2.2 vorgestellte RS wird hier genutzt. Dies trägt das SUT und einen Laptop zur Fusion der gesammelten Daten des SUT und der entsprechenden Groundtruth. Weiterhin werden die Daten in einer Datenbank gesammelt, welche durch eine webbasierte Benutzerschnittstelle bedient wird. In dieser kann der Benutzer ein virtuelles Experiment definieren und die Datensätze zur weiteren Analyse exportieren.

### 3.2.1 Vollständige Abdeckung

Das Problem der vollständigen Abdeckung wird in der Literatur unterschiedlich behandelt. Choset et al. beschreiben verschiedene Methoden der zellbasierten Repräsentation von Karten [125]. Dabei wird immer zwischen freien, partiell belegten, belegten und indirekt auch unbekanntem Zellen unterschieden. Unter einer Abdeckung wird meist das Areal beschrieben, welches durch die Grundfläche des Roboters befah-



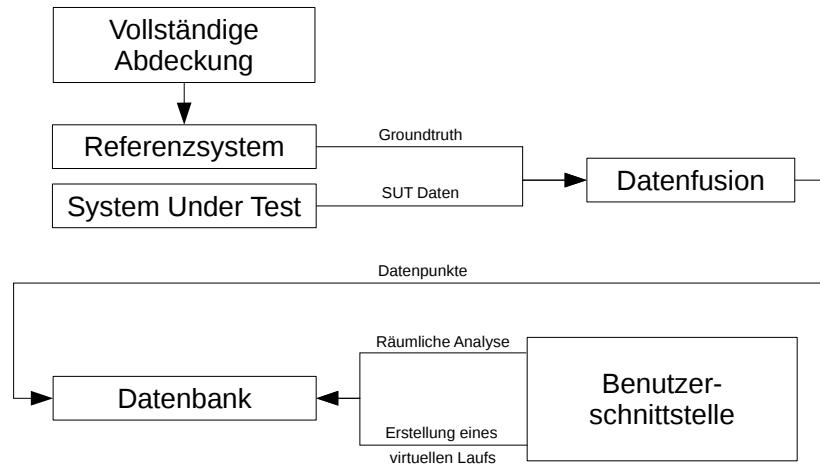


Abbildung 3.1: Komponenten und Informationsfluss des virtuellen Testbeds.

ren wurde. Die Literatur zeigt unterschiedliche Metriken für die Klassifizierung einer abgedeckten Zelle und damit unterschiedliche Definitionen über die vollständige Abdeckung. Diese Arbeit versteht unter dem Begriff das Befahren aller erreichbaren und freien Zellen. Der Mittelpunkt des Roboters soll dabei mit einer Unsicherheit in der Größenordnung seines Lokalisierungsfehlers den Mittelpunkt einer Zelle befahren, sofern der Radius des Roboters eine kollisionsfreie Traversierung erlaubt. Das Problem ähnelt dem Traveling-Salesman-Problem. Die Startposition und die anzufahrenden freien Knoten (Gitterpunkte bzw. Zellen) sind prinzipiell bekannt. Zwar registriert der Roboter in unmittelbarer Umgebung sämtliche Hindernisse, jedoch enthält die Karte keinerlei statischen (geschlossene Türen, Schränke, Pflanzen etc.) und erst recht keine dynamischen Hindernisse (Menschen, sich öffnende Türen etc.). Dadurch kann zum Startzeitpunkt keine ideale Lösung berechnet werden. Es existieren einige komplexere Algorithmen zur vollständigen Abdeckung [126–130] mit Potential zur Optimierung der Traversierung. Die nachfolgend dargestellte einfache Implementierung entstand im Rahmen meiner Masterarbeit und genügt den Anforderungen.

Genutzt wird eine eigene sehr simple Implementierung zur Traversierung des verfügbaren Raumes. Eine Pfadplanungskomponente berechnet, aufgrund der Position des Roboters in der Karte und des bereits abgedeckten Areal, abzufahrende Wegpunkte im Gitter. Um möglichst wenige Wegpunkte vergeblich zu berechnen, werden nur eine begrenzte Menge an abzufahrenden Wegpunkten gesucht, welche lokal bekannte Hindernisse beachten, bspw. im Radius von 2 m. Dazu wird eine einfache Implementierung der Breitensuche von Dijkstra [131] genutzt. Der Algorithmus sucht Wegpunkte

in Abhängigkeit von der Entfernung und bereits traversiertem Raum, welche noch nicht traversiert wurden. Dabei wird freier Raum nahe bereits traversiertem Raum bevorzugt und Mehrfachabdeckung möglichst vermieden, sodass ein dicht gepackter zickzack-ähnlicher Pfad entsteht. Sobald alle Wegpunkte abgefahren wurden, wird ein neues Pfadsegment generiert.

Aufgrund der Ungenauigkeiten der Lokalisierungskomponente des RS entsteht mit der beschriebenen Implementierung in der Praxis ein chaotischerer Pfad. Dies ist in Form der Groundtruth in nachfolgenden Abbildungen ersichtlich. Der wesentliche Unterschied besteht in der Anzahl der Drehungen des RS und der dafür benötigten Zeit. Der Algorithmus wurde später durch L. Zech et al. an der Freien Universität Berlin weiterentwickelt [132]. Da die Datensammlung das übergeordnete Ziel darstellt, genügt die beschriebene Traversierung den Anforderungen.

### 3.2.2 Datenformat und Datenbank

Zur Speicherung der Daten wird eine PostgreSQL Datenbank<sup>1</sup> auf einem dedizierten Server genutzt. Die Daten können wahlweise über eine Hypertext Transfer Protocol (HTTP) Representational State Transfer (REST) Schnittstelle während der Datensammlung an die Datenbank geschickt, oder zunächst auf dem Referenzsystem zwischengespeichert und anschließend in die Datenbank importiert werden. Zur Effizienzsteigerung bei räumlichen, koordinatenbasierten Anfragen an die Datenbank wird PostGIS<sup>2</sup> verwendet. Das Datenformat ist auf die Speicherung von ToF-basierten Daten ausgelegt, kann aber leicht auf andere Lokalisierungstechniken oder um andere relevante Daten, wie bspw. Luftdruck, erweitert werden.

Zur Zwischenspeicherung und zum Export der Daten wird das Network Common Data Form (NetCDF)<sup>3</sup> verwendet. Die Daten können so direkt mit anderen Analysetools (bspw. Matlab, Octave und R) ausgewertet werden. Im Wesentlichen enthält eine solche Datei neben beschreibenden Daten (Ankerbezeichner, -positionen, Anzahl Geräte, Ort etc.) eine Matrix mit folgenden Spalten:

**Zeitstempel** in Unix Zeit in Millisekunden

**Initiator** der Funkdistanzmessung

---

<sup>1</sup><https://www.postgresql.org/>

<sup>2</sup><https://postgis.net/>

<sup>3</sup><https://www.unidata.ucar.edu/software/netcdf/>

**Anker** zu dem die Funkdistanzmessung durchgeführt wurde

**Distanz** welche gemessen wurde, negativer Fehlercode bei Timeout

**Groundtruth** Position des RS zum Zeitpunkt der Messung

Die Definition des Formats ist damit auf eine Distanzmessung zwischen zwei Akteuren angepasst, kann jedoch leicht um weitere Spalten oder Matrizen erweitert werden. Auch eine Multi-Hop Lokalisierung ist hierdurch abgedeckt. Dies ist eine vereinfachte Beschreibung; aus Effizienzgründen werden auch berechenbare Größen wie der Distanzfehler, aus Gründen der Nachvollziehbarkeit auch Hardware-IDs gespeichert. Um die Messungen während des Experiments ganzheitlich zu erfassen, wird auch das Nichtzustandekommen einer Distanzmessung (bspw. aufgrund von Funkabschattung) gespeichert .

”Beim Datenbankdesign wurde [...] auf eine möglichst performante Abfragemöglichkeit der eigentlichen Messdaten Rücksicht genommen. So wurde die Tabelle mit den Messdaten komplett denormalisiert, so dass bei allen Abfragen auf die eigentlichen Pfaddaten keinerlei Joins nötig sind. Für statistische Zwecke wurden neben den gemessenen Entfernungen und den Groundtruth-Daten zusätzlich noch der sich daraus ergebende Messfehler, die MAC-Adressen der beiden beteiligten Knoten und eine Experimentkennung sowie eine Gebäudenummer mit erhoben. Mit diesen Daten ist es möglich, in den vorhandenen Daten nach Knoten zu suchen, die einen bestimmten Bias aufweisen oder zwei verschiedene Messsysteme zu vergleichen.

Die wesentlichen Entitäten haben dabei die folgenden Hintergründe:

**location** Der Ort der Messung mit Bezeichnung und Textbeschreibung.  
[...]

**map** Eine Karte als Bitmap. Die konkrete Bitmap ist dabei als Pfad des lokalen Dateisystems angegeben. Zusätzlich sind diverse Kartenparameter, die für eine grafische Darstellung benötigt werden, gespeichert - insbesondere sind dies Skalierungsfaktoren, Kartenursprung und Drehung. Die Drehung wird benötigt, wenn die Bitmap ein Gebäudeplan ist, der den Kartennorden nicht an der Oberseite hat.

**experiment** Ein Experiment ist die grundlegende Entität für eine Datenerhebung in einem Gebäude (einer *location*) mit feststehender An-

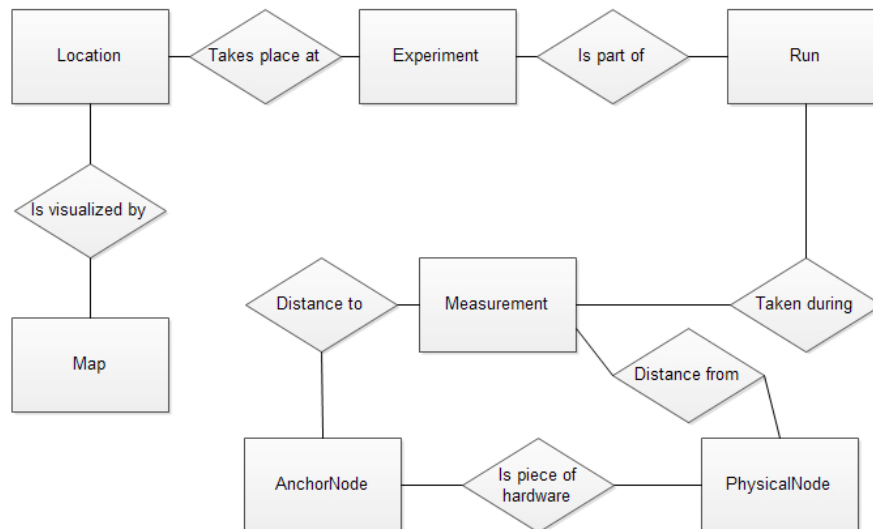


Abbildung 3.2: Datenbankrelation des virtuellen Testbeds. Übernommen aus der Dissertation von Hillebrandt und Will [4].

kerkonfiguration.

**run** Ein Lauf ist die Entität für eine kontinuierliche Aufnahme von Messdaten.

**physical\_node** Die Bezeichnung eines konkreten Knotens inkl. verwendeter Hardware und MAC-Adresse.

**anchor\_node** Ein Ankerknoten mit seiner Position.

**measurement** Die wesentliche Entität, in welcher neben der gemessenen Entfernung zu genau einem Ankerknoten, der Verweis auf diesen Knoten, der Groundtruth-Position und der konkrete Knoten gespeichert werden.

Die Relationen sind in Abb. 3.2 dargestellt.” [4]

### 3.2.3 Benutzerschnittstelle

Die Datenbank kann über eine webbasierte Schnittstelle bedient werden. Dem Anwender wird eine skalierbare Karte der Experimentumgebung mit zur Verfügung ste-

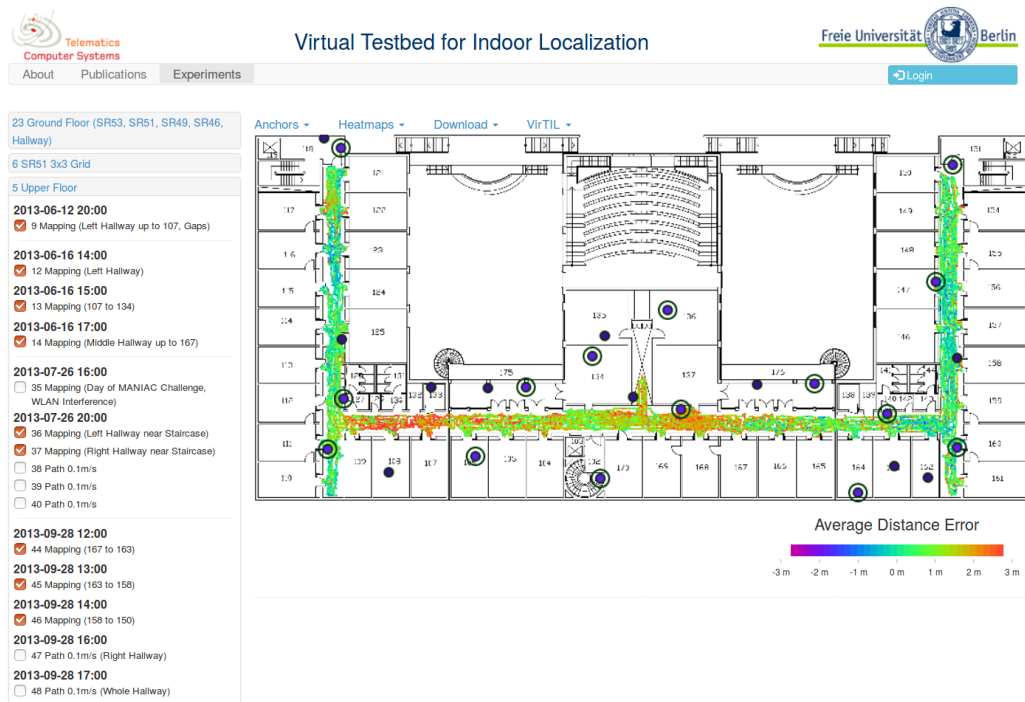


Abbildung 3.3: Durchschnittlicher räumlicher Distanzfehler des SUT (über alle aktivierten Anker). Die Zellgröße beträgt 0,1 m.

henden Anker zeigt. Mithilfe dieser kann in den zur Verfügung gestellten Daten eine virtuelle Teilmenge (ein Lauf) über das Zeichnen eines Pfades definiert werden. Danach werden die Datenpunkte entlang diesen Pfades als Export zur Verfügung gestellt. Aufgrund des verhältnismäßig langsamen RS sind im Allgemeinen mehrere Datenpunkte an einer Position verfügbar. Um ein möglichst realistisches Szenario zu erhalten, werden in dem Fall zufällige Datenpunkte gezogen. Zur visuellen Analyse der gesammelten Daten und installierter Lokalisierungsalgorithmen erlaubt das Testbed zudem, eine farbkodierte, Heatmap-artige Darstellung des Distanz- bzw. Lokalisierungsfehlers. Datenpunkte auf definierbarer Fläche (bspw.  $0,1 \times 0,1 \text{ m}^2$ ) werden zu einem zusammengefasst und zeigen eine konfigurierbare Farbe entsprechend der Metrik. Abbildung 3.3 zeigt die Webschnittstelle für ein beispielhaftes Szenario. Zur Analyse der Daten stehen einige Bewertungsmetriken zur Verfügung, weitere können leicht ergänzt werden: Datendichte, durchschnittlicher Distanzfehler, minimaler und maximaler Distanzfehler, Anzahl durchschnittlicher erreichter Anker, Rate der erfolgreichen Distanzmessungen, Varianz der Distanz sowie des Distanzfehlers und RMSE.

## 3.3 Vorstellung der Datensätze

Nun wird das verwendete SUT und die zur späteren Analyse genutzten Datensätze vorgestellt.

### 3.3.1 Konfiguration des Referenzsystems

Zur Generierung von Groundtruth-Daten und zum Tragen sowie Bewegen des SUT wird das in Kapitel 2.2 vorgestellte RS genutzt. Dies bietet einen Lokalisierungsfehler von ca. 7 cm, welcher deutlich niedriger als beim im folgenden Kapitel vorgestellten SUT ist.

Es werden sowohl einzelne Pfade (bspw. quer durch den Flur) als auch eine feingranulare Flächenabdeckung generiert. Im ersten Fall kann das System sowohl automatisch als auch manuell dirigiert werden. Die Flächenabdeckung des zur Verfügung stehenden Raums wird weitgehend automatisch durchgeführt. Das System versucht alle anfahrbaren Positionen in einem dichten Raster abzudecken. Der Rasterabstand beträgt dabei 20 cm. Die maximale Geschwindigkeit ist in beiden Fällen auf 10 cm/s begrenzt. Allerdings kann die Geschwindigkeit im Falle von Drehungen oder Hindernisumfahrungen signifikant geringer sein, sodass hier mehr Daten aufgezeichnet werden.

Das SUT generiert unabhängig vom Zustand des Roboters vier Distanzmessungen zu allen installierten Ankern pro Sekunde. Wenn eine Distanzmessung zu einem Anker länger als 10 ms andauert, wird diese als gescheitert angesehen. Bei maximal 25 Ankern dauert eine Messrunde folglich maximal 250 ms. In dieser Zeitspanne fährt das RS maximal 2,5 cm weit. Groundtruth-Daten werden unabhängig vom SUT mit 10 Hz, sprich alle 1 cm, gesammelt. Die Fusion der beiden Datenströme wird in Kapitel 2.2.2 beschrieben. Einer Messrunde wird eine Groundtruth-Position und ein Zeitstempel zugewiesen. Der dadurch eingeführte Fehler von maximal 2,5 cm ist vernachlässigbar und liegt deutlich unterhalb der Genauigkeit des RS.

### 3.3.2 Modular Sensor Board

Das SUT ist durch ein modifiziertes Modular Sensor Board (MSB) A2 [133] gegeben, welches während des FeuerWhere Projekts an der Freien Universität Berlin entwi-

ckelt wurde [134]. Abbildung 3.4 zeigt die sogenannte PTTU mit GPS-Antenne. Jeder Sensorknoten ist mit einem Transceiver von Nanotron, dem nanoPAN 5375 [135] zur Distanzmessung ausgestattet. Bei Experimentdurchführung wurde die GPS-Antenne entfernt, sodass lediglich eine Monopolantenne zur Distanzmessung am Gerät installiert war.

”[...] Dieser ermöglicht den Sensorknoten die Abstände zu benachbarten Knoten mittels ToF im 2,4 GHz Frequenzband zu bestimmen. Das eingesetzte Protokoll ist standardisiert unter ISO/IEC 24730-5:2010 und verwendet als Modulationsverfahren ein Chirp Spread Spectrum (CSS) mit einer Bandbreite von 80 MHz. Dies soll robust gegen schmal- und breitbandige Signalstörungen sowie gegen Mehrwegeempfang sein. Die exakte Methode, die zur Ermittlung einer Distanz eingesetzt wird, nennt sich Symmetrical Double-Sided Two Way Ranging (SDS-TWR). Bei diesem Verfahren ermittelt jeder der beteiligten Knoten mittels Umlaufzeitmessung eine Distanz zum jeweiligen anderen Knoten, die finale Distanz ergibt sich durch Durchschnittsbildung beider Messungen. [...]” [4]

Die weiteren Charakteristiken des Funkchips sind proprietär und werden hier bewusst nicht weiter beachtet. Die Hardware wird somit als unveränderliche Blackbox betrachtet, welche Distanzen in Metern misst. Aufgrund der technischen Eigenschaften des Messverfahrens werden Distanzmessung sequentiell durchgeführt, um eine Überlagerung auszuschließen.

Die Hardware erreicht einen durchschnittlichen Distanzfehler von 1,95 m und einen RMSE von 3,04 m, in seltenen Fällen kommen Distanzfehler bis zu 75 m vor. Auch sind Messungen mit negativem Distanzfehler möglich (gemessene Distanzen sind zu kurz). Da die Hardware als Blackbox betrachtet wird, ist die genaue Ursache nicht bekannt. Es lässt sich eine fehlerbehaftete Kalibrierung des SUT vermuten.

Während der Sichtung gesammelter Daten fiel eine zuvor unbekannte Winkelabhängigkeit des SUT auf. Die relative Ausrichtung zwischen zwei messenden Knoten beeinflusst deren Distanzmessung. Die statischen Anker werden derart positioniert, dass dieser Effekt möglichst wenig Einfluss im ausgeleuchteten Areal hat. Ein weiterer nicht beachteter Aspekt ist die Abhängigkeit von Messungen von Umgebungsparametern, wie bspw. Interferenz durch WLAN oder Bluetooth, und Einfluss durch sich potentiell verändernde Luftfeuchtigkeit oder Temperatur. Durchgeführte Datenaufzeichnungen



Abbildung 3.4: Eine PTTU mit Gehäuse und angeschlossener GPS-Antenne.

zu verschiedenen Zeiten bringen im Detail unterschiedliche Ergebnisse hervor. Daher sind die Daten als Momentaufnahme zu verstehen.

Zur Ortsbestimmung der Anker des SUT während des Aufbaus wird ein kostengünstiges Laser-Entfernungsmessgerät der Firma Bosch genutzt. Es wird auf einen Zentimeter-genauen Aufbau geachtet. Die Daten werden vor Ort durch mehrere Personen und bei der direkt anschließenden visuellen Auswertung über die Benutzerschnittstelle auf Plausibilität geprüft.

### 3.3.3 Raumszenario

Das erste Szenario nutzt einen  $6,90\text{ m} \times 7,40\text{ m}$  großen Seminarraum im Gebäude der FU. Abbildung 3.5 zeigt die Geometrie des Raumes und die Ankerverteilung in einem  $3 \times 3$  Raster. Tische und Stühle wurden an den Rand des Raumes geschoben. Acht Anker wurden auf ebendiesen Tischen installiert. Der zentrale Anker wurde an einer Aufhängung an der Decke befestigt, auf gleicher Höhe wie die übrigen, sodass kein zentrales Datenloch entsteht. Der Höhenunterschied zwischen mobilem Knoten auf dem RS und den statischen Ankern liegt im Bereich von wenigen Zentimetern. Aufgrund des höheren Distanzfehlers des SUT kann dieser vernachlässigt werden.



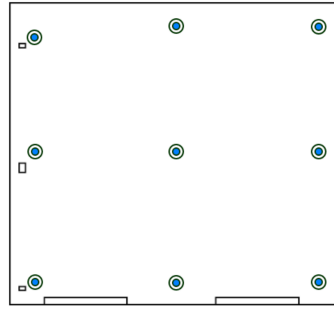


Abbildung 3.5: Seminarraum in der ersten Etage des FU-Gebäudes mit installierten Ankerpositionen (blau).

Die Aufzeichnung erfolgte am Tag bei gemäßigter Raumtemperatur. Während der Aufzeichnung befanden sich einzelne Menschen am Rande des Raumes, jedoch nie zwischen den Geräten. Es entstanden insgesamt 945.531 Distanzmessungen.

Dieser Experimentaufbau wird zwar oft in der Literatur genutzt, ist jedoch nur von geringem Nutzen zur Evaluation von Systemen und Algorithmen, da sich alle Anker zu jeder Zeit in Sicht (Line of Sight (LOS)) befinden und keine NLOS-Bedingungen entstehen. Allerdings zeigt ein solches Experiment die Limitierungen eines Systems, Algorithmus oder gar des RS auf einzigartige Weise.

### 3.3.4 Gebäudeszenario

Das zweite Szenario nutzt den Flur der zweiten Etage im Gebäude der FU. Die Abmessungen des Gebäudes betragen ca.  $70\text{ m} \times 30\text{ m}$ . Es handelt sich um ein ca. 20 Jahre altes büroartiges Gebäude, welches im Wesentlichen aus Stell-, Beton- und zwei Feuerschutzwänden besteht. Das Gebäude besitzt langgezogene Flure, Büroräume und zwei Innenhöfe. Die Räume besitzen weitgehend hölzerne Türen und schmale Sichtfenster vom Boden über die gesamte Höhe der Tür direkt daneben. Die Türen sind durch Einbuchtungen vom Flur abgesetzt. Die Büroräume sind durch eine tragende Betonwand vom Flur getrennt. Abbildung 3.6 zeigt die Geometrie des Gebäudes und die Ankerverteilung. Gegenüber der Räume 104, 105, 106 und 165, 166, 167 befinden sich gläserne Fenster und Türen mit eingelassenem Sicherheitsdraht mit ungefährem Gitterabstand von 1 cm. Zwischen den Räumen 106, 107 und auch zwischen 164, 165 befinden sich stahlverstärkte Feuerschutzwände und -türen, in der Abbildung durch dicke Wände ersichtlich. Es wird erwartet, dass beide Charakteristika die Signalaus-

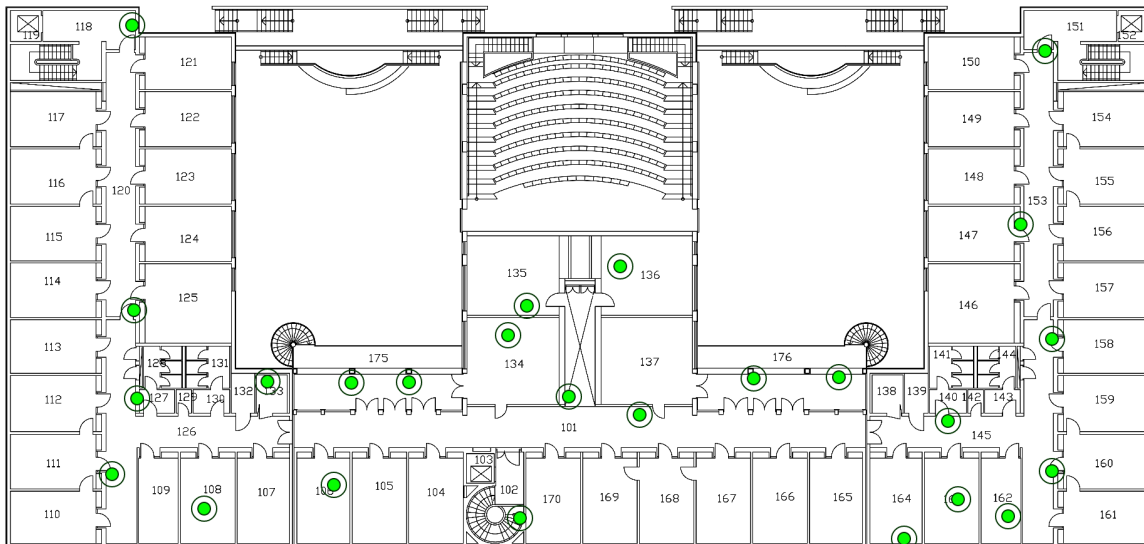


Abbildung 3.6: Zweite Etage des FU-Gebäudes mit installierten Ankerpositionen (grün).

breitung beeinflussen, da es sich hierbei um typische NLOS-Situationen handelt. An beiden Enden des horizontalen Flurs befindet sich ein ca.  $4\text{ m}^2$  großer Spiegel an der Wand montiert. Durch die Zusammensetzung und äußerst glatte Oberfläche könnten diese die Signalausbreitung ebenfalls besonders beeinflussen. Bei den Stellwänden wird im Wesentlichen ein ähnlicher, jedoch deutlich weniger ausgeprägter Einfluss wie bei den Betonwänden erwartet.

Es wurden insgesamt 25 statische Anker in zugängliche Büros und im Flur installiert. Es wurde darauf geachtet, dass insbesondere in den Büros keine Gegenstände die Sichtverbindung offensichtlich beeinträchtigen. Ziel war eine grundsätzliche und teilweise redundante Ausleuchtung des Flurs unter mehrheitlich NLOS-Bedingungen. Der Höhenunterschied zwischen mobilem Knoten auf dem RS und den statischen Ankern liegt im Bereich von wenigen Zentimetern. Aufgrund des höheren Distanzfehlers des SUT kann dieser vernachlässigt werden. Die Aufzeichnung erfolgte am Tag und auch bei Nacht bei gemäßigter Raumtemperatur. Während der Aufzeichnung befanden sich Menschen im Gebäude. Diese verdeckten sporadisch die direkte Sichtverbindung zwischen den Geräten. Es wurde darauf geachtet, dass die Aufzeichnung nicht durch bspw. stehenbleibende Menschen verzerrt wird. Es entstanden an vier Tagen insgesamt 2.863.675 Distanzmessungen auf ca.  $300\text{ m}^2$ .

Zu erwarten ist eine Positionsichte von  $400/\text{m}^2$  (bei  $10\text{ cm}$  Rasterabstand, einer maximalen Geschwindigkeit von  $10\text{ cm/s}$  und  $4\text{ Datenwerte/s}$ ). Die durchschnittliche

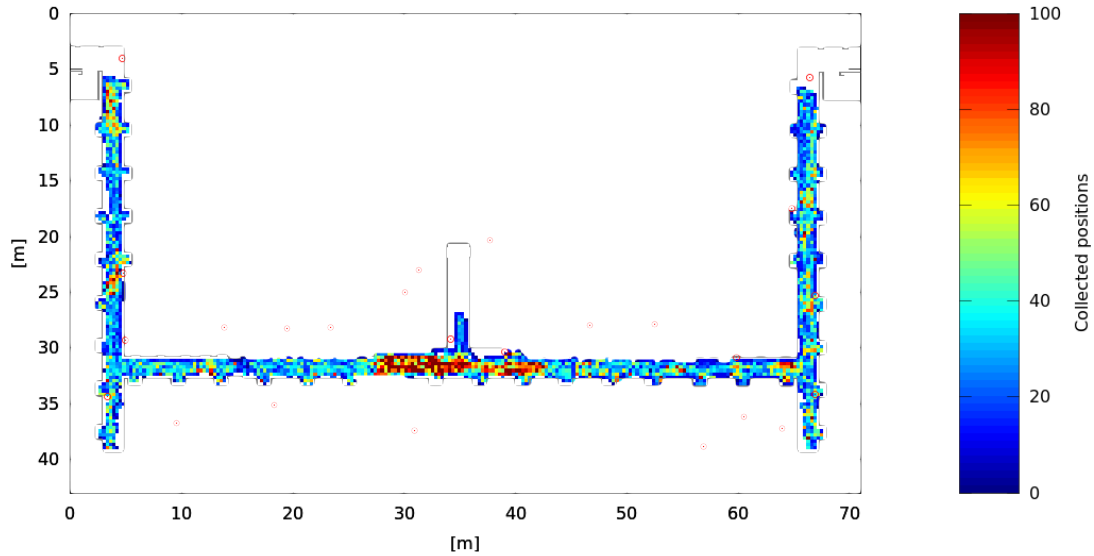


Abbildung 3.7: Dichte der gesammelten Daten. Die Zellgröße beträgt 0,3 m.

Positionsichte der Aufzeichnung liegt bei  $424,21/\text{m}^2$ , also unwesentlich höher. Die Differenz entsteht durch das nicht ganz optimale Abdeckungsschema des RS. Manche Areale im Flur waren durch Pflanzen oder Schränke nicht befahrbar. Auch der Raum in unmittelbarer Nähe zur Wand kann durch das RS nicht abgedeckt werden. Solche Areale erscheinen als weiße Gebiete in den Abbildungen. Abbildung 3.7 zeigt die Dichte der Messungen graphisch. Der mittlere Bereich wurde mehrfach durch das RS befahren und zeigt daher eine signifikant höhere Dichte.

Abbildung 3.8 zeigt den durchschnittlichen Distanzfehler des SUT. Hier erreicht der Transceiver den schon zuvor berichteten durchschnittlichen Distanzfehler von 1,95 m und einen RMSE von 3,04 m. Der maximal beobachtete Fehler lag bei 75 m, allerdings in sehr seltenen Fällen. Wie auch bereits erwähnt, sind Distanzwerte kleiner als der wahre Abstand möglich. Die meisten Kanalmodelle erklären das Auftreten eines solchen Phänomens nicht. Es lässt sich vermuten, dass es sich hier um eine fehlerbehaftete Kalibrierung handelt. Es könnte auch Teil der Arbeitsweise des SUT, bspw. bei signifikanter Interferenz, sein. Besonders im mittleren Teil des Gebäudes finden sich deutlich und konzentriert negative Werte. Es lässt sich vermuten, dass zur Zeit der Aufzeichnung in diesem Gebiet erhöhter WLAN-Verkehr durch eine Veranstaltung im Untergeschoss des Gebäudes auftrat.

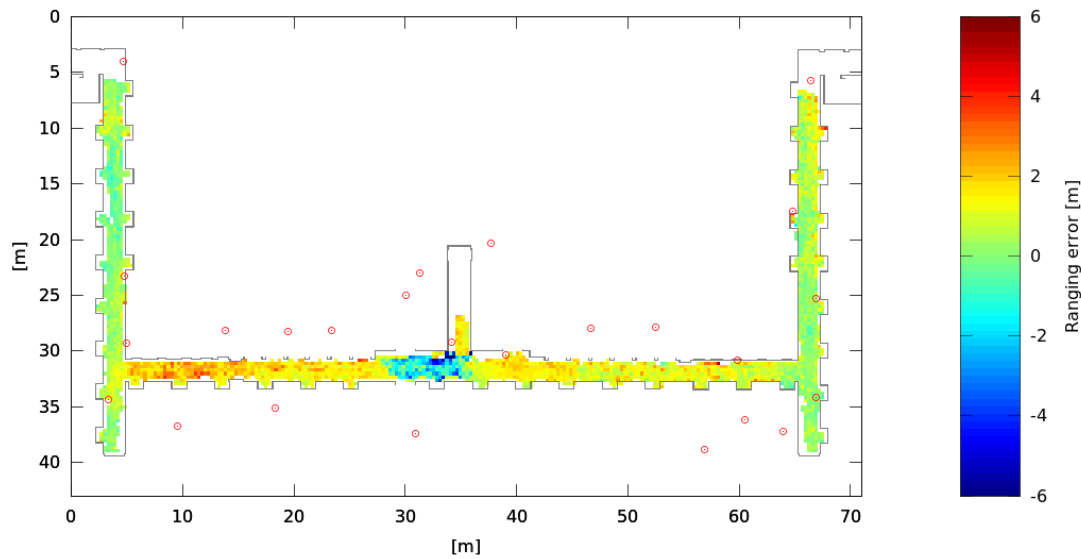


Abbildung 3.8: Räumlicher Distanzfehler des SUT. Die Zellgröße beträgt 0,3 m.

## 3.4 Evaluation

Um den Ansatz zu evaluieren, wird ein real gefahrener Pfad durch eine Büroumgebung und darauf gesammelte Daten genutzt. In dieser Evaluation werden die so gesammelten Distanzdaten und die darauf aufbauende Lokalisierung mit den entlang des Pfades extrahierten Distanzdaten und einer darauf aufbauenden Lokalisierung aus der Datenbank verglichen. Letztere entspricht einem sog. virtuellen Pfad.

### 3.4.1 Machbarkeitsnachweis

Für einen ersten Machbarkeitsnachweis wird der im Kapitel 3.3.3 vorgestellte Datensatz des Raumszenarios genutzt. Im ersten Lauf wird das RS manuell gesteuert (siehe Abb. 3.9b). Da real gesammelte Daten genutzt werden, wird dieser als *realer Lauf* bezeichnet. Die Flächenabdeckung ist in Abb. 3.9a dargestellt und produzierte 308.700 Distanzmessungen. Diese werden, wie oben beschrieben, in der Datenbank vorgehalten.

Zur Groundtruth-Position eines jeden Datenpunkts aus dem realen Lauf wird nun ein Datum in der Datenbank mit 5 cm Radius um die bekannte Groundtruth-Position gesucht. Falls hier kein Datum gefunden werden kann, wird mit dem nächsten Da-

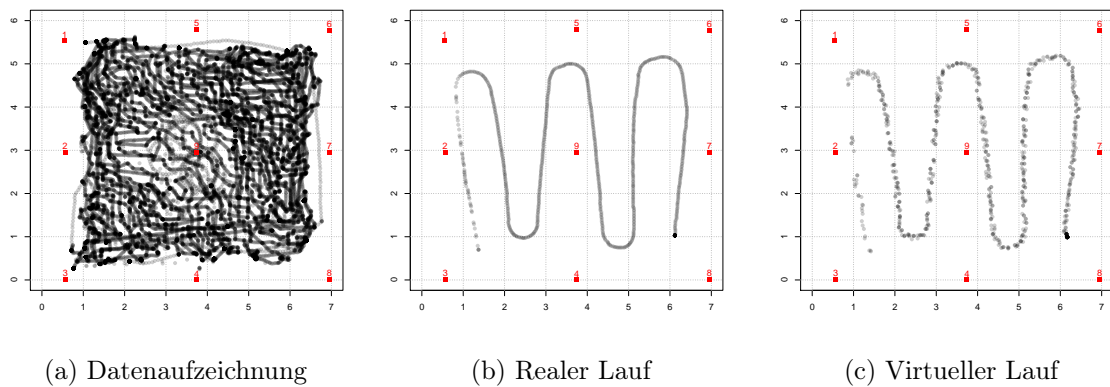


Abbildung 3.9: Datenaufzeichnungen und extrahierter virtueller Lauf der Machbarkeitsstudie. Die Einheit der Achsen ist Meter.

tenpunkt des realen Laufs fortgefahren. Falls ein Datum oder mehr Daten gefunden werden, wird zufällig eines ausgewählt, und dem virtuellen Lauf hinzugefügt. Der so erstellte virtuelle Lauf kann nun mit dem realen Lauf verglichen werden.

### Ergebnisse der Distanzmessungen

Als Beweis der Machbarkeit eines virtuellen Testbeds, wird eine starke Ähnlichkeit zwischen den Eigenschaften des virtuellen und des realen Laufs erwartet. Eine Eigenschaft ist die räumliche Form des Pfades. Abbildung 3.9b zeigt den realen, Abb. 3.9c den virtuellen Lauf. Es ist ersichtlich, dass beide Pfade bis zu einem gewissen Grad ähnlich sind. Die Abweichung kann dadurch erklärt werden, dass nicht für jede gesuchte Position ein Datum gefunden werden konnte. Das Rauschen des Pfades ist durch das Auswählen eines zufälligen Datums im Radius zu erklären. Eine weitere Eigenschaft ist die Definition des Suchraums. Hier wurde ein kreisförmiger Suchraum wie oben geschildert genutzt. Denkbar wären sowohl andere Formen, als auch die Einschränkung gewisser Eigenschaften der zu suchenden Datenpunkte (möglichst aufsteigende Zeitstempel, in Richtung des Pfades ohne Rückspringen etc.). Weitere Parameter können in Tabelle 3.1 verglichen werden. Diese zeigt, dass nicht für jede Position des realen Laufs eine entsprechende virtuelle Position gefunden werden konnte. Das hat jedoch keinen Einfluss auf die beiden anderen Werte. Die durchschnittlich erreichte Ankeranzahl zeigt, mit wie vielen Anker im Schnitt eine Distanzmessung erfolgreich war. Sie ist im virtuellen Lauf geringfügig geringer. Dies kann durch unterschiedliche Umgebungsbedingungen und eine andere Pfadausrichtung während der

Tabelle 3.1: Parameter des realen und virtuellen Laufs

	Realer Pfad	Virtueller Pfad
Erfolgreiche Messungen	6.453	6.363
Durchschnittlich erreichte Anker	8,21	8,01
Durchschnittlicher Distanzfehler	1,82 m	1,87 m

Aufzeichnung der Datenbasis erklärt werden. Die durchschnittliche Differenz des Distanzfehlers liegt bei nur 5 cm (2,75%). Für das Konzept des virtuellen Testbeds ist dies sowohl erwartet als auch vielversprechend.

Ein weitere wichtiger Vergleichsparameter ist die Fehlerverteilung der Distanzmessungen und auch die Verteilung des durchschnittlichen Fehlers pro Anker. Beide Verteilungen sollten eng miteinander verwandt sein, sodass eine Unterscheidung des virtuellen und realen Laufs schwierig ist. Die Fehlerverteilungen der Messungen sind in Abb. 3.10a für den realen und in Abb. 3.10b für den virtuellen Pfad dargestellt. Da die Fehlerverteilung eines Messsystems einen großen Einfluss auf das Lokalisierungsergebnis hat, ist es äußerst wichtig, dass die Verteilung des virtuellen Laufs sehr stark jener des realen Laufs gleicht. Es wird ein KS-Test durchgeführt, um herauszufinden, ob beide Verteilungen zur selben Verteilungsfunktion gehören. Da dieser sehr sensibel auf große Datenmengen reagiert, werden 100 Distanzfehler aus dem virtuellen und dem realen Lauf für den Test ausgewählt. Der Test wird mehrere Male wiederholt. Alle Tests legen eine Annahme der  $H_0$ -Hypothese nahe, was eine Übereinstimmung beider Verteilungen bedeutet. Die p-Werte liegen bei 0,682, was die Annahme bestätigt.

Abbildung 3.11 vergleicht den durchschnittlichen Fehler pro Anker. Die Ergebnisse zeigen einen signifikanten Unterschied für einige der Anker. Diese lassen sich nur schwer erklären. Es lässt sich vermuten, dass hier leichte Änderungen in den Umgebungsbedingungen zwischen beiden Datensätzen verantwortlich sind. Der Unterschied ist in Relation zum zu lösenden Problem zu setzen. Beim Bewerten von Lokisierungsalgorithmen wird hier kein Problem erwartet, jedoch könnten Algorithmen zur Ankerselektion von unterschiedlichem Verhalten der Datensätze negativ beeinflusst werden.

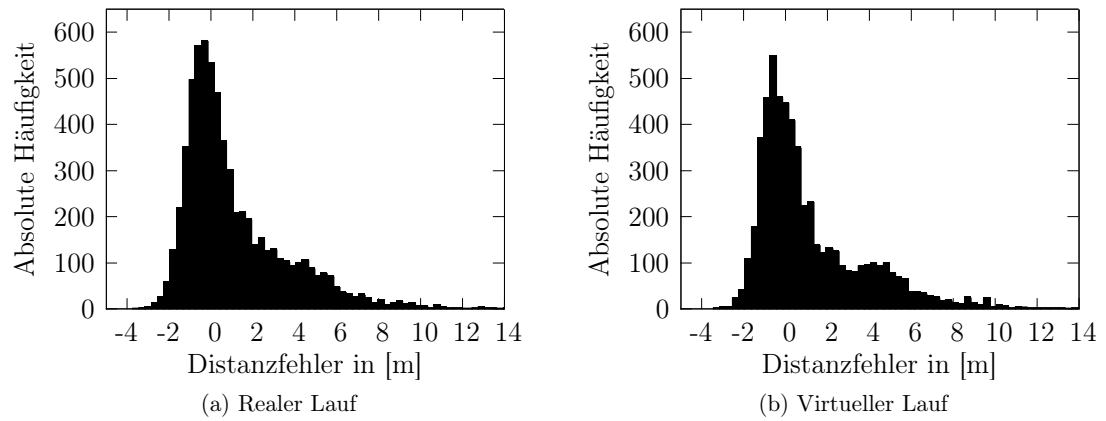


Abbildung 3.10: Fehlerverteilungen des realen und virtuellen Laufs.

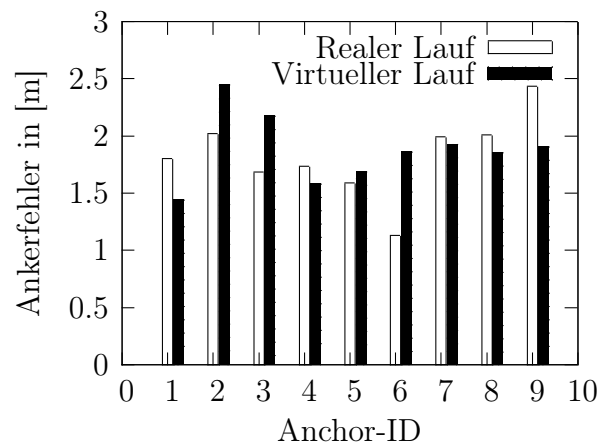


Abbildung 3.11: Durchschnittlicher Fehler pro Anker.

Tabelle 3.2: Resultate verschiedener Lokalisierungsalgorithmen

Alg.	Positionsfehler							
	Machbarkeitsstudie				Reale Büroumgebung			
	Realer Lauf		Virtueller Lauf		Realer Lauf		Virtueller Lauf	
	Ø	Max	Ø	Max	Ø	Max	Ø	Max
LLS	3,78 m	23,08 m	3,90 m	84,75 m	3,65 m	101,10 m	3,95 m	75,08 m
NLLS	2,41 m	10,78 m	2,43 m	9,99 m	1,98 m	11,87 m	2,34 m	8,70 m
Min-Max	0,89 m	3,71 m	0,90 m	3,72 m	1,52 m	8,54 m	1,75 m	7,11 m

### Ergebnisse der Lokalisierung

Neben der Evaluation der Distanzwerte werden im Folgenden auch die Resultate der Lokalisierung auf den so extrahierten Daten mit drei verschiedenen Algorithmen betrachtet. Es werden NLLS und LLS [109, 110] und Min-Max [136] genutzt. Tabelle 3.2 zeigt die betrachteten Parameter. Der durchschnittliche Positionsfehler des virtuellen gleicht dem des realen Laufs bei allen Algorithmen sehr. Der relative Unterschied zwischen realem und virtuellem Lauf pro Algorithmus lautet wie folgt:

- 0,83% für NLLS
- 3,17% für LLS
- 1,12% für Min-Max

Dies ist tatsächlich besser als zunächst erwartet. Wie bereits zuvor kurz erwähnt, hat der durchschnittliche Fehler pro Anker nur einen vernachlässigbaren Einfluss auf die Genauigkeit der Lokalisierungsalgorithmen. Der einzige Ausreißer ist im maximalen Fehler des LLS-Algorithmus zu finden. Dies kann durch eine nachteilige Konstellation im realen Lauf erklärt werden, wie Zekavat und Buehrer beschreiben [1].

#### 3.4.2 Reale Büroumgebung

Das vorige Kapitel zeigt, dass das virtuelle Testbed in einfachen Umgebungen valide Ergebnisse liefert. Zur Veranschaulichung und Evaluation von größeren Experimenten werden nun auch die Daten des Gebäudeszenarios betrachtet und analog zur Machbarkeitsstudie evaluiert. Der reale Pfad ist in Abb. 3.12 dargestellt. Dieser Pfad wurde gewählt, da die Datenbank zum Zeitpunkt der Messung hier genügend Daten enthielt, genauer 1.617.275 Datenpunkte im Flur.



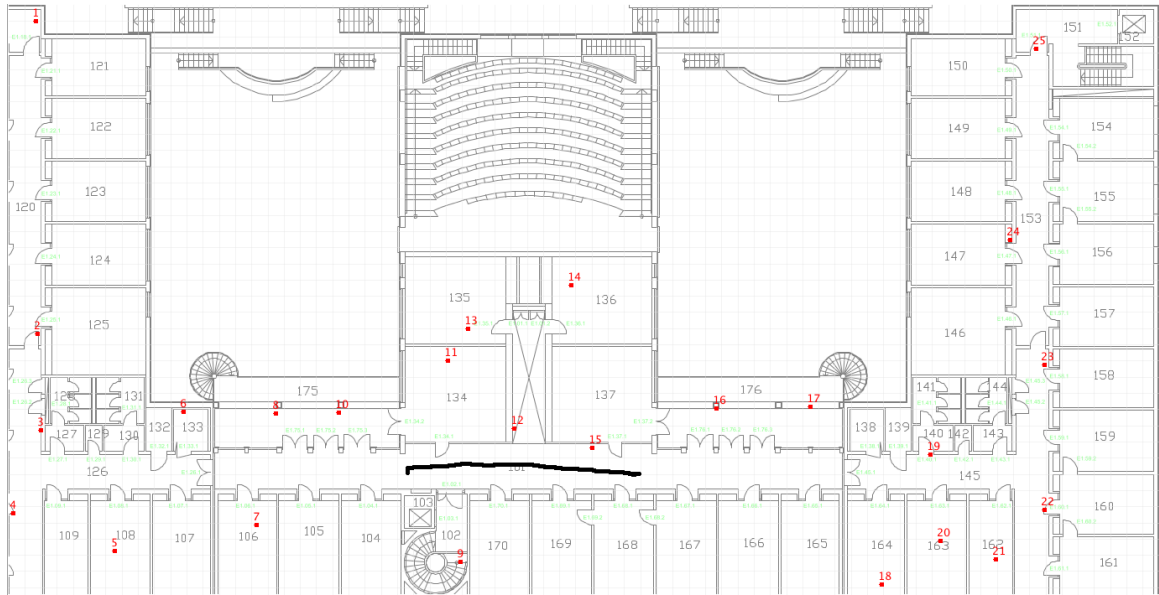


Abbildung 3.12: Reale Büroumgebung mit realem Pfad.

Tabelle 3.3: Parameter des realen und virtuellen Laufs

	Realer Pfad	Virtueller Pfad
Erfolgreiche Messungen	4.977	3.981
Durchschnittliche erreichte Anker	8,69	8,01
Durchschnittlicher Distanzfehler	1,85 m	1,94 m

### Ergebnisse der Distanzmessungen

Auch in der realen Büroumgebung liegt der virtuelle sehr nah am realen Pfad. Tabelle 3.3 vergleicht die Parameter beider Läufe.

In der Machbarkeitsstudie sind die Distanzfehler in beiden Läufen ähnlich. Die zugrundeliegende Annahme wird auch hier mit einem KS-Test unter den selben Bedingungen überprüft. Auch hier bestätigt sich die  $H_0$ -Hypothese mit einem p-Wert von 0,799. Diese Ergebnisse zeigen, dass die virtuellen Läufe auch unter NLOS-Bedingungen zur Simulation von realen Läufen genutzt werden können. Der relative Unterschied der durchschnittlichen Distanzfehler liegt bei 4,86%, was zwar leicht über dem Wert in der Machbarkeitsstudie liegt, aber in Anbetracht des komplexeren Szenarios ein gutes Ergebnis widerspiegelt.

Abbildung 3.13 zeigt den Fehler pro Anker. Hier zeigt sich zwischen realem und virtuellem Lauf eine bessere Übereinstimmung als in der Machbarkeitsstudie (vgl.

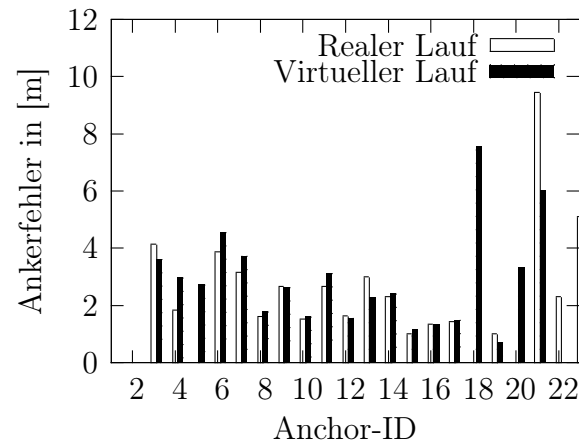


Abbildung 3.13: Fehler pro Anker in der realen Büroumgebung.

Abb. 3.10b). Ein Unterschied ist für vom realen Pfad relativ weit entfernte Anker erkennbar. Bei diese finden sich nur wenige Distanzmessungen auf dem Pfad. Es lässt sich vermuten, dass die signifikant größere, realistischere und von Menschen genutzte Umgebung hinter dem besseren Verhalten der realen Büroumgebung steckt. Die Machbarkeitsstudie wurde in einem leeren Raum durchgeführt. Die Umgebungsbedingungen dort waren größtenteils statisch, wobei Änderungen jedoch große Teile des Areals beeinflussen, sodass durchaus größere Unterschiede zwischen Aufzeichnung der Datenbasis und realem Lauf auftreten konnten. Dies hätte folglich Einfluss auf die Evaluation des virtuellen Laufs in dieser Studie. Die reale Umgebung hingegen ist mit unvorhersehbaren Änderungen in der Umgebung sehr viel dynamischer. Dies könnte zu höherem Rauschen über das gesamte Experiment hinweg führen und folglich bessere Ergebnisse bei der Evaluation des virtuellen Laufs nach sich ziehen.

### Ergebnisse der Lokalisierung

Auch hier werden die Ergebnisse der Lokalisierung unter den selben Bedingungen wie in der Machbarkeitsstudie betrachtet. Die Ergebnisse sind in Tab. 3.2 dargestellt. Der relative Unterschied zwischen realem und virtuellem Lauf pro Algorithmus lautet wie folgt:

- 18,18% für NLLS
- 8,21% für LLS
- 15,13% für Min-Max

Die Unterschiede sind gegenüber der Machbarkeitsstudie signifikant höher. Interessanterweise ist die relative Lokalisierungsgenauigkeit schlechter als der durchschnittliche Distanzfehler. Der Grund hierfür könnte in höheren Fehlern bei einzelnen Ankern liegen. Eine andere Erklärung könnte sein, dass die Algorithmen sensibel gegenüber der räumlichen Verteilung der Messungen mit großem Fehler sind. Dies wird auch durch Hillebrandt et al. gezeigt [137].

### 3.5 Zusammenfassung

Das virtuelle Testbed bietet die Möglichkeit, wiederholbare Experimente auf realen Daten durchzuführen. Die Ergebnisse dieser aufgezeichneten Experimente können der wissenschaftlichen Gemeinde helfen, das Verhalten von Lokalisierungstechniken und -algorithmen sowie den Einfluss von unvermeidbaren Messfehlern besser zu verstehen.

Das virtuelle Testbed kann u.a. genutzt werden für:

- Evaluation von Lokalisierungshardware und -techniken
- Leistungsanalyse für Lokisierungsalgorithmen in Gebäuden
- Experimente zur Platzierung von Ankern
- Strategieentwicklung für bestimmte Anwendungsszenarien

Das vorgestellte virtuelle Testbed zeigt die Machbarkeit derartiger Experimente. Eine Erweiterung auf andere Lokalisierungstechniken und -algorithmen ist leicht zu bewerkstelligen. Die hier vorgestellten Ergebnisse sind eine Momentaufnahme aus einem büroähnlichen Gebäude. Soll eine allgegenwärtige und alltagstaugliche Lokalisierung evaluiert werden, müssen weitere Innenräume unterschiedlicher Art untersucht werden. Dazu zählen zum Beispiel Wohnhäuser, öffentliche Gebäude wie bspw. Flughäfen oder Krankenhäuser, Kaufhäuser, Supermärkte, U-Bahnstationen, Industrieanlagen, Schiffsinnenräume und auch Bergwerksstollen, um einige zu nennen.

Eine weitere Möglichkeit des virtuellen Testbeds wäre die künstliche Erhöhung der Anzahl verwendeter Anker. Da bei Datenaufzeichnung lediglich Distanzen zu Ankern gemessen werden und noch keine Lokalisierung stattfindet, kann eine Aufzeichnung am selben Ort mit einem neuen Ankersetup wiederholt werden. Ist eine ausreichende Dichte der Datenpunkte gegeben, können so Distanzmessungen aus mehreren Aufzeichnungen an einem Ort zusammengefasst und einem Lokisierungsalgorithmus

übergeben werden. Hierbei müsste noch überprüft werden, ob eine Abhängigkeit zwischen Zeitpunkt und gemessener Distanzen besteht. Ist dies nicht der Fall, kann die Anzahl der Anker im SUT derart künstlich gesteigert werden.

Die vorgestellte Version des Testbeds erlaubt die Evaluation sog. Single-hop Systeme und Algorithmen. Hierbei messen mobile Teilnehmer Distanzen ausschließlich zu statischen Teilnehmern (Ankern). Bei Distanzmessungen zu anderen mobilen Teilnehmern, welche ggf. ihre Position durch bekannte Daten schätzen können, besteht ein dynamischeres Verhalten des Lokalisierungsprozesses. Das Testbed könnte zukünftig auch für derartige Verfahren genutzt werden. Hierzu ist keinerlei Anpassung notwendig, da das verwendete RS autark arbeitet und das Datenbankdesign diesen Fall bereits abdeckt.

---

## KAPITEL 4

---

# Räumliche Fehlerverteilung bei distanzbasierter Lokalisierung

Im Folgenden wird mit zuvor eingeführten Mitteln eine räumliche Analyse für Distanzmess- und Lokalisierungsfehler anhand eines Beispielgebäudes vorgestellt. Dieses Kapitel basiert auf folgenden eigenen Veröffentlichungen:

- S. Adler, S. Schmitt und M. Kyas, “Path loss and multipath effects in a real world indoor localization scenario”, in *Positioning, Navigation and Communication (WPNC), 11th Workshop on*, Best Paper Award, März 2014. DOI: 10.1109/WPNC.2014.6843300
- S. Adler, S. Schmitt, Y. Yang, Y. Zhao und M. Kyas, “Experimental evaluation of indoor localization algorithms”, in *Indoor Positioning and Indoor Navigation (IPIN), International Conference on*, Okt. 2014, S. 291–299. DOI: 10.1109/IPIN.2014.7275495
- S. Adler, S. Schmitt und M. Kyas, “Experimental evaluation of the spatial error distribution of indoor localization algorithms”, in *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS)*, Nov. 2014, S. 44–53. DOI: 10.1109/UPINLBS.2014.7033709

## 4.1 Einleitung

Die Effekte des Pfadverlusts und der Mehrwege-Ausbreitung von Funksignalen und die Leistung subsequenter Lokalisierungsalgorithmen sind in Theorie und in der Laborpraxis gut verstanden. Jedoch ist es überaus wichtig, reale Experimente durchzuführen, da Simulationen für allgemeine Gebäude schwierig zu bewerkstelligen sind. Zu den Unbekannten zählt der genaue Aufbau des Gebäudes, die Zusammensetzung der Wände, Wissen über Objekte und deren Materialien, bspw. sich bewegende Personen, oder auch störende Funkwellen anderer Systeme bzw. Luftfeuchtigkeit, Temperatur usw.

Bahillo et al. [138] stellen eines der realistischsten Experimente vor. Die Autoren evaluieren mit den Daten den neu vorgestellten Robust Least-Squared Multilateration (RLSM) Lokalisierungsalgorithmus. Das Experiment nutzt IEEE 802.11b konforme Geräte, welche RSSI und die Round Trip Time (RTT) in einem Gebäudeflur mit acht Ankerknoten in NLOS-Positionen messen. Der mobile Knoten wurde ca. 68 m durch das Gebäude getragen. Leider gibt die Veröffentlichung keine Auskunft über die Art und Weise der Groundtruth-Bestimmung und die Anzahl der individuellen Positionsbestimmungen.

Neben dieser Quelle finden sich einige weitere Feldtests in der Literatur. Ein einfacherer Ansatz ist die Nutzung der empfangenen Signalstärke (RSSI), um die Distanz zwischen Sender und Empfänger zu schätzen. Während dieser zwar sehr simpel umzusetzen ist, ist er für die meisten Anwendungsfälle zu ungenau. Ein gängiges Szenario ist die Einrichtung eines rechteckigen Bereichs, in dem ein oder mehrere unbekannte Knoten verortet werden, mit Ankerknoten an den Rändern der Fläche. Dieser Ansatz wird bspw. von Liu et al. [139] und in ähnlicher Art und Weise auch von anderen Autoren beschrieben [140–143]. Häufig werden wichtige Parameter nicht beachtet oder die gewählten Experimenteigenschaften erlauben nur sehr bedingt oder gar keine Aussagen über realistischere Szenarien. Evaluationen in sog. *Free-Space*- oder LOS-Umgebungen, unbekannte Knoten ausschließlich innerhalb der konvexen Hülle aller Anker oder die Installation von Ankern in einem Raster über die Spielfläche hinweg bilden Bedingungen nach, die in realen Umgebungen nur selten vorhanden sind.

Eine andere Klasse von Feldtests benutzt ToA-basierte Methoden. Da bei Funkdistanzmessungen extrem kurze Zeitabstände gemessen werden müssen, kann hier auf Ultraschall zurückgegriffen werden. Ein großer Vorteil ist die günstigere Sensorik und

breite Verfügbarkeit für solch einen Ansatz. Derartige Experimente werden bspw. von Wendeberg et al. [144] und auch von Priyantha et al. [145] und Moore et al. [146] gezeigt.

Amiot et al. beschreiben einen Ansatz für eine ganzheitliche Simulation für die Lokalisierung in Gebäuden [147]. Ein Framework erlaubt auch grundlegende Parameter über das Messsystem und die Umgebung, bis hin zu Oberflächen- und Materialeigenschaften, welche die Ausbreitung von Funkwellen beeinflussen, festzulegen. Hillebrandt et al. zeigen und motivieren in ihrer Dissertation die räumliche Analyse von Lokalisierungsalgorithmen [4]. Die Autoren zeigen anhand von Simulationen, dass die geometrischen Eigenschaften der räumlichen Fehlerausbreitung zwangsläufig zu einer wesentlichen Metrik zur Bewertung eines Algorithmus und damit auch eines Systems führen [137]. Der Durchschnitt des Lokalisierungsfehlers sagt hingegen nichts über diese Eigenschaften aus. Die Varianz des Fehlers kann auf eine ungleichmäßige Fehlerausbreitung hinweisen, reicht für eine Bewertung in konkreten Anwendungsfällen allerdings nicht aus. Die Autoren verbessern mit einer solchen Analyse den Min-Max Lokalisierungsalgorithmus.

Das RS und VirTIL bieten hier eine Analysemöglichkeit für ein dichtes Raster aus Evaluationspunkten von real gesammelten Distanzdaten. Dieses Vorgehen eliminiert die Notwendigkeit einer guten Auswahl an Evaluationspunkten. Weiterhin zeigen Untersuchungen, dass Änderungen der Umgebung, wie verschobene Einrichtungsgegenstände, eher vernachlässigt werden können [63]. Wichtiger ist demnach eine gute Platzierung der Anker. Diese Arbeit verwendet daher im Folgenden die räumliche Fehleranalyse auf den bisher gesammelten Daten aus VirTIL und konzentriert sich auf die Ankerplatzierung.

## 4.2 Pfadverlust und Mehrwege-Effekte

Im Folgenden werden Pfadverlust- und Mehrwege-Effekte bei Signalausbreitungen in LOS- und NLOS-Umgebungen unter realistischen Bedingungen betrachtet.

### 4.2.1 Motivation

Der Lokalisierungsprozess kann in die Informationssammlung und eine subsequente Ortsbestimmung durch Auswertung ebendieser unterteilt werden. Der Informations-

sammlung unterliegen dabei physikalische Effekte als Teil der Signalausbreitung. Die wesentlichen Effekte lauten:

**Diffraction (Beugung), Reflexion** Die Ablenkung eines Signals an einem Objekt. Die Richtung und Intensität ändern sich. Das Signal kann abgelenkt (Diffraction) und damit auch zurückgeworfen werden (Reflexion). Vor allem Metalle verursachen derartige Effekte.

**Refraktion (Brechung)** Das Signal wird bei Eintritt in ein anderes Medium durch eine sich ändernde Signalgeschwindigkeit, entsprechend dem Brechungsindex des Materials, gebrochen (abgelenkt). Der Eintritt in Glas oder Wasser ist hier in Gebäuden relevant.

**Streuung** Das Signal wird ähnlich der Refraktion in einem sich ändernden Medium in verschiedene Richtungen abgelenkt, ist also unvorhersehbarer als Refraktion. Unter anderem Rauch, Staub, Luftfeuchtigkeit oder unebene Oberflächen können eine Streuung verursachen.

**Absorption** Das Signal wird bei Transmission durch ein Medium entsprechend der Absorptionsrate gedämpft bzw. abgeschwächt (Signaldämpfung). Beispielsweise können tragende Wände eine hohe Absorption nach sich ziehen. Die Freiraumdämpfung limitiert das Signal im freien Raum (bspw. in sehr langen Korridoren).

**Interferenz** Das Signal wird durch Überlagerung verändert, wobei die übermittelten Informationen verloren gehen können. Reflexionen oder andere Signalquellen können Interferenzen verursachen.

Diese Effekte können zu geringeren Daten-, hohen Verlustraten, Mehrwege-Effekten oder, allgemein ausgedrückt, zu gegenüber der Realität verzerrten Messungen führen. Die International Telecommunications Union (ITU) gibt entsprechende Werte für verschiedene Gebäudematerialien und -strukturen an [148]. Der daraus resultierende Signal- oder auch Pfadverlust zwischen Sender und Empfänger kann in Bezug auf beliebige Standorte von Sender und Empfänger in Gebäuden grob modelliert werden. Die ITU definiert u.a. ein weitverbreitetes Modell für Innenräume (*ITU indoor propagation model*) wie folgt [149]:

$$L_{\text{total}} = L(d_0) + N \log_{10} \frac{d}{d_0} + L_f(n), \quad (4.1)$$

mit den Parametern:



$L_{\text{total}}$  Pfadverlust zwischen Sender und Empfänger in [dB]

$L(d_0)$  Pfadverlust bei Distanz  $d_0$  in [dB]

$N$  Koeffizient für den Signalverlust

$d$  Distanz zwischen Sender und Empfänger in [m]

$d_0$  Referenzdistanz zwischen Sender und Empfänger in [m]

$L_f(n)$  Etagenverlustfaktor in [dB]

$n$  Anzahl der Etagen zwischen Sender und Empfänger,  $n \geq 0$

Für eine Referenzdistanz  $d_0 = 1$  m und eine angenommene Freiraumausbreitung (engl. *free-space propagation*) gilt

$$L(d_0) = 20 \log_{10} f - 28 \quad (4.2)$$

mit der Frequenz  $f$  des Signals in [MHz]. Die Parameter  $N$  und  $L_f(n)$  werden empirisch für einen Gebäudetyp ermittelt, wobei die ITU bereits einige Werte vorschlägt. Diese Überlegungen helfen, die physikalischen Effekte zu verstehen. Allerdings sind diese in allgemeinen Umgebungen meist zu komplex, um eine verlässliche Beziehung zur intendierten Lokalisierung durch Funkwellen herzustellen. Daher wird die Signalausbreitung im Folgenden empirisch untersucht.

He et al. präsentieren ein Testbed zur Analyse von NLOS in ToA-basierten Systemen [150]. Die Autoren verwenden den gleichen Funkchip zur Distanzbestimmung wie in dieser Arbeit (nanoLOC, Nanopan 5375, [135]). Es wird ein kombinierter Ansatz aus Simulation und Feldtests beschrieben. He et al. entwickeln ein in realen und sehr realistischen Experimenten evaluiertes, komplexes Kanalmodell, um die Eigenschaften des Transceivers zu beschreiben. Ebenso wird eine gute Kategorisierung für verschiedene Mehrwege-Effekte gegeben. Die Autoren unterteilen den in der Literatur typisch anzutreffenden NLOS-Effekt in weitere Fälle wie folgt:

**Free-Space** Es existiert eine Sichtverbindung zwischen Sender und Empfänger. Es gibt keine die Verbindung beeinflussende Objekte in der Umgebung. Der Empfänger empfängt nur das direkte, und keine reflektierten Signale des Senders.

**LOS** (Line of Sight) Es existiert eine Sichtverbindung zwischen Sender und Empfänger, wobei das Signal auf dem direkten Weg gleichzeitig das stärkste empfangene Signal darstellt. Mehrwege-Signale können ebenso empfangen werden.

**NLOS-DDP** (Non Line of Sight - detectable direct path) Es existiert keine Sichtverbindung zwischen Sender und Empfänger, jedoch wird das direkte und nicht reflektierte Signal immer noch detektiert.

**NLOS-UDP** (Non Line of Sight - undetectable direct path) Es existiert keine Sichtverbindung zwischen Sender und Empfänger. Der Anteil des Signals, welcher den direkten Weg zurücklegen könnte, wird am Empfänger nicht detektiert. Der Empfänger detektiert lediglich Mehrwege-Signale.

Im Folgenden wird die Signalausbreitung des in Kapitel 3.3.4 vorgestellten Datensatzes untersucht (Gebäudeflur des FU-Gebäudes, 2,4 GHz Hardware). Unter Verwendung der von He et al. vorgestellten Klassifizierung wird die räumliche Verteilung des Distanzfehlers einzelner Anker analysiert und die Platzierung ebendieser diskutiert. Die Arbeit zeigt, dass der Vorschlag von Hillebrandt et al. (räumliche Fehlerbetrachtung als Bewertungsmetrik [4, 137]) auch für diese Verteilung gilt. Nach Begutachtung des Datensatzes werden zwölf der verwendeten 25 Anker mit der signifikantesten Abweichung vom Erwarteten ausgewählt. Alle Ergebnisse sind in farbkodierten Karten dargestellt. Eine Kachel deckt dabei eine Fläche von  $0,25\text{ m} \times 0,25\text{ m}$  ab. Sofern mehrere Werte in einer Kachel verfügbar sind, wird der durchschnittliche Distanzfehler farbkodiert dargestellt.

#### 4.2.2 Weitgehend NLOS-freie Ankerpositionen

Abbildung 4.1 zeigt eine Teilmenge der Ankerplatzierungen mit weitgehend geringem NLOS-Anteil im Experiment. Die räumlich betrachtete Fehlerverteilung ist weitgehend homogen. In manchen Fällen finden sich jedoch zunächst intuitiv als abnormal empfundene Effekte. Alle Anker wurden im Flur bzw. auf den Balkonen im Innenhof platziert, womit sie gute LOS-Bedingungen zu einem Großteil ihrer Umgebung haben sollten. Es gibt nur wenige erfolgreiche Messungen in den durch Wände abgeschotteten NLOS-Bereichen.

Anker 1 in Abb. 4.1a zeigt einen niedrigen und weitgehend homogen verteilten Distanzfehler im gesamten Flur, den er ausleuchtet. Bei genauerem Betrachten zeigt sich vor dem Raum 111, im linken unteren Bereich, ein höherer Fehler ( $>2,0\text{ m}$ ) als in der unmittelbaren Umgebung ( $<0,5\text{ m}$ ). Erwähnenswert ist hier, dass unmittelbar neben diesem Gebiet ein anderer Anker platziert war. Diese unvorteilhaften Bereiche

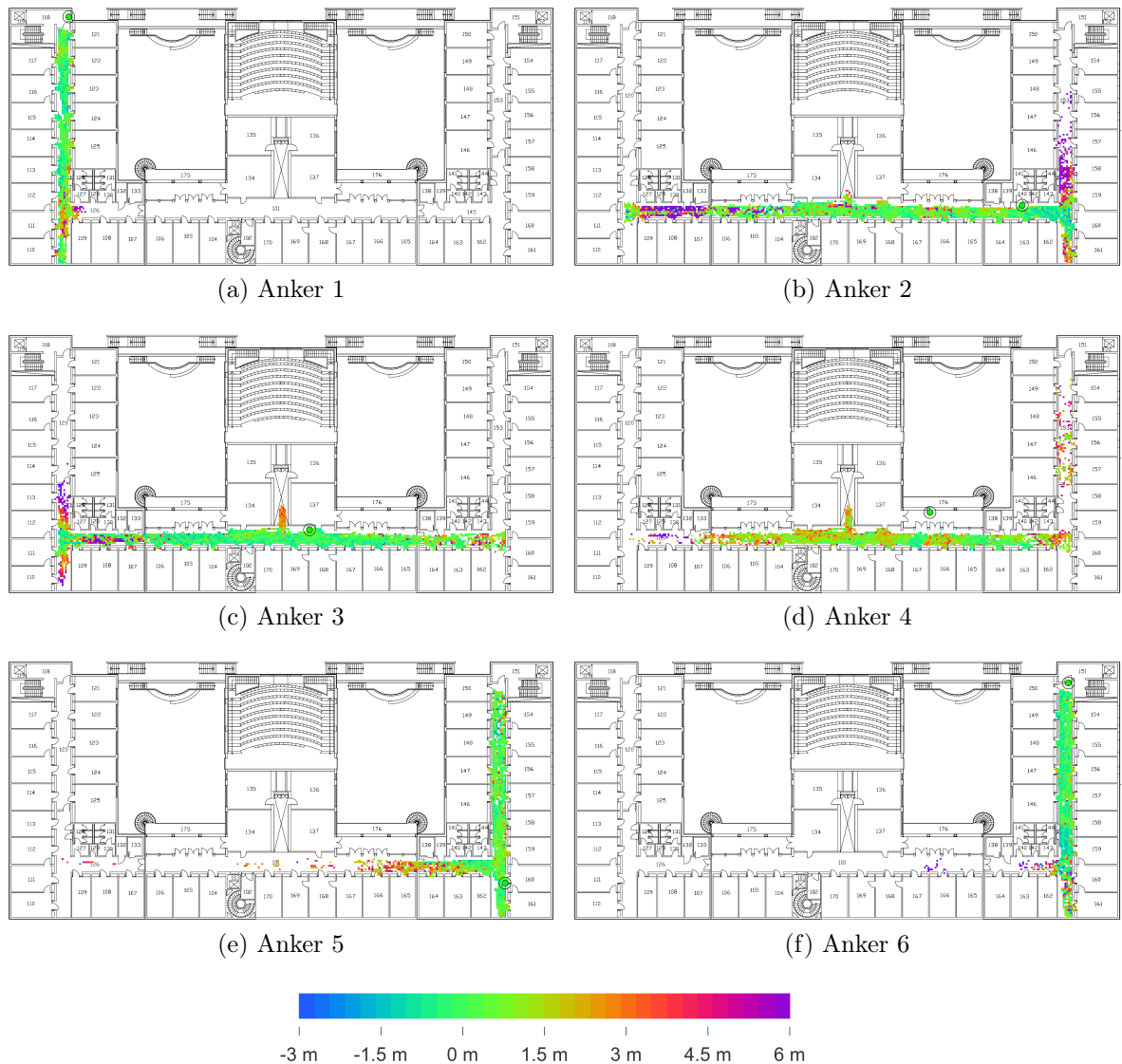


Abbildung 4.1: Räumliche Fehlerverteilung verschiedener Ankerplatzierungen mit wenig NLOS-Anteil.

finden sich auch bei anderen Ankern.

Anker 2 in Abb. 4.1b zeigt ein ähnliches Szenario. Im rechten vertikalen Flur vergrößert sich der Distanzfehler wie erwartet nach wenigen Dezimetern sehr schnell, bzw. kommen erfolgreiche Messungen aufgrund von NLOS-Bedingungen vermehrt nicht mehr zustande. Im linken Bereich kommt es zu ähnlichen Effekten wie bei Anker 1: Kurz vor Ende des Flurs erhöhen sich die Distanzfehler am Rande des Flurs rapide, und ergeben ähnlich unvorteilhafte Bereiche wie zuvor. Bemerkenswert ist der niedrige Distanzfehler hinter diesem Bereich. Eine mögliche Erklärung könnte hier die

(geöffnete) Feuerschutztür sein. Die metallische Tür bietet beidseitig vergleichsweise große Flächen im Flur zwischen Raum 106 und 107, welche die Ablenkung und, daraus folgend, Überlagerung der Funkwellen begünstigen können. Dadurch könnte hier das reflektierte Funksignal mit längerer Wegstrecke zur Distanzmessung genutzt worden sein.

Anker 3 in Abb. 4.1c befindet sich im mittleren Teil des Gebäudes. Vor den Räumen 107 und 108 leiden die Distanzmessungen in einem deutlich abgegrenzten Bereich in der Mitte des Flurs unter einem sehr hohem Distanzfehler ( $>6$  m). Dieser unvorteilhafte Bereich ist nach obiger Definition klar im LOS-Verhältnis zum Anker 3. Daher werden solche Effekte nicht erwartet. Auch hier ist der niedrige Distanzfehler hinter diesem Bereich am Ende des Flurs bemerkenswert.

Die Definition nach He et al. beschreibt die LOS-Bedingung derart, dass kein Hindernis den direkten Pfad des Signals zwischen Sender und Empfänger verdeckt, wobei dieses direkte Signal gleichzeitig das stärkste empfangbare Signal ist [150]. Mehrwege-Anteile des Signals können am Empfänger detektiert werden und auch den Spitzenwert des ersten Signals verschieben. Nach den Autoren kann diese Verschiebung allerdings nur moderate Distanzfehler zur Folge haben. Durch die beobachteten Distanzfehler muss folglich obige Definition von erwartbaren Fehlern erweitert werden. Die entsprechenden Effekte fanden unter LOS-Bedingungen statt. Dennoch werden räumlich gut separierbare Bereiche mit signifikant höherem Distanzfehler als in ihrer unmittelbaren Umgebung beobachtet. Diese Arbeit wählt hierfür die Bezeichnung *LOS-Multipath (LOS-MP)*. Diese Bedingung beschreibt eine LOS-Situation, in der nicht das direkte Signal zwischen Sender und Empfänger, sondern ein Mehrwege-Signal erkannt wird.

Bisher gibt es keine zufriedenstellende Erklärung für den LOS-MP-Effekt, da gängige Modelle einen solchen Fehler nicht vorhersagen. Aufgrund des experimentellen Aufbaus ist es auszuschließen, dass ein anderer Anker zu diesem Effekt beiträgt. Durch das Round-Robin-Verfahren sendet zu einem Zeitpunkt immer nur ein Gerät. Abbildung 4.2 zeigt die räumliche Verteilung von erfolgreichen Messungen. Im Umfeld des Bereichs sind lediglich ca. 60 % der Distanzmessungen erfolgreich, wohingegen es unmittelbar davor noch annähernd 100 % waren. Hinter dem fraglichen Bereich finden sich bei optimalen Distanzmesswerten ebenso nur rund 60 % erfolgreiche Messungen. Damit erklärt diese Korrelation nicht den erhöhten Distanzfehler. Ein weiterer Aspekt ist die bei Datenaufzeichnung relativ chaotische Fahrweise des Roboters. Da dieser die

Grenze zwischen dem fraglichen Bereich und der unmittelbaren Umgebung mehrmals überschritt und damit den Bereich nicht am Stück abdeckte, kann ein temporärer Ausfall der Hardware ausgeschlossen werden.

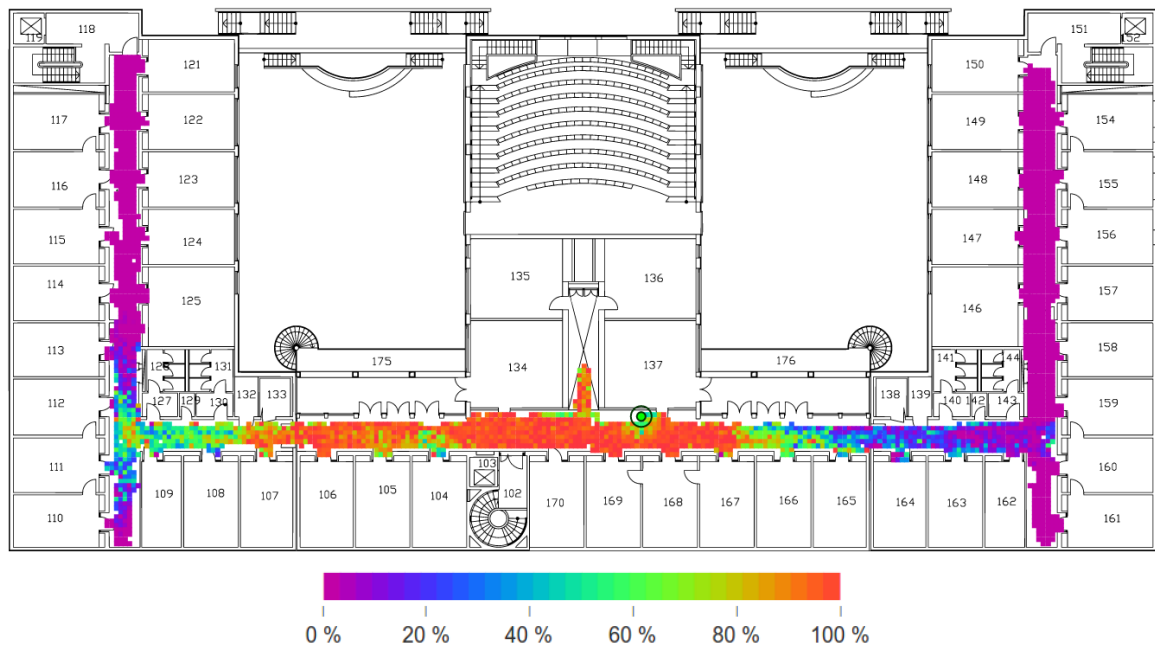


Abbildung 4.2: Erfolgreiche Distanzmessungen zu Anker 3.

Abbildung 4.1d zeigt bei Anker 4 den Einfluss von Pfeilern zwischen Anker und mobilem Knoten. Zwar werden rechts der Feuerschutzwand noch sehr gute Distanzmessungen erzielt, jedoch ergeben sich noch weiter rechts unter deutlicheren NLOS-Bedingungen wie erwartet höhere Distanzfehler. Allerdings gibt es, trotz einigen Wänden auf dem direkten Pfad, auch Bereiche im rechten unteren Ausschnitt hinter der Feuerschutzwand mit unerwartet signifikant niedrigen Distanzfehlern. Dies könnte ein NLOS-DDP-Phänomen sein. Auch weiter oben im Flur existieren Bereiche mit inhomogenen Distanzfehlern, welche allerdings auf Türen und Fenster auf dem direkten Weg zwischen Anker und mobilem Knoten zurückzuführen sind. Es handelt sich hierbei also um LOS- bzw. NLOS-DDP-Messwerte.

Abbildung 4.1e zeigt bei Anker 5 auch eine weitgehend homogene Ausleuchtung mit niedrigem Distanzfehler im rechten Flur. Durch die Positionierung an der rechten Wand im Flur ist ein niedriger Fehler auf den ersten Metern im horizontalen Flur möglich. Wie erwartet verschlechtert sich der Distanzfehler zunehmend. Die Rate erfolgreicher Messungen fällt ebenso vor Raum 167 auf Null. Allerdings sind immer

noch vereinzelte Messungen links bis ans Ende des Flurs mit hohem Distanzfehler möglich.

Abbildung 4.1f zeigt bei Anker 6 ähnliche Effekte wie schon bei Anker 1 und 3. Hier erhöht sich der Distanzfehler ebenso in bestimmten Bereichen unter LOS-Bedingungen, und kehrt dahinter zu Normalwerten zurück. Zwischen den Räumen 158 und 159 findet sich ein Bereich, in dem die Distanzfehler unter  $-1$  m fallen. Diese Beobachtung lässt sich nicht über physikalische Effekte der Signalausbreitung erklären. Stattdessen handelt es sich hierbei um temporär zu langsam laufende Uhren oder eine abweichende interne Signalverarbeitung aus unbekanntem Gründen im laufzeitbasierten, proprietären Messsystem. Eine statische Kalibrierung stellte sich daher als nicht praktikabel heraus. Im rechten unteren Bereich der Abbildung hinter dem fraglichen Bereich sind nur ca. 15 % der Distanzbestimmungen erfolgreich, was die inhomogenen Distanzfehler erklären könnte.

### 4.2.3 Ankerpositionen mit deutlichen NLOS-Effekten

Um den Effekt von NLOS-Bedingungen in dieser Umgebung zu untersuchen, werden einige Anker bewusst in (angenommen) unvorteilhaften Positionen installiert, welche zu hohen Distanzfehlern führen sollten. Abbildung 4.3 zeigt nun eine Teilmenge der so platzierten Anker.

Abbildung 4.3a und Abb. 4.3b zeigen bei Anker 7 bzw. Anker 8 eine gute Ausleuchtung unter LOS-Bedingungen mit kaum Distanzfehler im linken Flur. Beide haben jedoch einen unerwartet hohen Distanzfehler ( $>6$  m) im horizontalen Flur unter NLOS-Bedingung. Das Ausmaß der Distanzfehler ist bei Anker 8 besonders hoch, kann aber durch die Reflektion an der Wand zwischen Raum 111 und 112 erklärt werden. Die Feuerschutztüren im horizontalen Flur begünstigen die Signalausbreitung durch Reflektion vermutlich zusätzlich. Anker 8 produziert auch einige gute Distanzmessungen vor den Räumen 108 und 109 unter NLOS-DDP-Bedingung.

Abbildung 4.3c zeigt Anker 9 in einer Toilette mit Fliesen an allen vier Betonwänden des Raumes. Die Wand zur rechten Seite ist die Feuerschutzwand. Es lässt sich vermuten, dass diese Bedingungen zu äußerst wenigen erfolgreichen Distanzmessungen führen. Vor den Räumen 105 bis 170 waren nur zwischen 0 % und 30 % aller Messungen erfolgreich. Jedoch ist der Distanzfehler bei diesen Messungen unter NLOS-UDP-

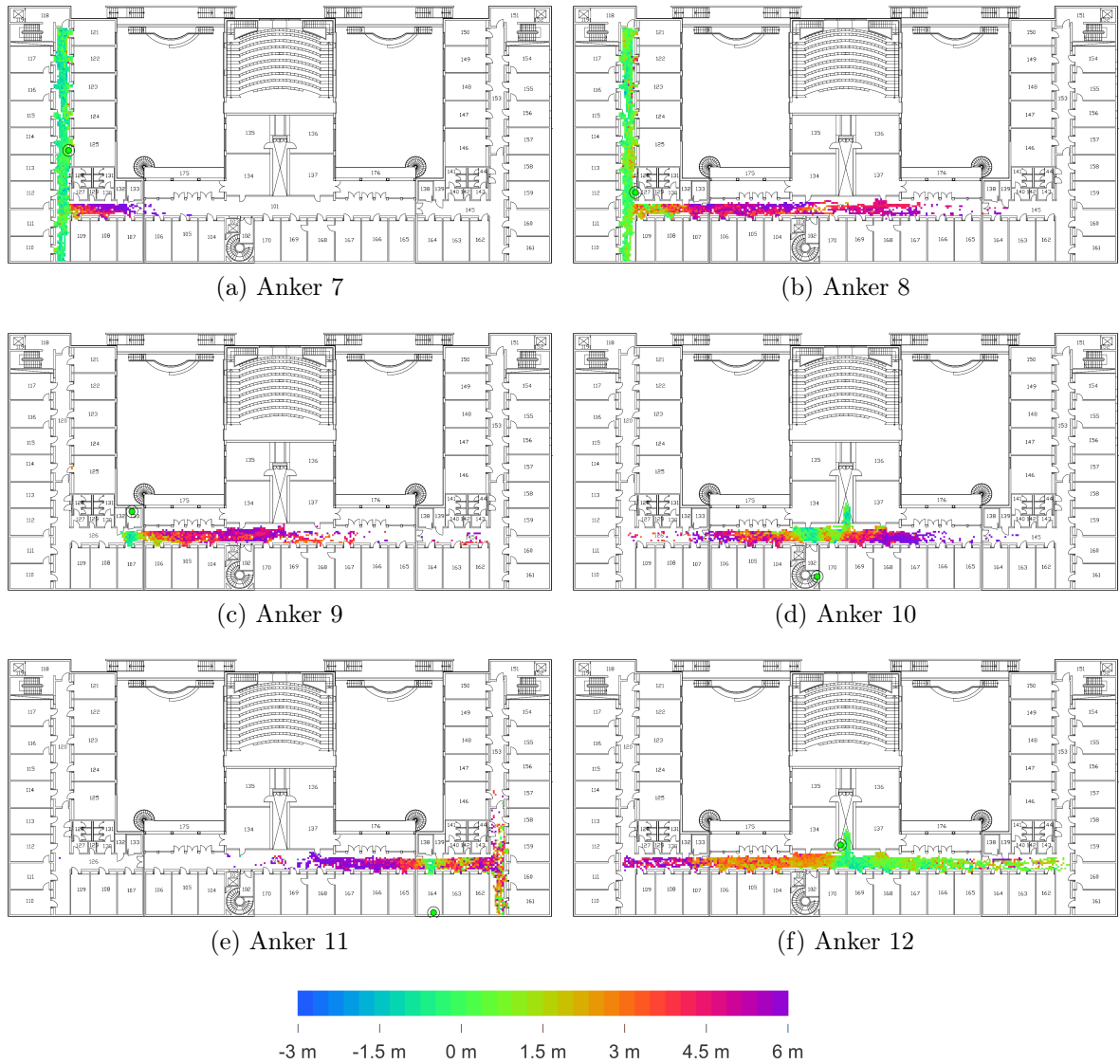


Abbildung 4.3: Räumliche Fehlerverteilung verschiedener Ankerplatzierungen mit hohem oder unerwartetem NLOS-Anteil.

Bedingungen meist größer als 4,5 m. Es ist erwähnenswert, dass links des Raumes 107 keine Messungen möglich waren.

Abbildung 4.3d zeigt Anker 10 im Treppenhaus neben einem Aufzugsschacht. Die horizontale Betonwand zwischen den Büroräumen und dem Flur zeigt einen viel größeren Einfluss auf den Distanzfehler als die ebenfalls aus Beton bestehende Wand direkt neben dem Anker. Dies erlaubt einige gute Distanzmessungen vor dem Raum 170 unter NLOS-DDP-Bedingung.

Abbildung 4.3e zeigt bei Anker 11 den selben Effekt. Nun schattet die Feuerschutzwand alle direkten Signale vor und links dem Raum 165 (NLOS-UDP). Die Beton- und Stellwände auf der rechten Seite des Ankers beeinflussen die Messungen auf dem Flur zur rechten Seite hin. Dennoch sind ca. 20 % der Messungen erfolgreich.

Anker 9, 10 und 11 wurden alle in mit Holztüren verschlossenen Räumen installiert. Sie zeigen alle einen sehr niedrigen Distanzfehler auf Höhe der Tür zum entsprechenden Raum. Hierbei handelt es sich ebenfalls um eine NLOS-DDP-Bedingung.

Abbildung 4.3f zeigt mit Anker 12 eine andere unvorteilhafte Position eines Ankers, direkt hinter einer Ecke. Hier scheint die gegenüberliegende Wand (zum Raum 137) das Signal sehr gut zu reflektieren, sodass ein weitgehend homogener Distanzfehler in ungefährer Höhe der doppelten Flurbreite an dieser Stelle vor den Räumen 106 bis 170 unter NLOS-UDP-Bedingung auftritt. Wieder ist der Einfluss der Feuerschutztüren links des Raumes 106 erkennbar, welcher zu einem nochmals signifikant höheren Distanzfehler führt.

Bei den Ankern 5 und 6 erkennt man einen homogen niedrigen Distanzfehler. Gegenteilig dazu besitzen die Anker 7 und 8 einen sehr großen Fehler im horizontalen Flur. Eine mögliche Erklärung der hohen Fehler ist die Platzierung der Anker in unmittelbarer Nähe zur rechten Wand, welche dem direkten Signal eine Reflektion an der linken Wand in den horizontalen Flur hinein ermöglicht. Es treten mehr erfolgreiche Messungen auf, wenn der Anker näher am horizontalen Flur platziert wird, wie es bei Anker 8 der Fall ist. Eine bessere Platzierung wäre nach dieser Einschätzung neben der gegenüberliegenden Wand möglich, wie es bei Anker 5 bereits gezeigt wurde. Dies begünstigt einige der möglichen Reflektionen nicht und führt daher zu niedrigeren Distanzfehlern im horizontalen Flur. Allerdings gibt es hier aufgrund weniger Reflektionsmöglichkeiten auch einen Einbruch in der Anzahl erfolgreicher Messungen.

#### 4.2.4 Zusammenfassung

In diesem Experiment wurden aufgrund der Limitierungen des Referenzsystems lediglich Daten im Flur des Gebäudes gesammelt. Daher werden auch nur Aussagen über ebendiesen getroffen. Es wurden Anker unter LOS- und NLOS-Bedingungen im Flur sowie in Büroräumen platziert. Bei Letzteren finden sich meist schlechte Distanzmessungen im Flur durch deutliche NLOS-UDP-Bedingungen.

Es wurde gezeigt, dass aktuelle Pfadverlustmodelle für Lokalisierung innerhalb von



Gebäuden nicht alle Effekte, die in dieser Umgebung vorgefunden wurden, erklären können. Während der Einfluss von NLOS-Bedingungen den Distanzfehler meist (wenngleich nicht immer) signifikant erhöht, existieren auch sehr hohe Distanzfehler unter LOS-Bedingungen, die zuvor nicht erwartet wurden. Es wird die Kategorie LOS-Multipath vorgeschlagen. Diese beschreibt die Situation, dass ein Teil des Signals für die Distanzbestimmung benutzt wird, welches, trotz keiner vorhandenen Hindernisse zwischen Sender und Empfänger, nicht den direkten Weg zurücklegte. Diese Arbeit motiviert den Ursprung dieser Effekte weiter zu untersuchen und vorhandene Modelle anzupassen.

Die Aussagen treffen für das verwendete SUT in diesem und ähnlichen Gebäuden zu. Beispielsweise ist die genaue Signalverarbeitung im verwendeten Transceiver nicht bekannt. Um die Aussagen als allgemeingültig anzusehen, sollten die Ergebnisse daher in anderen Gebäudearten und mit anderen Systemen reproduziert werden.

### 4.3 Betrachtung von Lokalisierungsalgorithmen

Im letzten Kapitel wurden die Distanzmessungen beleuchtet. Im Folgenden werden gängige Lokalisierungsalgorithmen vorgestellt und die Ortsbestimmung mithilfe dieser Messungen untersucht.

#### 4.3.1 Motivation

In RF-basierten Lokalisierungsszenarien bedarf es einer Algorithmik, die den Einfluss verschiedener LOS- und NLOS-Effekte bei einer Positionsschätzung mildern. Es wurden verschiedene, gegenüber solcher Effekte robuste, Lösungsverfahren vorgeschlagen. Die meisten dieser Algorithmen werden in Simulationen oder kleinen statischen Testbeds evaluiert. Oft fehlt ein rigoroser Vergleich zu bisherigen Algorithmen in realistischen Szenarien.

Nachdem die Distanz zu einer angemessenen Anzahl an Ankerknoten geschätzt wurde, findet eine Positionsschätzung aufgrund der Messdaten statt. Eine erste Herangehensweise ist die Anwendung von Laterationsalgorithmen wie NLLS oder LLS [109, 110]. Beide Algorithmen sind gut bekannt aus der Geodäsie, insbesondere bei GPS-basierten Problemen [151]. Die vergleichsweise hohen und vermehrt vorkommenden Distanzfehler in Gebäudeszenarien stellen jedoch eine große Herausforderung dar.

Diese können um Größenordnungen größer sein als Distanzwerte eines GNSS oder anderer geodätischer Methoden. Folglich wurden einige Lokalisierungsalgorithmen für Gebäude vorgeschlagen und oft ausschließlich durch Simulation evaluiert, bspw. Adapted Multi-Lateration (AML) [152], Iterative Clustering-based Localization Algorithm (ICLA) [153], Min-Max [111, 112] und Geo-n [48]. Diese werden im folgenden Kapitel kurz beschrieben. Viele Autoren berichten nicht über Experimente in realen Umgebungen. Oft ist die Simulation ein vermeintlich ausreichendes Mittel zur Evaluation. Gegenüber Feldtests in realen Umgebungen mit spezieller und teurer Hardware stellt dies die kostengünstigere und vor allem schneller zu implementierende Alternative dar. Ein geeignetes Referenzsystem ist außerdem nötig, insbesondere um die Ergebnisse numerisch miteinander vergleichen zu können [66, 96, 98].

Bei Simulationen sind unter Umständen ein fehlerhaftes Verhalten des Transceivers, physikalische Effekte in der Umgebung und Parameter der Funkausbreitung über angenommene Wahrscheinlichkeitsfunktionen modelliert, welche nicht immer der Realität entsprechen. Der Grund dafür ist die Verwendung durchschnittlicher Wahrscheinlichkeitsverteilungen für Distanzfehler, wie bspw. bei Hillebrandt, Will und Kyas [137] und im Simulator LS<sup>2</sup> [154]. Wie zuvor gezeigt, treten physikalische Effekte oft sehr lokal und gehäuft auf (verschiedene NLOS-Situationen). Die Bildung des Durchschnitts über alle Messwerte (aus LOS- und NLOS-Situationen) und die Annahme einer solchen Verteilung über die Grundgesamtheit hinweg kann einen Lokalisierungsalgorithmus schlechter erscheinen lassen, als er bei guter Ankerkonstellation unter Vermeidung solcher Effekte wäre. Daher ist die Kenntnis über meist unvermeidliche lokale Effekte essentiell für eine Bewertung und einen Vergleich von Algorithmen. Wird diese lokale Häufung physikalischer Effekte in Simulationen nicht berücksichtigt, werden Algorithmen unter Umständen falsch bewertet.

In den folgenden Kapiteln werden nun zunächst gängige distanzbasierte Lokalisierungsalgorithmen vorgestellt und Auswirkungen der aufgezeigten physikalischen Effekte bei Distanzmessungen unter diesen Bedingungen analysiert. Es wird dazu der im Kapitel 3.3.4 vorgestellte Datensatz (Gebäudeflur des FU-Gebäudes, 2,4 GHz Hardware) verwendet.

### 4.3.2 Ausgewählte Lokalisierungsalgorithmen

Dieses Kapitel (4.3.2) wurde von M. Kyas [155] und kurz darauf in abgewandelter Form von S. Adler, dem Autor dieser Dissertation, M. Kyas und anderen nochmals veröffentlicht [69, 71]. Demzufolge ist der Autor dieser Dissertation nicht der ursprüngliche Autor des Textes und macht sich weder Form noch Inhalt zu eigen. Zum besseren Verständnis der nachfolgenden Kapitel wurde der Rest des Kapitels aus diesen Veröffentlichungen frei übersetzt.

In den letzten Jahren wurden einige distanzbasierte Lokalisierungsalgorithmen vorgeschlagen. Manche fokussieren eine geringe Komplexität, um für sparsame Sensornetzwerke geeignet zu sein, andere fokussieren Robustheit gegenüber Messfehlern in Gebäudeszenarien. Letztere wurden speziell auf eine Reduzierung der Effekte von Distanzmessfehlern durch eine unvorhersehbare Mehrwegeausbreitung, geringe Signal to Noise Ratio (SNR) oder Reflexionen zur Verbesserung der Positionsschätzung optimiert [109, 110, 142, 156, 157]. Gitter-basierte Ansätze unterteilen den Suchraum in Zellen und wenden Abstimmungsverfahren zur Auswahl der korrekten Zelle als geschätzte Position an [158, 159]. Verfeinerte geometrische Ansätze bestimmen auf Basis verrauschter Distanzmessungen eine relative anstatt einer absoluten Position [146, 157].

Es werden insgesamt sechs Algorithmen verglichen. Drei davon sind gut bekannte und sehr einfach gehaltene Algorithmen, welche von Autoren bei Vorstellung neuer Verfahren oft als Vergleich oder als initiale Schätzung für eine anschließende Verfeinerung der Position genutzt werden: Multilateration mit NLLS [110], LLS [109] und Min-Max [111, 112]. Die drei aktuelleren Algorithmen nutzen die Schnittpunkte der Distanzkreise um bekannte Ankerknoten zur Schätzung: AML [152], ICLA [153] und Geo-n [48]. Letztere wurden explizit für den Gebrauch in NLOS-behafteten Umgebungen entwickelt. Es handelt sich dabei um eine repräsentative Auswahl vorhandener distanzbasierter Lokalisierungsalgorithmen.

## Nonlinear Least Squares Multilateration (NLLS)

Die Lösung des Lokalisierungsproblems kann als Gleichungssystem verstanden werden. Eine Gleichung modelliert dabei den Distanzkreis einer Distanzmessung um einen Anker mit bekannten Koordinaten. Gegeben seien  $N$  Ankerknoten mit einer bekannten Position im Punkt  $b_i = (x_i, y_i)$  für  $i = 1, 2, \dots, N$  mit möglicherweise rauschenden Distanzmessungen  $r_i$  zwischen diesen Ankern und einem unbekanntem Knoten im Punkt  $u = (x, y)$ . Die durch Multilateration geschätzte Position wird  $\hat{u}$  genannt. Daraus wird ein Gleichungssystem wie folgt formuliert:

$$\begin{aligned} (x - x_1)^2 + (y - y_1)^2 &= r_1^2 \\ (x - x_2)^2 + (y - y_2)^2 &= r_2^2 \\ &\vdots \\ (x - x_N)^2 + (y - y_N)^2 &= r_N^2 \end{aligned} \tag{4.3}$$

Dieses Problem wird meist über die Minimierung der Summe der quadrierten Residuen zwischen den Distanzmessungen  $r_i$  und den geschätzten Distanzen  $\|u - b_i\|$  gelöst:

$$\hat{u} = \underset{u}{\operatorname{argmin}} \sum_{i=1}^N (\|u - b_i\| - r_i)^2 \tag{4.4}$$

Dieses Problem kann durch ein Newton-Verfahren gelöst werden [160]. Diese beginnen mit dem Raten einer Lösung, hier eine Positionsschätzung, und einer nachfolgenden Iteration mit dem Ziel, diese graduell zu verbessern bis ein Minimum der Zielfunktion, Formel 4.4, gefunden wurde. Es existiert eine nicht unerhebliche Wahrscheinlichkeit, ein lokales Minimum zu finden. Damit die Lösung in der Nähe des globalen Minimums gefunden wird, sollte der Algorithmus mit verschiedenen initialen Positionen gestartet werden. Dies erhöht entsprechend die benötigte Rechenzeit. Die im Folgenden verwendete Implementierung benutzt zwei initiale Positionen: die Lösung von LLS und den Zentroid aller Ankerpositionen.

### Linear Least Squares Multilateration (LLS)

Das NLLS-Problem kann durch Subtraktion einer der Gleichungen des Gleichungssystems 4.3 von den übrigen  $N - 1$  Gleichungen linearisiert werden. Das resultierende Gleichungssystem kann folglich algorithmisch gelöst werden, um die Position zu bestimmen.

### Min-Max

Der Min-Max-Algorithmus, auch als Bounding-Box-Algorithmus bekannt, ist im Vergleich zum rechenintensiveren LLS oder NLLS eine deutlich einfachere Methode. Ein Quadrat (Bounding-Box), gegeben durch  $[x_i - r_i, y_i - r_i] \times [x_i + r_i, y_i + r_i]$ , wird um jede Ankerposition  $(x_i, y_i)$  mit der entsprechenden Distanzmessung  $r_i$  gelegt, um den Schnitt aller Quadrate zu berechnen. Das Zentrum dieses Schnitts stellt die geschätzte Position des unbekanntes Knotens dar. Der Schnitt wird berechnet durch:

$$\left[ \max_{1 \leq i \leq N} (x_i - r_i), \max_{1 \leq i \leq N} (y_i - r_i) \right] \times \left[ \min_{1 \leq i \leq N} (x_i + r_i), \min_{1 \leq i \leq N} (y_i + r_i) \right] \quad (4.5)$$

Weitere Varianten des Algorithmus verbessern die Genauigkeit [161, 162]. Hier werden den Distanzen oder den Ecken einer Box meist Gewichte zugeordnet, um die Schätzung zu verbessern.

### Adapted Multi-Lateration (AML)

Ähnlich der Multilateration und Geo-n schätzt AML die Position durch Kreisschnittpunkte. Eine gemessene Distanz gibt dabei den Radius eines Kreises um die Koordinaten des Ankers, zu dem diese Entfernung bestimmt wurde, an. AML besteht aus drei Schritten: Bestimmung der Kreisschnittpunkte und Elimination, einer ersten Schätzung und einer Verfeinerung der Lösung. Zuerst werden zwei sich schneidende Distanzkreise gewählt. Falls hier zwei Schnittpunkte existieren, wird der mit größerer Distanz zu einem dritten Anker entfernt. Im ersten Schätzschritt wird der übrige Schnittpunkt zum Mittelpunkt der Linie zwischen seiner ursprünglichen Position und dem am nächsten gelegenen Punkt auf dem Distanzkreis des dritten Ankers verschoben. Dies dient zur Reduzierung

des Distanzfehlers. Die Lokalisierung kann nun durch weitere, bisher nicht genutzte, Distanzmessungen zu anderen Ankern, wie im zweiten Schritt oben, verfeinert werden.

#### Iterative Clustering-based Localization Algorithm (ICLA)

ICLA bildet Cluster aus allen verfügbaren Schnittpunkten, um eine Position ebenfalls in drei Schritten zu schätzen. Zunächst werden alle Schnittpunkte der Distanzkreise berechnet, von denen angenommen wird, dass sie um den unbekanntem Knoten herum angeordnet sind. Alle Schnittpunkte werden mit gleichem Gewicht initialisiert. Als nächstes wird das Iterative-Clustering-Model (ICM) zur Bestimmung der repräsentativen Schnittpunkte angewandt. Alle Schnittpunkte werden iterativ in ihrer Bewegungsrichtung verschoben, bei Aufeinandertreffen miteinander verschmolzen und das Gewicht entsprechend hochgestuft. Die zum Verschmelzen nötige Entfernung zweier Schnittpunkte entspricht der Größe des Bewegungsschritts. Weiterhin emittieren Punkte eine Anziehungskraft proportional zu ihrem Gewicht, um andere Punkte in ihrer Bewegungsrichtung zu beeinflussen. Zum Schluss werden alle Punkte im Allgemeinen zu mehreren Clustern zugeordnet. Der Ort des unbekanntem Knotens wird nun durch den Centroid der größten Schnittpunktgruppe berechnet.

#### Geo-n

Geo-n identifiziert ein repräsentatives Set von Schnittpunkten der gesuchten Position. Verfügbare Schnittpunkte werden ebenfalls Clustern zugeordnet, gefiltert und zur Berechnung eines gewichteten Centroids genutzt. Falls sich zwei Kreise nicht schneiden, wird ein virtueller Schnittpunkt mit gleichem Abstand zu beiden Kreisen auf der Geraden zwischen den Kreismittelpunkten hinzugefügt. Hier behandelt der Algorithmus Kalibrierungsfehler und mutmaßliche Fehler des Transceivers. Dann werden diejenigen Schnittpunkte im repräsentativen Set belassen, welche mindestens in  $n - 2$  Kreisen enthalten sind. Falls nun mehr als drei Schnittpunkte übrig sind, wird ein Median-Filter auf Basis der Schnittpunktabstände zu allen anderen Schnittpunkten angewandt, um diejenigen Punkte zu ent-

fernen, welche die Schätzung nicht begünstigen. Bei weniger oder gleich drei Schnittpunkten wird angenommen, dass das weitere Entfernen zu ungünstigen Schätzungen führt. Die übrigen Schnittpunkte werden durch den gewichteten Centroid ihrer selbst bestimmt.

### 4.3.3 Methodik

Es werden nun für jeden der ausgewählten Lokalisierungsalgorithmen die jeweils geschätzten Positionen berechnet, um die Lokalisierungsergebnisse anschließend zu evaluieren. Hierbei werden die Schätzungen mit der Groundtruth des RS verglichen und der Lokalisierungsfehler in einer Art Heatmap mit hinterlegtem Gebäudeplan eingetragen, ähnlich zu den Distanzfehlern in vorherigen Kapiteln. Um den Einfluss der Ankergeometrie besser zu verstehen, werden zwei verschiedene Anker setups betrachtet. Das folgende Kapitel behandelt das Szenario, in dem sich der unbekannte Knoten ausschließlich innerhalb der konvexen Hülle aller prinzipiell verfügbarer Anker bewegt. Im zweiten Szenario bewegt sich der unbekannte Knoten meist außerhalb der verfügbaren Anker.

Es werden dabei verschiedene Effekte erwartet. Eine einfache Annahme ist eine strenge Korrelation des jeweiligen Positionsfehlers mit dem räumlichen Distanzfehler, welcher in Abb. 3.8 dargestellt ist. Je nach Einfluss der LOS- und NLOS-Effekte auf den Algorithmus, wird ebenso eine Verstärkung des Positionsfehlers in den Bereichen mit großen Distanzfehlern erwartet. Weiterhin werden Unterschiede in den Laufzeiten der Algorithmen bei beiden Ankerszenarien erwartet. Hillebrandt et al. legen eine solche Annahme anhand einiger Simulationen nahe [137]. Außerdem soll die Frage beantwortet werden, ob manche im Durchschnitt nicht optimale Algorithmen in spezielleren Szenarien (bspw. unvorteilhafte Ankerkonstellationen oder deutliche NLOS-Bedingungen) ein besseres Ergebnis liefern könnten als im Durchschnitt optimale Algorithmen.

Um beide Szenarien zu vergleichen, werden die räumlichen Positionsfehlerverteilungen für jeden Algorithmus in Abb. 4.4 und Abb. 4.5 dargestellt. Tabelle 4.1 zeigt den numerischen Fehler der Algorithmen.

#### 4.3.4 Innerhalb der konvexen Hülle

Die Abb. 4.4 zeigt deutliche Unterschiede in der räumlichen Fehlerverteilung über alle Algorithmen hinweg. Der durchschnittliche Positionsfehler des LLS-Algorithmus (siehe Abb. 4.4a) liegt dabei bei 12,65 m und ist somit nicht vergleichbar mit den übrigen Algorithmen (unter 3,5 m Positionsfehler im Durchschnitt). Der Grund dieses großen Fehlers könnte sein, dass LLS sehr empfindlich gegenüber systematischen Fehlern in der Eingabe ist. Dies trifft auf das verwendete SUT zu, da bspw. die Messungen eher positive Fehler enthalten.

Die räumliche Verteilung des Fehlers zeigt den gleichen Trend für alle übrigen Algorithmen. Areale mit hohen Fehlern überschneiden sich genauso wie Areale mit niedrigen Fehlern. Auch ist ersichtlich, dass die Verteilung stark mit dem Messfehler (Abb. 3.8) korreliert. Bei genauerem Betrachten lassen sich kleine, jedoch nennenswerte Unterschiede erkennen. Sowohl AML als auch ICLA scheinen den Messfehler in der oberen Hälfte der vertikalen Flure im linken und rechten Gebäudeteil noch zu verstärken. ICLA bietet in Arealen mit relativ geringem Distanzfehler einen dazu relativ hohen Positionsfehler (sichtbar um die Position (50 m, 32 m)). Min-Max zeigt ein Verhalten, welches bspw. bereits Hillebrandt et al. nahelegen [137]. Hier vergrößert sich der Positionsfehler signifikant, sobald der unbekannte Knoten die konvexe Hülle verlässt. Dies kann auch der Fall sein, wenn einzelne Anker bei einer Messung nicht erreichbar sind. Dadurch zeigt sich dieser Effekt bereits in diesem Szenario, bspw. auf den Positionen (3 m, 38 m) und (48 m, 32 m).

Zuvor wurde die Annahme aufgestellt, dass manche im Durchschnitt schlecht abschneidende Algorithmen in manchen Bereichen besser sein könnten als im Durchschnitt gute Algorithmen, sodass sich ein Wechsel des Lokalisierungsalgorithmus je nach gegenwärtiger Ankerkonstellation empfehlen könnte. Die Auswertung zeigt hier jedoch deutlich, dass Min-Max (Abb. 4.4e) und Geo-n (Abb. 4.4f) die anderen ausgewählten Algorithmen im Positionsfehler in allen Bereichen, inklusive unvorteilhafte Ankerkonstellationen und NLOS-Bedingungen, unterbieten.

Im Gegensatz zu Kuruoglu et al. [152] zeigen die Daten bei AML in diesem Szenario in vielen Bereichen einen deutlich schlechteren Positionsfehler als beim Min-Max Algorithmus. Daraus lässt sich schließen, dass AML bei rauschenden Messungen in realen Umgebungen nicht immer so robust wie von den Autoren dokumentiert ist.

Auch ICLA zeigt in diesem Szenario einen größeren Lokalisierungsfehler als in Simu-



lationen von Haiyong et al. gezeigt [153]. Die Autoren zeigen, dass ICLA in Simulationen deutlich besser als Min-Max und dabei sowohl präzise als auch fehlertolerant ist. Dies kann unter diesen Bedingungen nicht bestätigt werden. Das könnte daran liegen, dass das Fehlermodell in der durchgeführten Simulation nicht ausreichend für den Gebrauch in (allen) Gebäuden war. In diesem Experiment liefert Geo-n die besten (Abb. 4.4f), Min-Max, mit deutlichem Abstand zu ICLA, die nächstbesten Ergebnisse (Abb. 4.4e).

### 4.3.5 Außerhalb der konvexen Hülle

Bei Lokalisierungsexperimenten innerhalb von Gebäuden wird häufig angenommen, dass sich der unbekannte Knoten innerhalb der konvexen Hülle von (erreichbaren) Ankerknoten befindet bzw. bewegt. Zwar trifft diese Annahme für vorbereitete Gebäude zu, jedoch nicht mehr zwingend für Rettungseinsätze oder andere ad-hoc aufgebaute Ankerkonstellationen. So könnte die Platzierung von Knoten bspw. auf eine Seite eines Unfallbereichs beschränkt sein. Um den Einfluss der Ankerplatzierung auf die Genauigkeit von Lokisierungsalgorithmen besser zu verstehen, werden nun einige Anker aus dem Experiment entfernt. Durch das Experimentdesign können die entsprechenden Distanzmessungen im Datensatz entfernt werden, ohne das Experiment neu durchführen zu müssen. Es werden acht Anker derart entfernt, dass der horizontale Flur und beide Areale rechts und links unterhalb des horizontalen Flurs nun außerhalb der konvexen Hülle der 17 übrigen Anker liegen.

Abbildung 4.5 zeigt die Ergebnisse dieses Szenarios. Alle ausgewählte Algorithmen werden sichtlich ungenauer in bestimmten Bereichen. Im Falle von Geo-n (Abb. 4.5f) fallen die Ergebnisse besser als erwartet aus. In Tab. 4.1 wird der MAE und der RMSE beider Szenarien verglichen. Der MAE entspricht hier dem arithmetischen Mittel. Der relative Unterschied beider Szenarien bei Geo-n liegt lediglich bei 10,7% (MAE) und 10,1% (RMSE). Im Falle von NLLS ist der Unterschied mit 4,1% (MAE) und 9,8% (RMSE) zwar noch kleiner, jedoch bietet Geo-n eine deutlich höhere Genauigkeit als alle anderen Algorithmen im Vergleich. Wie erwartet erhöht sich der Lokalisierungsfehler bei Min-Max aufgrund des Bounding-Box-Ansatzes deutlich, sodass keine sinnvolle Lokalisierung in den Bereichen außerhalb der konvexen Hülle mehr möglich ist.

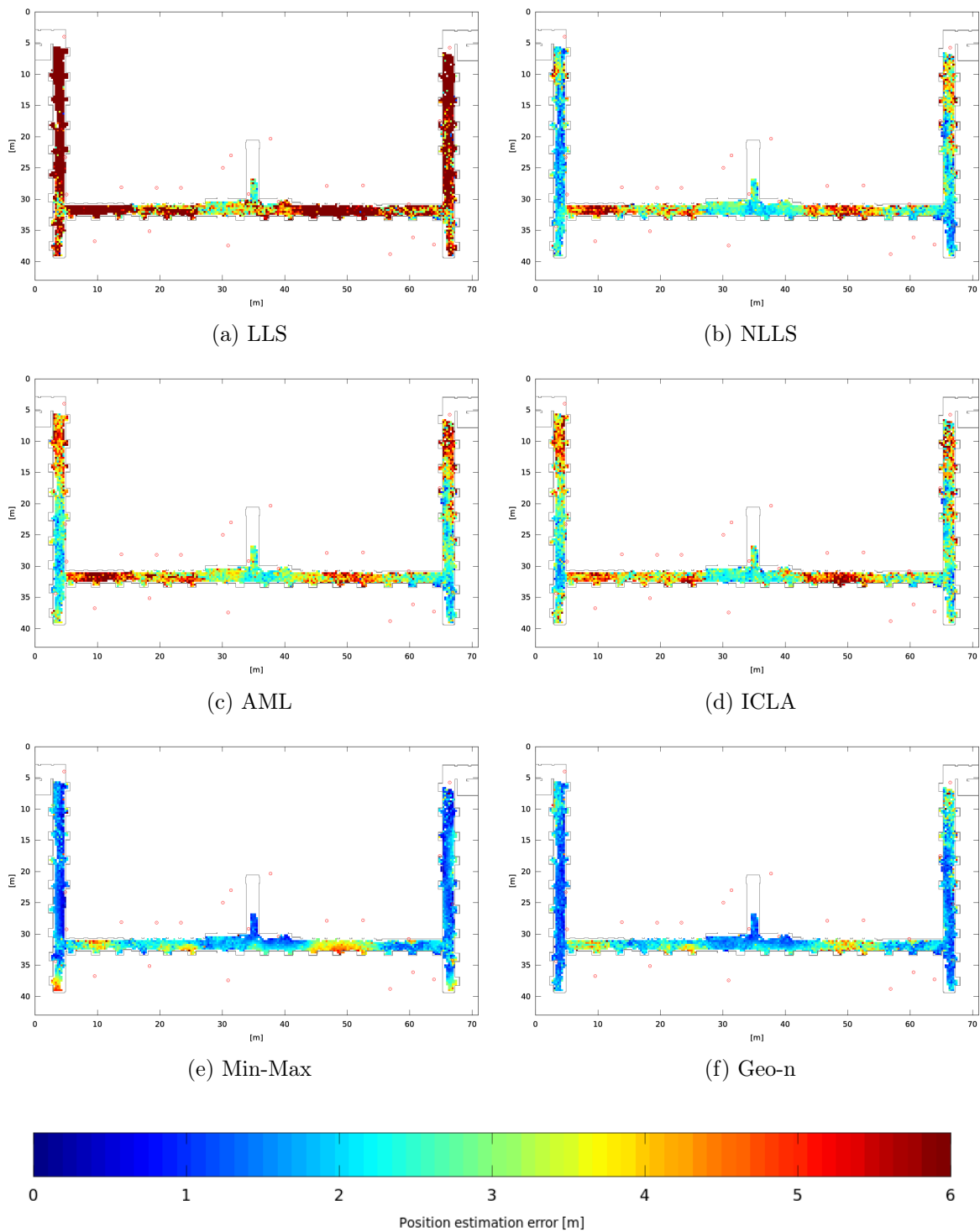


Abbildung 4.4: Räumliche Fehlerverteilung verschiedener Lokalisierungsalgorithmen im ersten Szenario. Für einen guten Vergleich der besseren Algorithmen sind höhere Fehler (bspw. Abb. 4.4a) farblich nicht mehr unterscheidbar.

Tabelle 4.1: Lokalisierungsfehler ausgewählter Algorithmen in beiden Szenarien

Algorithmus	25 Anker		17 Anker	
	MAE	RMSE	MAE	RMSE
Geo-n	1,97 m	2,47 m	2,18 m	2,72 m
Min-Max	2,05 m	2,56 m	3,07 m	3,72 m
NLLS	2,90 m	3,78 m	3,02 m	4,15 m
ICLA	3,30 m	4,41 m	3,80 m	5,44 m
AML	3,46 m	4,46 m	4,14 m	5,53 m
LLS	12,65 m	82,68 m	42,96 m	585,97 m

Eine weitere Beobachtung, auch im Vergleich mit dem ersten Szenario, ist, dass Min-Max (Abb. 4.5e) in den vertikalen Fluren oberhalb des horizontalen Flurs besser als Geo-n abschneidet. Weiterhin resultiert Bereich (45 m - 55 m, 32 m) eine bessere Lokalisierung bei AML, ICLA, NLLS und Geo-n mit reduziertem Ankereset. Dies zeigt, dass das Filtern von NLOS-behafteten Distanzwerten auch bei dann eigentlich unvorteilhafterer Ankerkonstellation von Vorteil sein kann.

#### 4.3.6 Zusammenfassung

Es wurde eine erste rigorose Evaluation der vorgestellten Lokisierungsalgorithmen in einem Gebäudeszenario unter realen Bedingungen vorgestellt. Die Ergebnisse zeigen Defizite bei reiner Simulation. Die Autoren von AML und ICLA fokussieren explizit das Lokalisierungsproblem bei Distanzfehlern, schneiden im Experiment allerdings deutlich schlechter als NLLS ab, welchen sie zu verbessern versuchen. Min-Max, der simpelste vorgestellte Algorithmus liefert vergleichsweise erstaunlich gute Ergebnisse. Geo-n produziert die niedrigsten Lokalisierungsfehler und zeigt sich räumlich betrachtet äußerst resilient gegenüber Änderungen der Ankerkonfiguration und Distanzfehlern. Eine interessante Beobachtung ist der Einfluss der Ankerkonstellation auf die räumliche Lokalisierungsgenauigkeit bei den übrigen Algorithmen. Hier sollten zukünftige Untersuchungen auch unvorteilhaftere Szenarien untersuchen, um keinen systematischen Fehler in der Aussagekraft eines Experiments zu haben.

Obwohl die Ergebnisse auch die Eigenschaften dieses Gebäudes darstellen, ist es äußerst wahrscheinlich, dass das System in ähnlichen Gebäuden vergleichbare Ergebnisse liefert. Daher schließt diese Arbeit auf einen Mangel an geeigneten Simulationsmethoden und -modellen im Kontext der Lokalisierung innerhalb von Gebäuden.

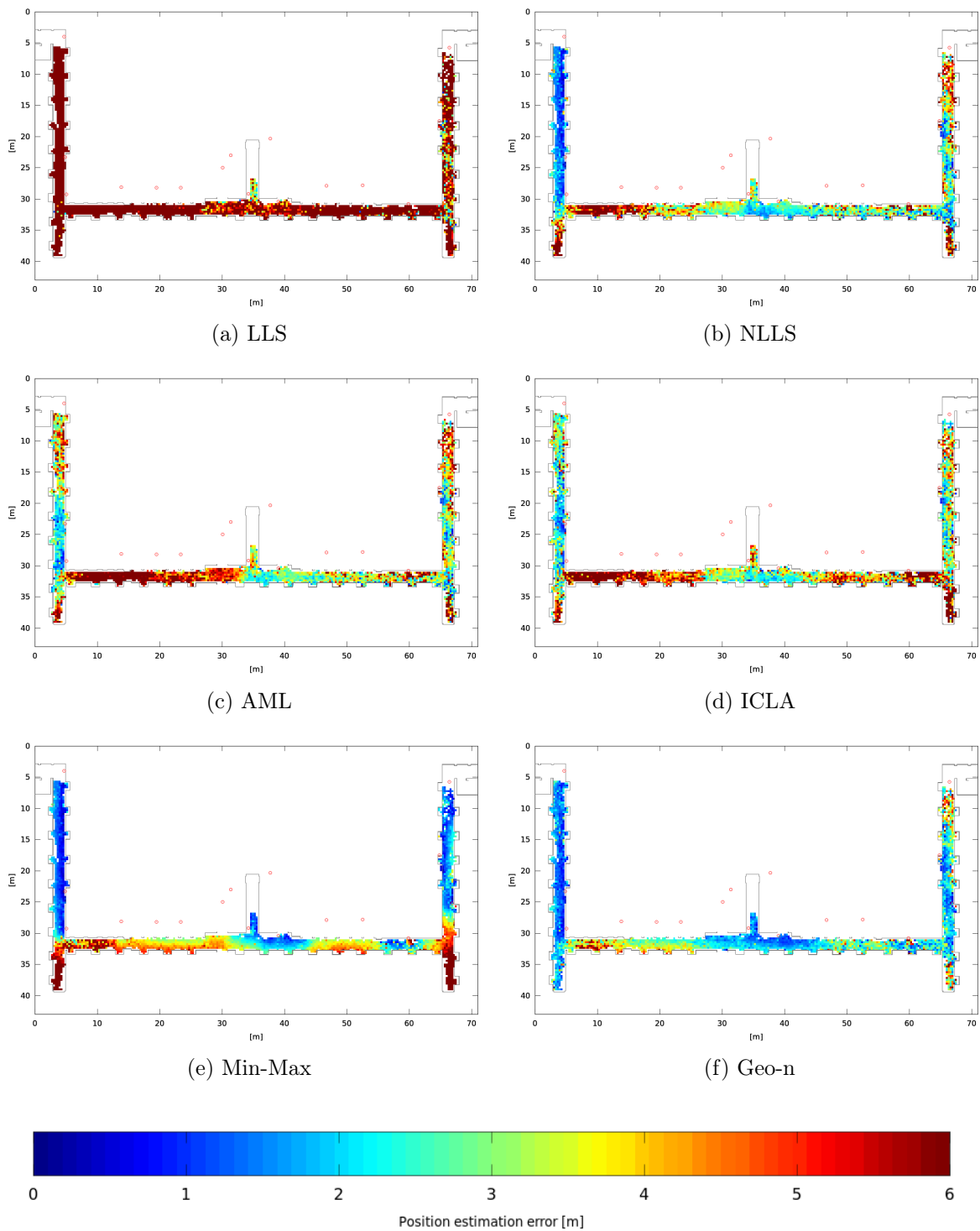


Abbildung 4.5: Räumliche Fehlerverteilung verschiedener Lokalisierungsalgorithmen im zweiten Szenario. Für einen guten Vergleich der besseren Algorithmen sind höhere Fehler (bspw. Abb. 4.4a) farblich nicht mehr unterscheidbar.

## 4.4 Vergleich mit Simulationsergebnissen

Nun werden die Erkenntnisse aus der Distanz- und Positionsfehleranalyse der vorherigen Kapitel mit einer Simulation anhand vergleichbarer Setups verglichen.

### 4.4.1 Motivation und Methodik

Ein guter, allgemein einsetzbarer Algorithmus sollte mit einem überbestimmten System aus Distanzmessungen umgehen können und gegenüber NLOS-Effekten sowie anderen Fehlern robust sein. Auch sollte ein solcher ebenso bei nicht-optimalen Ankerkonstellationen gut funktionieren. Viele vorgeschlagene Algorithmen zur Lokalisierung in Gebäuden sind in Simulationen oder kleinen Experimenten evaluiert. Die Resultate verschiedener Veröffentlichungen oder sogar innerhalb einer Veröffentlichung sind meist nur schwer miteinander vergleichbar, da Experimentdesigns und Annahmen nicht einheitlich, oft nicht vollständig beschrieben und Laborbedingungen nicht immer mit realen Umgebungen vergleichbar sind. Damit können vorgeschlagene Algorithmen auch untereinander schlecht verglichen werden, da verwendetes Fehlermodell, Dichte der Anker, deren Platzierung und die Eigenschaften des (simulierten) Kanals stark voneinander abweichen. Dadurch gestaltet sich die Vorhersage der Leistung bzw. die Übertragung auf andere Szenarien schwierig. Hillebrandt et al. zeigen hier einen rigorosen Vergleich von Algorithmen durch eine einheitliche Simulation [137].

Es werden zwei experimentell gesammelte Datensätze mit einer in Ankersetup und weiteren Parametern vergleichbaren Simulation verglichen und signifikante Unterschiede aufgezeigt. Das Ankersetup und die Fehlerverteilung in der Simulation entspricht den Bedingungen der in den Kapiteln 3.3.4 und 3.3.3 vorgestellten Datensätze (Raum und Gebäudeflur des FU-Gebäudes, 2,4 GHz-Hardware). Es werden fünf verschiedene Lokisierungsalgorithmen in beiden Szenarien verwendet und die Ergebnisse mit den experimentell gesammelten Daten verglichen. Es werden Gründe diskutiert, die zu einer Diskrepanz führen und Punkte vorgeschlagen, die während des Designs eines Algorithmus, eines realen Experiments oder einer Simulation beachtet werden sollten.

Diese Arbeit verwendet den LS<sup>2</sup> Simulator von Will et al. [154]. Die Ergebnisse sind in farbkodierten Bildern gegeben. Falls vorhanden, zeigen grüne Bereiche einen Positionsfehler, welcher kleiner als der erwartete Distanzfehler des Systems ist. Für das SUT

zeigt die Fehlerverteilung einen durchschnittlichen Distanzfehler von 2,4 m. Graue Bereiche zeigen einen Positionsfehler zwischen dem erwarteten Distanzfehler und einer Obergrenze von 25 m. Je dunkler die Bereiche dargestellt sind, desto höher ist der berechnete Fehler. Blaue Bereiche zeigen einen Fehler über 25 m. Für einen besseren Kontrast wird hier kein differenzierter Fehlerwert mehr dargestellt. Eine Änderung der Farbstufe im Bild entspricht einer Fehleränderung von 10 cm. Die roten Punkte stellen die Anker dar. Jeder Pixel entspricht dem durchschnittlichen Fehler über 800 simulierten Messungen zu allen Ankern. Um realistische Ergebnisse zu erhalten, wird jedem Distanzwert zwischen Anker und der zu bestimmenden Position ein simulierter Distanzfehler aus der dem SUT entsprechenden Fehlerverteilung addiert. Die im FU Gebäude unter realen Bedingungen beobachtete Fehlerverteilung entspricht einer Erlang-Verteilung mit  $k = 3$ ,  $\lambda = 0,522882$ , Skalierung 10 und einer Verschiebung von 3,3106 (siehe Formel 4.6).

$$f(x; k, \lambda) = 10 \frac{\lambda^k x^{k-1} e^{-\lambda x}}{(k-1)!} - 3.3106 \quad (4.6)$$

#### 4.4.2 Raumszenario

Aufgrund des theoretischen und praktischen Wissens über das SUT und die Simulation wird eine relativ homogene räumliche Verteilung des Positionsfehlers erwartet. Auch wird eine Korrelation des Fehlers mit der tatsächlichen Distanz zwischen Anker und unbekanntem Knoten erwartet. Zudem werden für verschiedene Lokalisierungsalgorithmen höchst unterschiedliche Ergebnisse bei verschiedenen Ankerkonstellationen erwartet. Die Ergebnisse der Simulation im Raumszenario durch LS<sup>2</sup> sind in Abb. 4.6 dargestellt.

Im Folgenden wird Geo-n (4.6a, 4.6c) mit Min-Max (4.6b, 4.6d) verglichen, da beide Algorithmen bereits zuvor unter nicht optimalen Ankerkonstellationen gut abschnitten. Es werden zwei verschiedene Szenarien simuliert. Abbildungen 4.6a und 4.6b zeigen vier Anker, deren Anordnung und Skalierung dem realen Experiment entsprechen. Abbildungen 4.6c und 4.6d zeigen eine gleiche Anordnung, allerdings wurden die Koordinaten mit Faktor 4 herunterskaliert. Wie bereits durch Will et al. gezeigt [48], erreicht Geo-n eine im Allgemeinen bessere Fehlerverteilung als Min-Max, wohingegen Min-Max innerhalb der konvexen Hülle aller Anker besser abschneidet. Dies stellt ein potentiell Problem für das Experimentdesign dar, da die Größe des Areal in

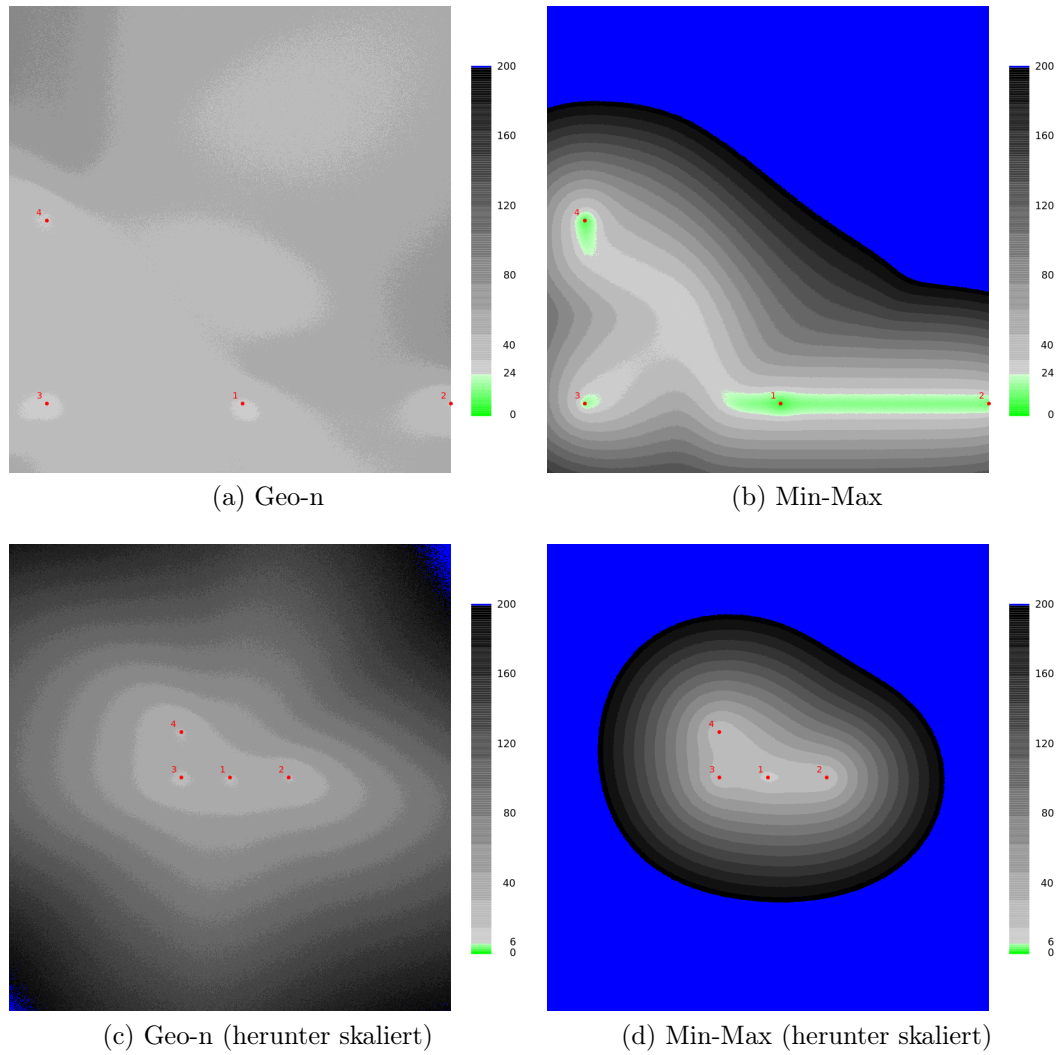


Abbildung 4.6: Durch Simulation erwartete räumliche Positionsfehlerverteilung. Fehler in [dm].

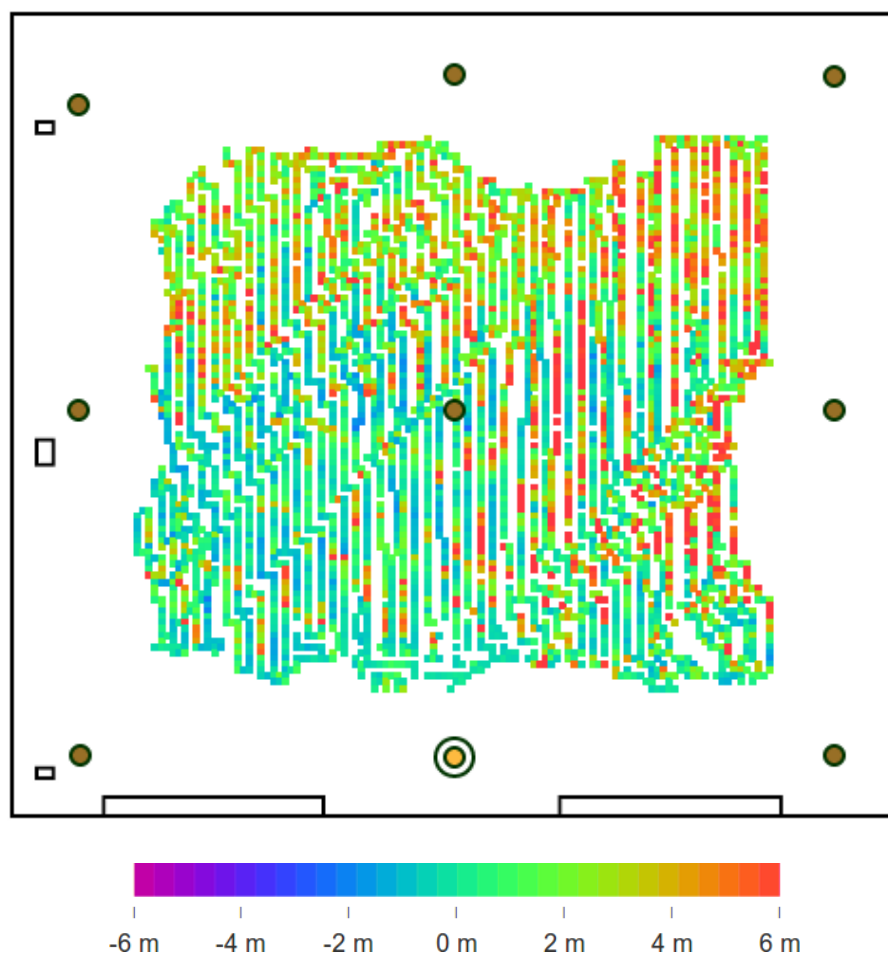


Abbildung 4.7: Räumliche Fehlerverteilung im Raumszenario für einen Anker (unten, mitte). Ein Pixel entspricht  $5\text{ cm} \times 5\text{ cm}$ .

der selben Größenordnung wie der erwartete Distanzfehler des SUT (2,4 m) liegt. Dadurch schneidet ein Algorithmus mit Bias in Richtung der konvexen Hülle oder der Anker in einem signifikanten Teil des Spielfeldes besser ab als in größeren Experimenten oder in größerer Entfernung zu den Ankern. Dies wird im Vergleich der Abb. 4.6c und 4.6d deutlich.

Abbildung 4.7 zeigt eine Karte des Raumes des realen Experiments. Die Farbkodierung zeigt den Distanzfehler am Messpunkt zum eingekreisten Anker (mitte, unten). Da das eingesetzte Messsystem mit einer nichtverstärkenden Monopolantenne ausgestattet und das Gerät am höchsten Punkt des Referenzsystems angebracht wurde, lies sich zunächst annehmen, dass der Distanzfehler unabhängig von der Fahrriichtung des



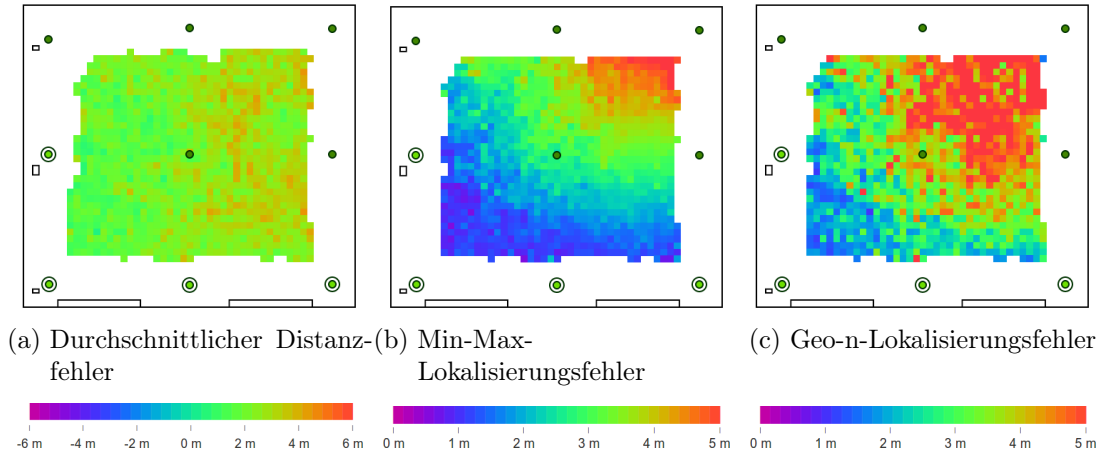


Abbildung 4.8: Räumliche Fehlerverteilungen im Raumszenario. Ein Pixel entspricht  $15\text{ cm} \times 15\text{ cm}$ .

Tabelle 4.2: Durchschnittliche Positionsfehler im Raumszenario

Algorithmus	MAE	RMSE	MAX
Geo-n	3,48 m	5,20 m	19,22 m
Min-Max	2,40 m	2,71 m	7,64 m

Roboters ist. Bei genauem Betrachten sind in Abb. 4.7 große Fehlerunterschiede bei unmittelbar benachbarten Pfaden erkennbar. Bei genaueren Tests ließ sich eine Abhängigkeit in der Ausrichtung zwischen unbekanntem Knoten und Anker bestätigen. Durch Verwendung einer Monopolantenne wurde angenommen, dass die Signalausbreitung hinreichend gleichförmig in der horizontalen Ebene stattfindet. Mögliche Gründe für die Abweichungen könnte die Funkcharakteristik des Geräts sein. Diese Ergebnisse zeigen, dass eine Aufzeichnung in der Fläche unerlässlich ist, um ein System ganzheitlich zu betrachten. Insbesondere von Interesse ist dies, wenn die Ausrichtung der installierten Hardware nicht beeinflusst werden kann.

Die räumliche Fehlerverteilungen von Min-Max (Abb. 4.8b) und Geo-n (Abb. 4.8c) in dieser nicht optimalen Ankerkonstellation zeigen im Vergleich mit der Simulation (Abb. 4.6a und 4.6b) eine nur beschränkte Aussagekraft dieser. Da Min-Max nur Positionen innerhalb der konvexen Hülle liefert, erhöht sich der Positionsfehler in Richtung der oberen rechten Ecke wie erwartet stetig. Dies trifft im Experiment jedoch auch auf Geo-n zu, obwohl die Simulation eine um Größenordnungen homogenere Verteilung schätzt. Tabelle 4.2 zeigt (wie die Simulation), dass Min-Max hier besser abschneidet. Demnach scheint die generelle Einordnung der Algorithmen durch  $LS^2$

dem Experiment zu entsprechen, die räumliche Fehlerverteilung unterscheidet sich jedoch deutlich.

Der wahrscheinlichste Grund hierfür stellt die Experimentumgebung selbst dar. Bei Einschränkung auf nur einen Raum entsprechen die Messwerte nicht mehr zwingend der Fehlerverteilung für das gesamte Gebäude. Der Effekt der alternierenden Pfade zeigt außerdem weitere Abhängigkeiten, die die Simulation nicht beachtet. In Szenarien mit aufeinanderfolgenden Messungen und nur geringer oder keiner Bewegung des unbekanntes Knotens ist eine Autokorrelation des Distanzfehlers zu erkennen. Dies wird in Simulationen nicht beachtet, da hier ein zufälliger Distanzfehler aus der genutzten Verteilung gezogen wird.

### 4.4.3 Gebäudeszenario

Wie zuvor soll die Frage beantwortet werden, wie gut Lokalisierungsalgorithmen innerhalb und außerhalb der konvexen Hülle der verfügbaren Anker funktionieren. Daher wird wie zuvor (siehe Kapitel 4.3.3) ein signifikanter Anteil der Anker in der unteren Hälfte des Gebäudes deaktiviert. Die Simulationsergebnisse durch  $LS^2$  sind in Abb. 4.9 dargestellt. Abbildung 4.4 im letzten Kapitel zeigt bereits die Lokalisierungsergebnisse für verschiedene Algorithmen und beide Ankerkonstellationen in der realen Umgebung. Tabelle 4.1 gibt einen numerischen Vergleich der Varianten wieder.

Auch in der Simulation stechen Geo-n (Abb. 4.9i) und Min-Max (Abb. 4.9g) positiv hervor. Die größte Änderung bei Ankerreduzierung ergibt sich wie erwartet bei Min-Max, da hier keine gute Lokalisierung außerhalb der Hülle möglich ist. Beachtlich ist die gute Übereinstimmung der Simulation mit dem realen Experiment bei Min-Max. Die Simulationsumgebung beachtet hier alle relevanten Parameter für die Lokalisierung mit Min-Max. Für die anderen Algorithmen scheint das nicht zuzutreffen. Sowohl NLLS als auch LLS zeigen eine weitgehend homogene Verteilung in der Simulation, was sich im realen Experiment nicht zeigt. Der wahrscheinlichste Grund hierfür liegt in der Autokorrelation der Distanzfehler, welche durch die Simulation nicht beachtet wird.

Generell ist anzunehmen, dass sich bei derartiger Reduzierung der Anker der Positionsfehler erhöhen bzw. sich die räumliche Fehlerverteilung verschlechtern sollte. Dies wurde auch in der realen Umgebung bereits gezeigt (siehe Tab. 4.1). Bei NLLS

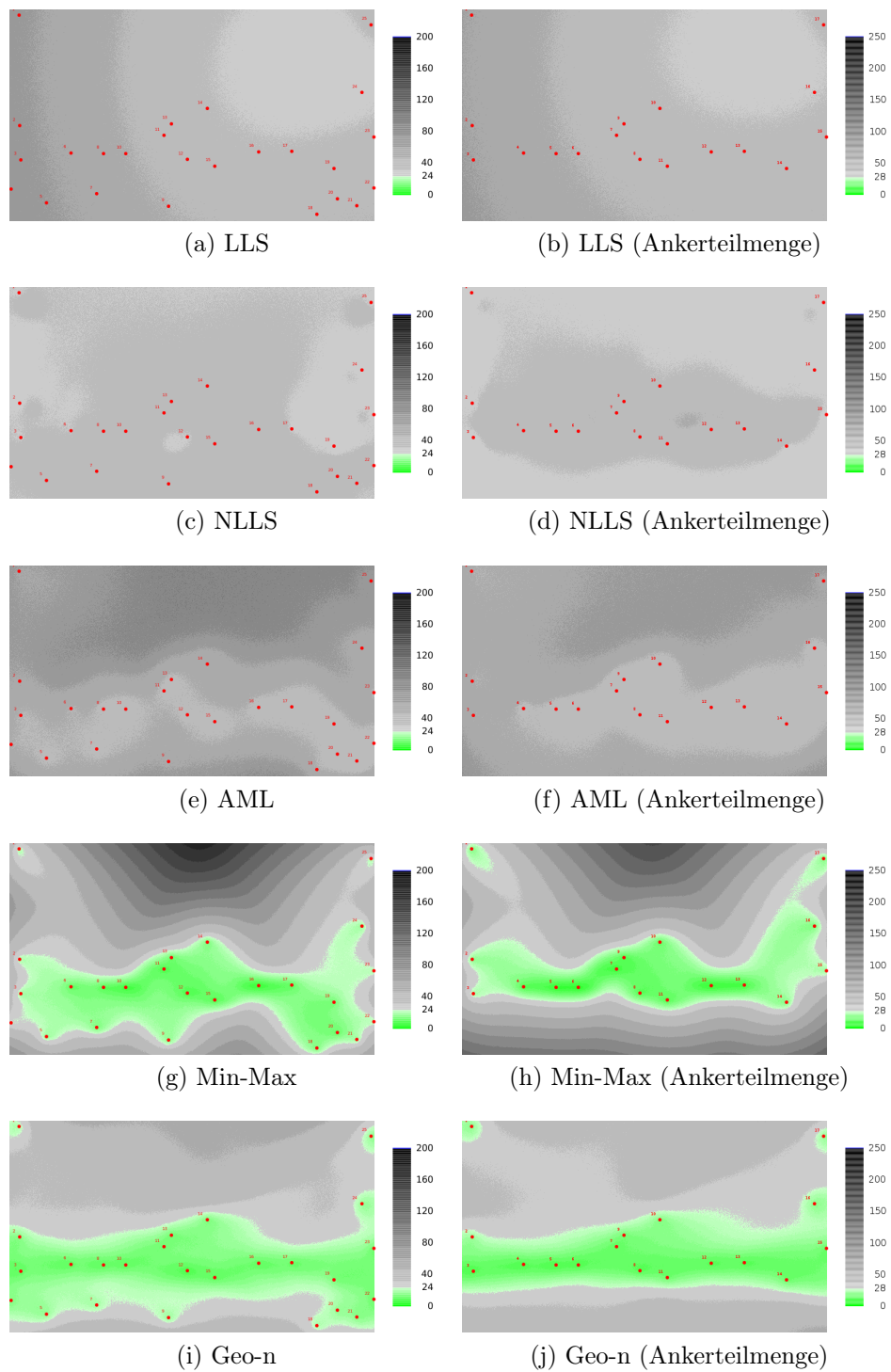


Abbildung 4.9: Durch Simulation erwartete räumliche Positionsfehlerverteilung für verschiedene Lokalisierungsalgorithmen. Fehler in [dm].

(Abb. 4.9c und 4.9d) und AML (Abb. 4.9e und 4.9f) verringert sich jedoch der simulierte Positionsfehler bei Reduzierung der Anker (Bereiche in den Abbildungen werden heller). Im realen Experiment resultiert bei LLS ein um das Vierfache erhöhter Lokalisierungsfehler bei Ankerreduzierung. Dies kann anhand der Simulation nicht nachvollzogen werden.

#### 4.4.4 Zusammenfassung

Die Ergebnisse zeigen, dass Experimente, und vor allem Simulationen, mit gut überlegten Szenarien durchgeführt werden sollten, um die relevanten Parameter zu verstehen. Obwohl eine mit Bedacht entwickelte Simulationsumgebung genutzt wird, finden sich viele durch Simulation vorhergesagte Eigenschaften nicht in der realen Welt. Andersherum zeigen die realen Experimente Eigenschaften der Algorithmen, die nur durch Simulation nicht zu erwarteten gewesen wären.

Die spezifische Ankerkonstellation kann einen hohen Einfluss auf die Lokalisierungsgenauigkeit, die räumliche Verteilung des Lokalisierungsfehlers und damit auf Aussagen über den Vergleich verschiedener Algorithmen haben. Ebenso muss die Größe des Spielfeldes mit Bedacht gewählt werden. Hier spielt neben der Reichweite des Signals und der Konstellation auch der erwartete Distanzfehler des Messsystems eine Rolle. Liegt die Größe des Spielfeldes bspw. in seiner Größenordnung, profitiert ein Algorithmus mit Bias in Richtung der konvexen Hülle bzw. der Anker.

Weiterhin erlauben die Simulationen nicht bei allen vorgestellten Algorithmen eine valide Aussage in Bezug auf die gezeigte Ankerreduktion. Dies führt zu deutlich falschen Vermutungen im Vergleich mit den realen Experimentergebnissen.

## 4.5 Zusammenfassung

Diese Arbeit schlägt die folgenden Aspekte vor, welche bei der Evaluation eines Lokalisierungssystems und/oder Algorithmus beachtet werden sollten:

**Größe des Spielfeldes:** Falls die Größe des Spielfeldes in der Größenordnung des erwarteten Distanzfehlers liegt, könnten manche Algorithmen unverhältnismäßig davon profitieren, falls sie Positionen innerhalb der konvexen Hülle oder in der Nähe von Ankern bevorzugen, wobei sie in größeren Szenarien deutlich schlechter abschneiden

würden.

**Ankerkonstellation:** Die räumliche Verteilung der Anker sollte auch nicht optimale Gebiete abbilden, sodass die Leistung eines Algorithmus nicht ausschließlich in besonders guten Konstellationen gemessen wird.

**LOS/NLOS:** In realistischen Umgebungen werden immer NLOS-Effekte auftreten, welche unter Umständen auch Vielfache der tatsächlichen Distanz zur wahren Distanz addieren können. Da vorgeschlagene Lokisierungsalgorithmen gegenüber physikalischen Effekten robust sein sollten, müssen sie auch ausreichend unter eben diesen evaluiert werden. Dazu zählt auch das Betrachten der verwendeten Materialien der Anwendungsumgebung, da die Funkausbreitung dadurch erheblich beeinflusst sein kann.

**Geräteinschränkungen:** Vor der Durchführung eines Experiments oder einer Simulation sollten alle relevanten Parameter des Messsystems bekannt sein. Parameter können zudem zeitabhängig sein (bspw. Temperatur, Auslastung des Funkkanals etc.), oder von der Ausrichtung des Gerätes abhängen.

**Autokorrelation der Distanzfehler:** Ein Hauptaugenmerk sollte auf die Autokorrelation der Distanzwerte und damit der Distanzfehler gelegt werden. Ein hoher Distanzfehler erhöht demnach die Wahrscheinlichkeit weiterer hoher Fehler. Anders ausgedrückt lässt sich pro Anker sehr gut zwischen Gebieten verschiedener LOS- und NLOS-Klassen unterscheiden. Damit werden Simulationsergebnisse verwässert, da hier zufällig aus einer grundsätzlich oder durch Beobachtung eines bestimmten Gebietes angenommenen Fehlerverteilung gezogen wird, ohne auf lokal konsistente Eigenschaften der tatsächlichen Umgebung einzugehen.

Diese Arbeit regt weiter die Nutzung von standardisierten, ausführlichen Experimenten mit verschiedenen und auch nicht intuitiven Ankerkonstellationen auch außerhalb der konvexen Hülle als Referenzdatensets an. Allerdings muss auch hier der Kontext der Anwendung während der späteren Lokalisierung betrachtet werden. So spielt bspw. ein zu lokalisierender Mensch, welcher das Gerät am Körper (oder weit davon entfernt) trägt, eine Rolle im Lokalisierungsprozess.

Die aufgezeigten Untersuchungen lassen den Schluss zu, dass Feldtests zwingend nötig sind, um Eigenschaften und Einsatzmöglichkeiten eines Lokalisierungsalgorithmus zu verstehen.

---

## KAPITEL 5

---

# Erzeugung von Routing Graphen

Im Folgenden wird ein Verfahren vorgestellt, das mittels regelbasierter Bildverarbeitung Gebäudestrukturen aus Gebäudeplänen extrahiert. Dieses Kapitel basiert auf folgenden eigenen Veröffentlichungen:

- S. Schmitt, L. Zech, T. Willemsen, H. Sternberg und M. Kyas, “Conditional erosion to estimate routing graph out of floor plans”, in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Okt. 2016. DOI: 10.1109/IPIN.2016.7743602
- S. Schmitt, L. Zech, K. Wolter, T. Willemsen, H. Sternberg und M. Kyas, “Fast Routing Graph Extraction from Floor Plans”, in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sep. 2017

### 5.1 Motivation

Jede Lösung zur Navigation lässt sich ganz allgemein in drei Komponenten einteilen: Positionsschätzung, Wegfindung und Visualisierung. Im Kontext der Indoorlokalisierung ist eine Positionsschätzung durch die Abwesenheit eines GNSS-Signals meist schwierig.

Mögliche Lösungen können nach Blankenbach et al. [163] in infrastrukturbasiert, visuelle, autonome und hybride Systeme kategorisiert werden. Infrastrukturbasierte Verfahren leiden unter meist hohen Installations- und wiederkehrenden Wartungskosten, wie bspw. der Energieversorgung, dem Austausch von defekten Geräten oder Batte-

rien. Visuelle Verfahren schätzen eine Position durch den Abgleich aufgenommener Bilder mit bereits bekanntem Wissen über die Umgebung oder mittels Merkmalsextraktion und Abgleich konsekutiver Bilder. Diese Verfahren benötigen zwar hohe Rechenkapazitäten, können jedoch, ähnlich der infrastrukturbasierten Verfahren, eine hohe Genauigkeit erreichen. Gängige autonome Verfahren, sog. INSS bspw. mit Barometern (siehe auch PDR), leiden unter prinzipbedingten Ungenauigkeiten und einem daraus resultierenden Drift. Die nötigen Positionskorrekturen können durch Informationsfusion mit anderen Lokalisierungssystemen (hybrider Ansatz) oder anhand zuvor bekanntem Wissen über die Umgebung berechnet werden. Dies kann ebenso die Implementierungskosten reduzieren.

Eine Informationsquelle, welche bspw. in öffentlichen Gebäuden oft frei zugänglich vorhanden ist, sind Gebäudepläne. Da zur Aufgabe eines Lokalisierungssystems neben der Schätzung einer Position meist auch eine Navigation im Gebäude und eine Visualisierung des Ortes gehört, ist eine Karte zur Orientierung des Benutzers meist ebenso nötig. Wenn digitale Gebäudepläne nicht vorhanden sind, können sie von Flucht- und Rettungsplänen extrahiert werden [164]. Sternberg et al. stellen eine alternative Methode zur 3D-Modellierung eines Gebäudes mittels eines Lasers vor [165], wobei ein 2D-Gebäudeplan aus horizontalen Schnitten der Punktwolken erstellt werden kann.

Während sich solche Pläne gut zur Visualisierung eignen, sind sie in ihrer unverarbeiteten Form nicht optimal für eine Wegfindung. Ein extrahierter Graph, welcher die generellen Laufwege und Abzweigungen bzw. nur die Struktur des Gebäudes und möglicher Laufwege modelliert, reduziert Kosten zur Wegfindung und kann ebenso zur Positionsschätzung beitragen. Eine Möglichkeit ist der Abgleich von inertialbasierten Positionsschätzungen. Willemsen et al. präsentieren hierfür bereits eine Lösung [166]. Positionsschätzungen finden hier lediglich auf dem Graphen statt. Dies führt unweigerlich zu einer höheren Ungenauigkeit, je weiter sich das zu lokalisierende Gerät vom Graphen entfernt. Jedoch erlaubt dieser Ansatz im Grundgedanken eine raumgenaue Lokalisierung. Dieser Vorteil kommt durch einen wesentlich reduzierten oder gar vollständig eliminierten Drift und geringeren Rechenkosten zustande. Willemsen et al. erstellen einen solchen Graphen für Untersuchungen händisch. Für die weitere Anwendung und Implementierung einer jeden Wegfindungslösung ist eine weitgehend automatisierte Extraktion aus vorhandenen Plänen wünschenswert.

Weitere Anwendungen solcher Datenstrukturen sind die Wegfindung und das Treffen strategischer Entscheidungen auf digitalen Spielkarten. Halldórsson und Björns-

son zeigen eine Dekomposition einer Karte, um bspw. die Entscheidungsfindung einer künstlichen Intelligenz zu beschleunigen [167]. Dieser Ansatz detektiert sog. Gateways (Türen) zwischen sinnvoll getrennten Regionen (Räumen) einer Karte. Ein Graph könnte hiervon mit Pfadplanungsalgorithmen extrahiert werden. Uriarte und Ontañón zeigen ebenso einen Algorithmus zur Graphenextraktion [168]. Die zugrundeliegende Technik verkleinert iterativ ein Voronoi-Diagramm der freien Räume. Die nötige Rechenleistung zur Reduzierung der Verästelungen hängt von deren Anzahl ab. Dies führt zu hohen Rechenanforderungen bei hochaufgelösten Karten mit komplexen Gebäudestrukturen.

Daneben existieren Anwendungsfelder in den Bereichen Medizin, Neurowissenschaften, Biologie und Kartographie, bspw. die Extraktion von Gehirnstrukturen, Blutadern und -gefäßen, Atmungssystem, Filament-Strukturen, Rissen in Gebäudewänden oder die Straßen- und Flusseextraktion aus Luftbildern [169]. Sironi et al. kategorisieren dabei in *manuell-definierte (engl. hand-designed) Filter*, *angelernte (engl. learned) Filter*, *Konturerkennung (engl. contour-detection)* und *deep-learning basierte Architekturen*. Die in dieser Arbeit vorgestellte Methode fällt dabei in die erste Kategorie.

In vielen Anwendungen ist neben der zu extrahierenden Struktur auch deren Breite von Interesse, um bspw. Arterienverkalkung während der Koronarangiographie festzustellen. Diese Arbeit wird später darstellen, dass der im Folgenden vorgestellte Algorithmus leicht erweitert werden kann, um diese Anforderung zu erfüllen.

Im Folgenden wird explizit die Graphenextraktion aus Gebäudekarten betrachtet. Die Argumentationen und Zielsetzungen dieser Arbeit können mit anderen Anwendungsfeldern übereinstimmen, müssen es jedoch nicht. In Anlehnung an Willemsen et al. wird hier die extrahierte Datenstruktur *Routing-Graph* genannt. Die Namensgebung stellt dabei eine mögliche Anwendung dar.

## 5.2 Allgemeine Anforderungen

Der Graphenextraktion liegen die folgenden Anforderungen zugrunde:

- 1. Die Eingabe ist im 2D-Rasterformat mit hinreichender Auflösung.**

Digitale Gebäudepläne existieren meist in 2D-Vektor- oder Rasterformat. Während für erstere geometrische Operationen nötig sind, können für letztere klassische, pixelbasierte Bildverarbeitungsmechanismen angewandt werden. Da beide



Formate ineinander konvertiert werden können, ist die weitere Herangehensweise nicht vom Format abhängig. Daher wird explizit das Rasterformat als Eingabe gewählt. Das Problem der Diskretisierung des Vektorformats soll hierbei nicht beachtet werden. Es wird davon ausgegangen, dass die Eingabe in hinreichend hoher Qualität zur Verfügung steht, damit resultierende Pixelkoordinaten mit hinreichender Auflösung in metrische Koordinaten umgewandelt werden können. Eine Auflösung von bspw. 20 cm pro Pixel im Rasterbild hat eine systematische Ortsungenauigkeit der Graphenstruktur von maximal 20 cm zur Folge.

## 2. Ein Pixel besitzt nur zwei mögliche Zustände:

- belegt (bspw. Wand), oder
- frei bzw. begehbar.

Typische Gebäudepläne zeigen Räume und Korridore. Je nachdem, was die Quelle der Eingabe ist, könnte diese mit konstruktionsrelevanten Informationen oder Einrichtungsgegenständen (wie bspw. Öffnungsrichtung von Türen, Sicherheitsinformationen, Raumbeschreibungen) angereichert sein. Die Natur des im Folgenden vorgestellten Ansatzes gibt vor, dass lediglich freie Bereiche und Wände bzw. nicht betretbare Bereiche in der Eingabe unterschieden werden dürfen. Daher muss die Eingabe binär (bspw. schwarz-weiß) kodiert und alle entsprechend nicht als Hindernis gewünschte Objekte entfernt worden sein. Dies kann ggf. einfach bewerkstelligt werden, falls die Daten zuvor als Computer-Aided Design (CAD) vorlagen. Falls derartige Informationen im Rasterformat enthalten sind, müssen weitere Bildverarbeitungsmechanismen angewandt werden, welche nicht Teil dieser Arbeit sind. Da Laufwege üblicherweise nicht über Objekte wie Tische, Stühle oder Schränke hinwegführen, sollten solche, sofern vorhanden, nicht aus der Karte entfernt werden. Durch schwarze Pixel abgeschlossene Bereiche führen zu nicht verbundenen Teilbäumen. Da keine Metainformationen (bspw. Raumannotationen) benutzt werden, kann der Algorithmus die Teilbäume nicht selbst bewerten und ggf. aussortieren. Daher wird dieses Problem explizit nicht behandelt, sodass auch kleine Teilbäume im Resultat verbleiben.

## 3. Der resultierende Graph bildet die Mitte von freien Räumen ab.

Das übergeordnete Ziel ist die Extraktion einer Graphenstruktur, welche den

Gebäudeplan in einer Art und Weise repräsentiert, die zur Positionsschätzung, Wegplanung und/oder Navigation von Menschen oder Robotern geeignet ist. Einerseits führt dies zur Anforderung, möglichst wenige Kanten zu produzieren, um effizientere Anwendungen zu ermöglichen. Andererseits sollten mögliche bzw. übliche Laufwege möglichst gut abgebildet werden, um Positionsschätzungen zu unterstützen. Diese Arbeit fokussiert in diesem Aspekt auf die Verbesserung der Positionsschätzung und versucht explizit nicht, Kanten zu vermeiden. Nichtsdestotrotz ist dies in einem nachgelagerten Schritt vorstellbar. Da die Definition *üblicher Laufwege* nicht in jedem Fall trivial ist, wird die Anforderung derart eingeschränkt, dass Wege mindestens in der Mitte von Fluren und Räumen gefunden werden sollen. Es ist anzunehmen, dass dies ohne kontextuelles Wissen über die Umgebung und deren Nutzungsart einer nahezu optimalen Lösung entspricht. Das *Raten* üblicher Laufwege innerhalb großer Räume würde eine nachgeschaltete Positionsschätzung auf dem Graphen zu sehr beeinflussen und zusätzlichen Aufwand für den Algorithmus bedeuten, daher werden derartige Lösungen an dieser Stelle explizit vermieden. Dies hat zur Folge, dass der Graph mindestens die Mitte eines Bereiches traversieren soll, um diesen hinreichend zu beschreiben. Es wird hierbei implizit angenommen, dass alle Wege bidirektional beschriftet werden können. Zwei aneinander vorbei gehende Personen bewegen sich also, sofern es den Graphen betrifft, auf der selben Kante.

### 5.3 Conditional Erosion Algorithm (CEA)

Im Folgenden wird der Algorithmus Schritt für Schritt eingeführt und Verbesserungen werden auf dem Weg vorgeschlagen. Die Eingabe sei ein zweidimensionales, binäres Bild mit weißen und schwarzen Pixeln, welche freien und belegten Raum darstellen. Die Grundidee des Algorithmus ist die Erosion weißen Raumes bis zu dem Punkt, an dem eine ein Pixel breite, weiße Linie zwischen den ursprünglich schwarzen Pixeln übrig bleibt. Diese stellt dann einen möglichen Graphen dar. Abhängig vom Pixelabstand zwischen den Strukturen können im Allgemeinen mehrere Lösungen existieren.

### 5.3.1 Iterative Erosion freien Raumes

Es sei

$$p_i^t = \begin{cases} \text{true} & \text{falls Pixel } i \text{ zum Zeitpunkt } t \text{ weiß ist} \\ \text{false} & \text{falls Pixel } i \text{ zum Zeitpunkt } t \text{ schwarz ist} \end{cases} \quad (5.1)$$

der Zustand des Pixel  $i$  zum Zeitpunkt  $t$ . Für jeden weißen Pixel ( $p_i^t = \text{true}$ ) wird eine möglichst einfach zu berechnende binäre Metrik  $K$  gesucht um zu entscheiden, ob dieser auf schwarz (false) gesetzt werden soll oder (ggf. noch) nicht. Schwarze Pixel bleiben dabei für immer schwarz; es wird also weißer Raum erodiert, aber niemals dilatiert. Die Grundidee des Algorithmus ist, für diese Metrik lediglich die acht Nachbarpixel  $N_i^t$  von  $p_i^t$  zu betrachten, sprich eine lokale Entscheidung zu treffen. Eine solche kann für jeden weißen Pixel sequentiell getroffen werden. Wird ein Pixel überprüft und trifft die Metrik nicht zu, wird er ggf. später nochmals überprüft. Der weiße Pixel  $p_i^t$  wird also geschwärzt, wenn die Belegung seiner Nachbarn einem vordefinierten Muster  $k_j \in K$  entsprechen:

$$p_i^{t_1} = \text{true} \wedge N_i^{t_1} \in K \Rightarrow p_i^{t_2} = \text{false} \quad (5.2)$$

Damit das Resultat die Mitte eines Flurs repräsentiert, sollte eine Wand um höchstens einen Pixel pro Iteration in jede Richtung dilatieren. Ohne Beschränkung der Allgemeinheit wird im linken oberen Pixel des Bildes begonnen und spaltenweise fortgefahren. Wenn nun, wie zuvor definiert, für Pixel  $p_i^t$  seine Nachbarn  $N_i^t$  zum selben Zeitpunkt  $t$  betrachtet werden, könnte eine Wand von links nach rechts mehr als einen Pixel oder gar über die Mitte des Flurs hinaus dilatieren. Das liegt daran, dass eine Dilatation die Belegung für den nachfolgend zu überprüfenden Pixel so verändert, dass dieser ebenfalls geschwärzt wird. Da die resultierende weiße Linie in der Mitte des Flurs sein soll, müssen zunächst spätere Pixel im Bild ebenso überprüft werden. Daher wird die Formel 5.2 derart abgeändert, dass zwei Überprüfungen stattfinden. Der Buchstabe  $m$  steht nun für eine Iteration bzw. deren Ergebnis:

$$p_i^m = \text{true} \wedge N_i^m \in K \wedge N_i^{m-1} \in K \Rightarrow p_i^m = \text{false} \quad (5.3)$$

Es erfolgt also eine Überprüfung des Pixels  $p$  in der aktuellen Iteration ( $m$ ), und gleichermaßen im Zustand am Ende der letzten Iteration ( $m - 1$ ).

Die in  $K$  enthaltenen Muster werden im weiteren Verlauf und in Anlehnung an gängige

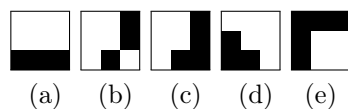


Abbildung 5.1: Ausgewählte Kernel (Beispiele).

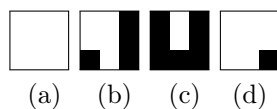


Abbildung 5.2: Beispielkernel, welche mindestens ein Kriterium verletzen. Kernel 5.2a verletzt Krit. 1), 5.2b verletzt Krit. 2), Kernel 5.2c verletzt Krit. 3) und Kernel 5.2d verletzt Krit. 4).

Bildverarbeitungstechniken *Kernel* genannt. Es werden zunächst händisch 120 von möglichen  $2^8 = 256$  Kernen ausgewählt. Die übrigen 136 verletzen mindestens eines der folgenden Kriterien. Ein Kernel soll

- 1) keine Wand erschaffen,
- 2) zwei weiße benachbarte Bereiche nicht trennen,
- 3) eine weiße Linie nicht verkleinern bzw. einzelne Pixel nicht löschen
- 4) und an Ecken nicht erweitern.

Eine durchgehende weiße Linie wird derart definiert, dass sich Linienpixel an ihrer Kante, nicht jedoch an ihren Ecken berühren dürfen, um als benachbart zu gelten. Da nach dieser Definition die möglichen Zustände einer Linie welche durch einen Kernel führt eingeschränkt sind, wurde so die Auswahl der Kernel vereinfacht. Der entstehende Pixelgraph unterliegt folglich dieser Tatsache. Abbildung 5.1 zeigt einige ausgewählte, Abb. 5.2 einige aussortierte Kernel, die ein Kriterium verletzen. Dabei ist zu beachten, dass der Kernel die acht Nachbarn, exklusiv der Mitte, beschreibt. Der mittlere Pixel ist der zu überprüfende im Bild.

Zum Betrachten der Randpixel des Bildes ist eine Definition jenseits des Bildes nötig. Entsprechend des Schemas kann der Rand als Wand oder freier Raum betrachtet werden. Falls er als Wand angesehen wird, erodiert weißer Raum zusätzlich von außen nach innen. Dies führt zu Kanten außerhalb des Gebäudes, um ebendieses herum. Da jedoch keinerlei Wissen über die Umgebung außerhalb der Wände des Gebäudeplans vorausgesetzt wird, wird in dieser Arbeit der Rand des Bildes als freier Raum

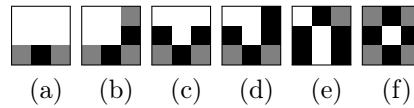


Abbildung 5.3: Repräsentanten mit drei Zuständen: weiß, schwarz und irrelevant (grau). Gruppen 5.3a bis 5.3c erlauben jeweils vier Rotationen. Gruppen 5.3d und 5.3e erlauben zusätzlich Reflexionen, folglich acht Möglichkeiten. Die Gruppe 5.3f besteht nur aus einem Kernel, welcher nicht rotiert oder gespiegelt werden kann.

angesehen.

Da eine möglichst einfache Metrik gefunden werden soll, sollte die Menge der benötigten Kernel möglichst klein gehalten werden. Um diese weiter zu reduzieren, werden nun Pixel gesucht, welche nicht zur Entscheidungsfindung beitragen. Wenn sich zum Beispiel zwei Kernel um genau einen Pixel unterscheiden, ist dieser irrelevant. Daher ergibt es Sinn, stattdessen nur einen Kernel zu verwenden und einen dritten Pixelzustand einzuführen. Abbildung 5.3 zeigt Repräsentanten aller sechs Gruppen ausgewählter Kernel. Dies reduziert die Anzahl der Kernel auf 29.

Der folgende Algorithmus zeigt die vollständige bisherige Vorgehensweise:

**Daten :** Rastergrafik  $A$

**Ergebnis :** Erodierte Rastergrafik  $A$

```

do
  B ← A
  für jedes Pixel  $p = (x, y)$  tue
    wenn  $N_p^A \in K \wedge N_p^B \in K$  dann
      A ← Schwärzen( $p$ )
    Ende
  Ende
while  $A \neq B$ 

```

**Algorithmus 1 :** Einfache Version des Conditional Erosion Algorithm (CEA).

Dies ist im Sinne dieser Arbeit ein vollständiges Verfahren, um die Erosion durchzuführen. Um zusätzlich die Breite von Strukturen (Flurbreite, Arterienbreite usw.) zu schätzen, könnte ein zusätzliches Datenfeld pro Pixel ausreichen, welches nach jeder Iteration den kürzesten Pixelabstand zu einer tatsächlichen Wand beschreibt. Da dies eine Erweiterung darstellt, wurde die Idee hier nicht weiter verfolgt.

### 5.3.2 Formulierung als Gleichungssystem

Um mit den ausgewählten Kernen effizienter zu arbeiten, werden diese in boolesche Formeln umgewandelt. Die Menge der Nachbarpixel  $N_i$  des Pixels  $p_i$ , numeriert von 1 bis 8, ist wie folgt definiert:

$$N_i = \left\{ p_i^j \mid p_i^j \in \begin{pmatrix} p_i^1 & p_i^2 & p_i^3 \\ p_i^4 & & p_i^5 \\ p_i^6 & p_i^7 & p_i^8 \end{pmatrix} \right\} \quad (5.4)$$

Die folgende Matrix beschreibt nun einen allgemeinen Kernel  $k$ , wobei  $k^j$  dem gewünschten Pixelzustand des numerierten Nachbarpixels  $j$  entspricht (d.h. weiß, schwarz oder irrelevant). Der Pixel  $c$  stellt die Mitte des Kernels dar, also den Pixel, auf den der Kernel angewandt wird.

$$k = \begin{pmatrix} k^1 & k^2 & k^3 \\ k^4 & c & k^5 \\ k^6 & k^7 & k^8 \end{pmatrix} \quad (5.5)$$

Gleichung 5.6 wendet den beispielhaften Kernel  $k_e$  auf die Nachbarschaft  $N_i$  an:

$$k_e(N_i) = \begin{array}{|c|c|c|} \hline \square & \blacksquare & \blacksquare \\ \hline \blacksquare & \blacksquare & \blacksquare \\ \hline \blacksquare & \blacksquare & \blacksquare \\ \hline \end{array} = \begin{pmatrix} p_i^1 & p_i^2 & \text{true} \\ p_i^4 & \text{true} & \overline{p_i^5} \\ \text{true} & \overline{p_i^7} & \text{true} \end{pmatrix} \quad (5.6)$$

Die Einträge in  $k_e(N_i)$ , welche bereits wahr (true) sind, repräsentieren irrelevante Pixel. Da der Kernel auf  $p_i$  (im Zentrum des Kernels) angewandt wird, ist dieser im Kernel ebenfalls irrelevant.

Ob ein Kernel  $k$  anwendbar ist oder nicht, gibt die Konjunktion über den Kernel wieder:

$$f_k = \bigwedge_{i=1}^8 k^i = \begin{cases} \text{true} & \text{falls Kernel } k \text{ anwendbar ist} \\ \text{false} & \text{falls Kernel } k \text{ nicht anwendbar ist} \end{cases} \quad (5.7)$$

Da die Funktionen  $f_k$  boolescher Natur sind, kann durch Disjunktion beschrieben werden, ob mindestens ein Kernel auf die Umgebung von  $p_i$  passt:

$$f(N_i) = \bigvee_k f_k(N_i) = \bigvee_k \bigwedge_{j=1}^8 k^j(N_i) \quad (5.8)$$

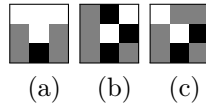


Abbildung 5.4: Repräsentanten der vereinfachten Gruppen mit drei Zuständen: weiß, schwarz und irrelevant (grau). Gruppen 5.4a und 5.4c erlauben jeweils vier Rotationen, Gruppe 5.4b erlaubt zusätzlich Reflexionen, also acht Möglichkeiten.

Nach obigen Regeln ausformuliert, kann diese Formel vereinfacht werden<sup>1</sup>. Es ergeben sich nach obiger Definition in vereinfachter Form 17 Klauseln mit 124 möglichen Belegungen. Diese Vereinfachung ist hier dargestellt:

$$\begin{aligned}
 f(N_i) = & \left( p_i^1 \wedge p_i^2 \wedge p_i^3 \wedge \overline{p_i^7} \right) \vee \left( p_i^1 \wedge \overline{p_i^2} \wedge \overline{p_i^4} \wedge \overline{p_i^5} \right) \vee \\
 & \left( p_i^1 \wedge \overline{p_i^2} \wedge \overline{p_i^4} \wedge \overline{p_i^7} \right) \vee \left( p_i^1 \wedge p_i^4 \wedge \overline{p_i^5} \wedge p_i^6 \right) \vee \\
 & \left( \overline{p_i^2} \wedge p_i^3 \wedge \overline{p_i^4} \wedge \overline{p_i^5} \right) \vee \left( \overline{p_i^2} \wedge p_i^3 \wedge \overline{p_i^5} \wedge \overline{p_i^7} \right) \vee \\
 & \left( \overline{p_i^2} \wedge \overline{p_i^4} \wedge p_i^6 \wedge \overline{p_i^7} \right) \vee \left( \overline{p_i^2} \wedge \overline{p_i^5} \wedge \overline{p_i^7} \wedge p_i^8 \right) \vee \\
 & \left( \overline{p_i^2} \wedge p_i^6 \wedge p_i^7 \wedge p_i^8 \right) \vee \left( p_i^3 \wedge \overline{p_i^4} \wedge p_i^5 \wedge p_i^8 \right) \vee \\
 & \left( \overline{p_i^4} \wedge \overline{p_i^5} \wedge p_i^6 \wedge \overline{p_i^7} \right) \vee \left( \overline{p_i^4} \wedge \overline{p_i^5} \wedge \overline{p_i^7} \wedge p_i^8 \right) \vee \\
 & \left( p_i^1 \wedge \overline{p_i^5} \wedge \overline{p_i^7} \right) \vee \left( \overline{p_i^2} \wedge \overline{p_i^4} \wedge p_i^8 \right) \vee \\
 & \left( \overline{p_i^2} \wedge \overline{p_i^5} \wedge p_i^6 \right) \vee \left( p_i^3 \wedge \overline{p_i^4} \wedge \overline{p_i^7} \right) \vee \\
 & \left( \overline{p_i^2} \wedge \overline{p_i^4} \wedge \overline{p_i^5} \wedge \overline{p_i^7} \right)
 \end{aligned} \tag{5.9}$$

Die Darstellung als disjunktive Normalform (DNF) zeigt, dass die Klauseln ebenso als Kernel verstanden und visualisiert werden können. Die 17 Klauseln entsprechen folglich 17 Kernen. Abbildung 5.4 zeigt die visuelle Gruppierung der übrigen Klauseln in drei Klassen.

<sup>1</sup>Durchgeführt mit Wolfram Mathematica 11.1.1.0, BooleanMinimize-Funktion, <http://reference.wolfram.com/language/ref/BooleanMinimize.html>

### 5.3.3 Weitere Reduzierung der Kernel

Zuvor wurde definiert, dass verschiedene weiße Bereiche explizit durch schwarze Pixel voneinander getrennt werden sollen, sodass sich weiße Pixel immer an Kanten, niemals an Ecken, berühren. Bei Missachtung dieser Definition lässt sich erkennen, dass die Gruppe mit Repräsentant 5.4b mit nur einer Einschränkung in der Gruppe mit Repräsentant 5.4c inbegriffen ist. Letztere limitiert den Pixel  $p^1$  auf weiß, wohingegen Erstere hier einen irrelevanten Pixel besitzt. Die beiden weiteren unterschiedlichen Pixel ( $p^2$  und  $p^3$ ) sind in Repräsentant 5.4c irrelevant. Wenn nun  $p^1$  im Bild schwarz wäre, würde nur noch Repräsentant 5.4b in Frage kommen. Der Unterschied zwischen beiden Repräsentanten kann in diesem Sinne argumentativ also auf  $p^1$  reduziert werden. Da der Fokus auf der Reduzierung der Kernel liegt, entfernen wird im Folgenden die Gruppe mit Repräsentant 5.4b entfernt, da diese neben Rotationen auch Reflexionen erlaubt und somit acht statt vier Kernel repräsentiert. Dadurch treffen die Kernel im Falle  $p^1 = \text{false}$  nicht mehr zu. Des Weiteren wird der Kreuzkernel (Abb. 5.3f) entfernt, da dieser nicht wesentlich zur Erosion beiträgt. Dies reduziert die Anzahl der Kernel insgesamt auf acht. Die so reduzierte Funktion  $f$  lautet nun:

$$\begin{aligned}
 f(N_i) = & \left( p_i^1 \wedge p_i^2 \wedge p_i^3 \wedge \overline{p_i^7} \right) \vee \left( p_i^1 \wedge p_i^4 \wedge \overline{p_i^5} \wedge p_i^6 \right) \vee \\
 & \left( \overline{p_i^2} \wedge p_i^6 \wedge p_i^7 \wedge p_i^8 \right) \vee \left( p_i^3 \wedge \overline{p_i^4} \wedge p_i^5 \wedge p_i^8 \right) \vee \\
 & \left( p_i^1 \wedge \overline{p_i^5} \wedge \overline{p_i^7} \right) \vee \left( \overline{p_i^2} \wedge \overline{p_i^4} \wedge p_i^8 \right) \vee \\
 & \left( \overline{p_i^2} \wedge \overline{p_i^5} \wedge p_i^6 \right) \vee \left( p_i^3 \wedge \overline{p_i^4} \wedge \overline{p_i^7} \right)
 \end{aligned} \tag{5.10}$$

Die Abbildung 5.5 zeigt beispielhaft die Differenz im Resultat der beiden Kernelversionen. Es lassen sich nur geringfügige Unterschiede erkennen. Der blaue Graph zeigt die Anwendung der 17 Kernel. Zusätzliche Graphstrukturen bei Anwendung der Version mit nur acht Kernen sind rot überlagert. Im Detail ist ersichtlich, dass die 17 Kernel manche Strukturen zerstückeln, wohingegen die acht Kernel diese weitgehend intakt lassen.

Um besser zu verstehen, warum die Kernel entfernt werden dürfen, wird in Abb. 5.6 die relative Häufigkeit der Treffer pro Kernel dargestellt. Diese gruppiert die 17 zuvor definierten Kernel in zwei Gruppen. Die Kernel 1, 4, 9, 10, 13, 14, 15 und 16 liegen im Bereich zwischen 2, 4 (Minimum) und 26, 3 % (Maximum) Anwendungsquote. Die



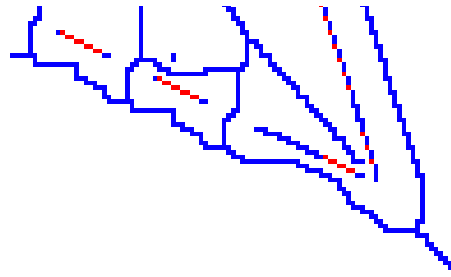


Abbildung 5.5: Ausschnitt aus einem erodierten Gebäudeplan. Der blaue Graph zeigt die Erosion mit 17 Kernen. In rot sind die zusätzlich übrig bleibenden Pixel bei der Erosion mit nur acht Kernen dargestellt.

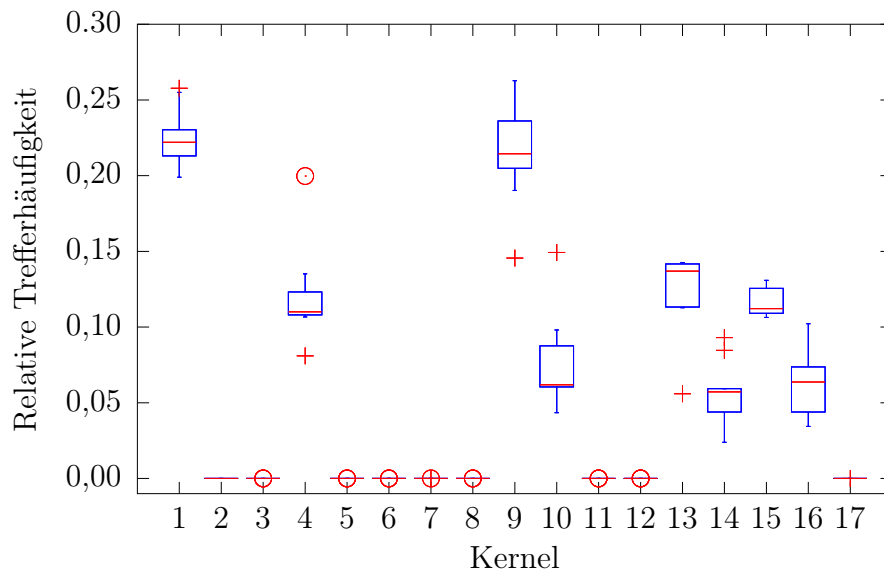


Abbildung 5.6: Relative Häufigkeit der Treffer pro Kernel. Die Nummerierung entspricht der Reihenfolge in den Formeln 5.9 und 5.10.

übrigen Kernel kommen hier lediglich auf 0,0 und 0,006%. Letztere finden also nur relativ selten Anwendung.

Jede weitere Reduktion führt zu signifikant anderen Ergebnissen, welche obige Bedingungen an einen Graphen weiter verletzen. Die bisherige Extraktion zielte auf eine ortsbezogene Lokalisierung. Wenn jedoch nur die grobe Struktur, Anzahl der Verzweigungen und dergleichen interessiert, müssten Wände nicht zwingend gleichmäßig dilatieren; der extrahierte Graph müsste nicht in der Mitte von Fluren verlaufen. Dafür werden im Folgenden die Kernel derart reduziert, dass nur noch zwei statt vier Rotationen erlaubt sind. Dadurch dilatieren Wände aus einer Ecke des Bildes in die andere. Abbildung 5.7 zeigt die vier verwendeten Kernel in diesem Szenario. Die hier

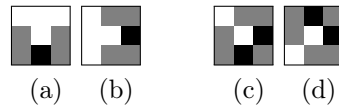


Abbildung 5.7: Weiter reduzierte Kernmenge, welche nur zwei Rotationen erlaubt.

verwendete Variante der möglichen Formeln lautet:

$$f(N_i) = \left( p_i^1 \wedge p_i^2 \wedge p_i^3 \wedge \overline{p_i^7} \right) \vee \left( p_i^1 \wedge p_i^4 \wedge \overline{p_i^5} \wedge p_i^6 \right) \vee \left( p_i^1 \wedge \overline{p_i^5} \wedge \overline{p_i^7} \right) \vee \left( \overline{p_i^2} \wedge \overline{p_i^5} \wedge p_i^6 \right) \quad (5.11)$$

#### 5.3.4 Liste weißer Pixel

Der bisher definierte Algorithmus iteriert über alle Pixel des Bildes. Zur Steigerung der Effizienz wird eine Liste weißer Pixel (WPL) eingeführt, die fortan iterieren wird.

$$\text{WPL} = \{p_i \mid p_i = \text{true}\} \quad (5.12)$$

Diese Liste wird bei Start initialisiert. Jeder Eintrag enthält die Koordinaten eines weißen Pixels im Bild. Der Eintrag wird entfernt, sobald  $p_i = \text{false}$  gesetzt wird. Sobald in einer Iteration der Liste kein Eintrag entfernt werden kann, ist der Algorithmus beendet. Die übrigen Einträge stellen den extrahierten Graphen dar. Sofern die Liste in gleicher Reihenfolge traversiert wird, wie die Iteration über das Bild durchgeführt wurde, führt diese Vorgehensweise zum selben Resultat und ist dabei effizienter.

Da bisher nur bekannt ist, welche Pixel noch weiß sind, wird die gesamte Liste WPL iteriert. Ein Kernel passt jedoch ausschließlich dann möglicherweise auf  $p_i$ , wenn in der vorherigen Iteration ein Nachbarpixel geschwärzt wurde oder der Algorithmus zuvor initialisiert wurde. Falls dies nicht der Fall ist, kann  $p_i$  auch jetzt nicht geschwärzt werden. Daher wird hier eine zweite Liste verwendet. Die ursprüngliche, WPL, hält nun alle Pixel, welche in dieser Iteration geprüft werden müssen, sprich deren Nachbarn sich in der letzten Iteration geändert haben. Der erste Eintrag wird aus WPL entfernt und die Kernel auf dieser Position geprüft. Die Nachbarn der nun geschwärzten Pixel werden in eine zweite Liste,  $\text{WPL}_{\text{next}}$ , eingefügt. Diese wird in der

nächsten Iteration abgearbeitet. Die aktuelle Iteration ist vorbei, sobald WPL leer ist. Falls Einträge in  $WPL_{\text{next}}$  existieren, beginnt die nächste Iteration mit ausgetauschten Listen WPL und  $WPL_{\text{next}}$ . Wenn dies nicht der Fall ist, ist der Algorithmus beendet.

Ein Pixel  $p_i$  kann zu einem Zeitpunkt also nur in einer oder keiner der beiden Listen enthalten sein. Wenn er in WPL enthalten ist, wird er noch in der aktuellen Iteration geprüft. Wenn nun ein anderer Pixel  $p_j \in N_i$  geschwärzt wird, noch bevor  $p_i$  geprüft wird, dann wird  $p_i$  schlicht später in der Iteration geprüft. Falls  $p_j$  erst geprüft wird, nachdem  $p_i$  entfernt wurde ohne ihn zu schwärzen, wird  $p_i$  (evtl. erneut) in  $WPL_{\text{next}}$  eingefügt. Dies zeigt bereits die Abhängigkeit zur Sortierung der Pixel in den Listen. Ohne Beschränkung der Allgemeinheit werden die Pixel spaltenweise nach Koordinaten von oben links bis unten rechts sortiert. Mit anderen Worten: Falls  $p_i = \text{false}$  gesetzt wird, wird  $N_i$  aufgeteilt:

$$N_i^< = \{p \in N_i \mid p < p_i\} \quad (5.13)$$

$$N_i^> = \{p \in N_i \mid p > p_i\} \quad (5.14)$$

Dabei ist die Relation  $<$  auf der Menge aller Pixel eine strenge Totalordnung. Wenn nun  $p_i$  geprüft wird, kann die Menge  $N_i^<$  nun nicht mehr in WPL enthalten sein, da sie dann entsprechend der Sortierung bereits vor  $p_i$  geprüft worden wäre. Diese Pixel müssen bei Schwärzung von  $p_i$  in der nächsten Iteration geprüft werden, sprich sie müssen dann in  $WPL_{\text{next}}$  eingefügt werden.

Einträge in  $N_i^>$  können unabhängig von  $p_i$  in WPL enthalten sein. Die, die dort nicht enthalten sind, müssen bei Schwärzung von  $p_i$  ebenfalls in  $WPL_{\text{next}}$  eingefügt werden. Man könnte hier argumentieren, dass  $N_i^>$  bei Schwärzung von  $p_i$  direkt in WPL eingefügt werden könnten, um so die Anzahl der benötigten Iterationen zu reduzieren. Allerdings würde dies der Bedingung, Pixel, bspw. auf beiden Seiten eines Flurs, gleichwertig zu behandeln, entgegen stehen. Eine andere Sichtweise ist, dass das Nichtbeachten der Sortierung an dieser Stelle dazu führt, dass die Prüfung wichtige Änderungen, welche noch nicht stattgefunden haben, noch nicht sieht.

### 5.3.5 Implementierung und vollständiger Algorithmus

Der Algorithmus wird in *C++* implementiert und verfügbare Compiler Optimierungen<sup>2</sup> ausgenutzt. Es werden *std::priority\_queue* und *std::vector* Container für Listen und *boost::tuple* für Bildkoordinaten verwendet. Die Sortierung erfolgt zuerst anhand der horizontalen, dann anhand der vertikalen Koordinate. Das Iterieren über eine solche Liste bedeutet also eine spaltenweise Traversierung des Bildes von links oben nach rechts unten.

Die Existenz eines Tupels mit Bildkoordinaten beschreibt einen weißen Pixel. Sofern ein Pixel geschwärzt wird, wird das Tupel gelöscht. Bei Initialisierung werden alle weißen Pixel in Form eines Zeigers auf ein Tupel in WPL eingefügt. Um nach Schwärzung eines Pixels die Selektierung von weißen Nachbarn, welche in der nächsten Iteration geprüft werden müssen, weiter zu beschleunigen, wird eine Nachbartabelle NLT (engl. für *neighbor lookup table*) eingeführt. Diese besitzt die gleichen Dimensionen wie das Eingabebild und beinhaltet ebenso Zeiger auf die selben Tupel. Sofern ein Pixel bereits schwarz ist, ist der entsprechende Zeiger *NULL*, sodass keine weitere Abfrage nötig ist. Die Struktur wird gleichzeitig mit WPL initialisiert. Beim Schwärzen eines Pixels können so die weißen Nachbarn, in Form eines Zeigers, in *WPL<sub>next</sub>* eingefügt werden. Die Koordinate in NLT wird dabei schlicht auf *NULL* gesetzt.

Algorithmus 2 zeigt den vollständigen Algorithmus zur Erosion freien Raumes. Wie zuvor diskutiert, wurde die Eingabe möglicherweise vorverarbeitet. Das resultierende Rasterbild besitzt die selben Dimensionen wie die Eingabe.

Durch das Arbeiten auf Pixelebene ist die Laufzeit des Algorithmus zwar nicht notwendigerweise von der Auflösung der Eingabe abhängig, jedoch direkt proportional zur Anzahl der weißen Pixel im Bild. Dadurch entstehen bei hochaufgelösten Eingaben zwangsläufig unbefriedigende Wartezeiten, bis das Ergebnis zur Verfügung steht. Eine geschickte Parallelisierung des Algorithmus kann die wahrgenommene Laufzeit verringern. Eine mögliche Parallelisierung kann auf verschiedenen Ebenen betrachtet werden. Zunächst wird die Vorgehensweise in Bezug auf einzelne Pixel diskutiert.

Wenn der Rand des Bildes als weißer Raum betrachtet wird, besitzt jeder Bildpixel acht Nachbarn. In der ersten Iteration wird ein weißer Pixel überprüft, ohne

---

<sup>2</sup> GCC-Option *-O3*

```

Daten : Rastergrafik  $A$ 
Ergebnis : Erodierte Rastergrafik  $A$ 

/* Initialisierung */
WPL  $\leftarrow$  {}
WPLnext  $\leftarrow$  {}
für alle Pixel  $p = (x, y) \in A$  tue
  wenn Pixel  $p$  weiß ist dann
     $p.b \leftarrow$  true /*  $p$  ist in WPLnext */
    WPLnext  $\leftarrow$  WPLnext  $\cup$  Pointer( $p$ )
    NLT( $p.x, p.y$ )  $\leftarrow$  Pointer( $p$ )
  Ende
Ende

/* Erosion */
solange WPLnext nicht leer ist tue
  /* Tausche WPL und WPLnext */
  WPL  $\leftarrow$  WPLnext
  WPLnext  $\leftarrow$  {}
  B  $\leftarrow$  A

  für jedes Pixel  $p = (x, y, b) \in$  WPL tue
    WPL  $\leftarrow$  WPL  $\setminus$   $p$ 
     $p.b \leftarrow$  false
    wenn  $N_p^A \in K \wedge N_p^B \in K$  dann
      A  $\leftarrow$  Schwärzen( $p$ )
      NLT( $p.x, p.y$ )  $\leftarrow$  NULL

      /* Füge alle weiße Nachbarn WPLnext hinzu */
      W  $\leftarrow$  ( $p | p \in N_p^A \wedge p.b =$  false)
      für alle Pixel  $p = (x, y, b) \in W$  tue  $p.b =$  true
      WPLnext  $\leftarrow$  WPLnext  $\cup$  W
    Ende
  Ende
Ende

```

**Algorithmus 2** : Vollständiger Conditional Erosion Algorithm (CEA).

dass dies durch einen Nachbarpixel angestoßen wurde. Falls hierbei keine Schwärzung erfolgt, gibt es noch maximal acht Nachbarpixel, welche bei Schwärzung zu einer erneuten Überprüfung führen. Daher kann ein weißer Pixel insgesamt maximal neunmal überprüft werden. Da die möglichen Überprüfungen hierbei voneinander abhängen, erscheint eine Parallelisierung auf dieser Ebene nicht sinnvoll. Da der extrahierte Graph die Mitte des Flures abbilden soll, könnten bspw. zwei verschiedene Wände parallel dilatiert werden. Ein weißer Pixel ist jedoch auch hier stets abhängig von allen Nachbarn. Da lokal nicht entschieden werden kann, ob die Mitte des Flures erreicht ist, muss jede Überprüfung auch den aktuellen Zustand der Nachbarpixel, und damit den Zustand der Dilatation der anderen Wand, mit betrachten. Dies führt zu unverhältnismäßig hohem Synchronisationsaufwand.

Für die nächste Parallelisierungsmöglichkeit wird die bisherige Abbruchbedingung diskutiert: das Erreichen der Mitte eines freien Bereiches. Dieser ist durch keine weiteren Treffer der Kernel auf einem weißen Pixel bzw. dessen acht Nachbarn definiert. Man könnte die Erosion bzw. Dilatation nun für eine Wand oder einen Bildausschnitt durchführen, ohne diese Abbruchbedingung zu betrachten. Dazu stelle man sich eine Wand vor, die dilatiert, bis keine Kernel mehr zutreffen. Dies entspricht nun einer allgemeineren Abbruchbedingung als obige, da die Mitte explizit nicht betrachtet und über sie hinweg dilatiert wird. Zusätzlich wird der Abstand zur Wand pro Pixel nach einer passenden Abstandsmetrik gespeichert. Solange sich die Resultate der parallel arbeitenden Threads überlappen, kann so das Gesamtergebnis durch das Verschmelzen über eine Minimum-Funktion erstellt werden. Dadurch wird die direkte Abhängigkeit zweier Threads eliminiert, jedoch gleichzeitig und deutlich der Aufwand eines einzelnen Threads erhöht. Wenn diese Variante nach jeder Iteration synchronisiert wird, sodass Threads eine Iteration unabhängig voneinander arbeiten können, wird der Synchronisationsaufwand auf die Ränder der Eingaben eines jeden Threads reduziert. In einer Iteration dilatieren Wände um maximal einen Pixel. Wenn alle Threads nach jeder Iteration abgeglichen werden, wird der Fehler der durch diese Art der Parallelisierung zwangsläufig entsteht minimiert. Ohne eine solche Barriere würden Pixel jenseits eines Bereichs nicht beachtet, sodass sich Kanten im Graphen am Rande zweier Bereiche nicht unbedingt an einem Punkt treffen würden. Es muss jedoch beachtet werden, dass dieses Verfahren dennoch die globale Reihenfolge der Pixel missachtet. Eine einfache Implementierung ist die Aufteilung der Eingabe in Streifen. Das Eingabebild wird in horizontale Streifen geteilt wobei jeder Thread einen Streifen bearbeitet. Die generelle Vorgehensweise zeigt Algorithmus 3. Jedem Thread wird ein

statischer Streifen des Eingabebildes zugeordnet. Innerhalb einer Iteration arbeiten die Threads unabhängig voneinander parallel. Danach wird das Bild aktualisiert und die nächste Iteration beginnt. Da Pixel nicht gleichberechtigt erodieren, bildet das Resultat im Allgemeinen nicht mehr die Mitte von Fluren oder Räumen ab.

**Daten** : Rastergrafik  $A$

**Ergebnis** : Erodierte Rastergrafik  $A$

**wiederhole**

```

     $S \leftarrow$  Teile Bild in Streifen
    /* Arbeite eine Iteration an jedem Streifen parallel          */
     $S = \text{parallel\_erode}(S)$ 
    /* Aktualisiere Bild                                          */
     $A = S$ 
bis keine Änderung in  $A$  mehr eintritt

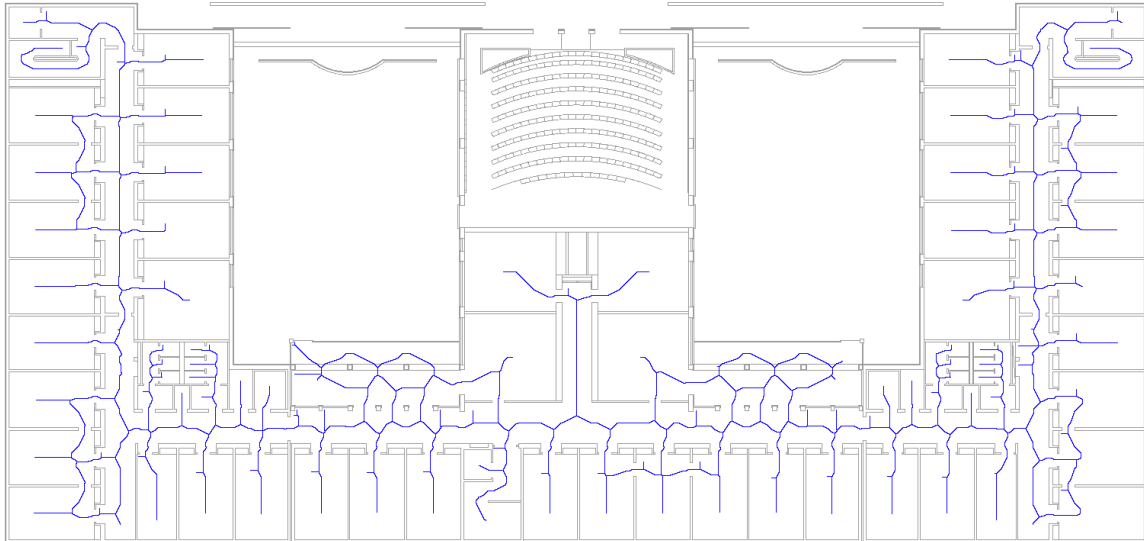
```

**Algorithmus 3** : Parallelisierung der Erosion.

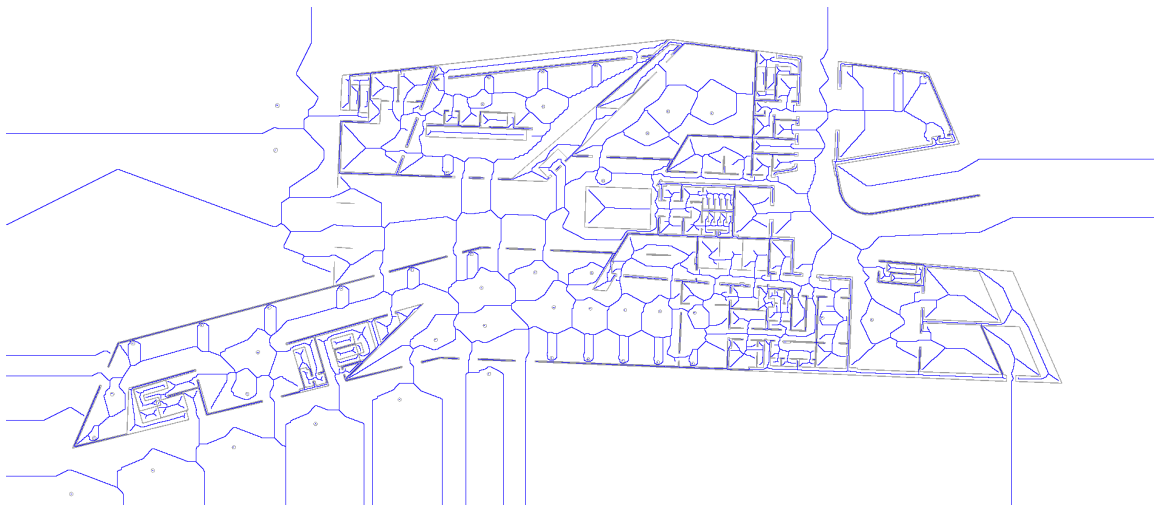
## 5.4 Ergebnisse und Diskussion

Zur Demonstration und Laufzeitevaluation werden Gebäudepläne der FU und der HafenCity Universität (HCU) genutzt. Diese enthalten enge Raumstrukturen und auch offenere Bereiche. Die Abb. 5.8a und 5.8b zeigen sowohl die Eingabe (unterliegende Gebäudestruktur) als auch den resultierenden Routing-Graphen (blau).

Abbildung 5.8a zeigt das FU-Gebäude. Im Gebäudeplan finden sich überwiegend regelmäßige Strukturen in Form von Büros. Die Eingabe wurde insofern vorverarbeitet, dass nicht betretbare Bereiche als Wände markiert wurden. Dadurch ergeben sich nur in Fluren, Treppenhäusern und Räumen Graphenstrukturen. Der Graph bildet vor allem im linken und rechten Flur des Gebäudes mit der Mitte die intendierten, angenommenen Gehwege ab. Im horizontalen Flur fällt die nicht geradlinige Struktur auf. Grund dafür ist die unsymmetrische Struktur des Flurs. In Räumen erreicht der Graph bis auf wenige Ausnahmen die Raummitte. Der Ursprung dieser Ausnahmen, ist nicht direkt bekannt, tritt aber an vielen Stellen in verschiedenen Karten auf. Der Effekt lässt sich so beschreiben, dass eigentlich intendierte Graphenstrukturen aufgrund der Kernausswahl nicht korrekt gebildet werden. Allerdings bildet der Graph dennoch in allen Fällen den Flur ab und betritt angrenzende Räume. Da Laufwege



(a) Erodiertes Gebäudeplan des FU-Gebäudes.



(b) Erodiertes Gebäudeplan des HCU-Gebäudes.

Abbildung 5.8: Gebäudepläne mit resultierendem Routing-Graphen.



in Räumen ohne weiteres Wissen grundsätzlich nicht automatisch abgeleitet werden können, beschränkt das Resultat die Intention der Arbeit nicht.

Abbildung 5.8b zeigt das HCU-Gebäude, welches ein grundsätzlich anderes Layout als das FU-Gebäude besitzt. Die Eingabe wurde hier nicht vorverarbeitet. In den Räumen resultiert ein verzweigterer Graph. Es lässt sich vermuten, dass sich aufgrund der schrägen Wände nicht die gleichen Situationen ergeben, die in der FU-Gebäudekarte zu nicht entstehenden Graphenstrukturen führen. In Fluren zeigt sich ein weitgehend identisches Bild. Die Abhängigkeit zur Ausrichtung von Wänden verringert sich mit steigender Auflösung. Weiter fallen die Graphstrukturen außerhalb des Gebäudes auf, obwohl der Rand als weiß betrachtet wird, sodass dieser nicht nach innen dilatiert. Allerdings lässt sich bei der Anwendung der lokalen Metrik nicht ohne Weiteres entscheiden, ob diese gerade innerhalb oder außerhalb eines Gebäudes angewandt wird. Daher führen konkave äußere Gebäudestrukturen auch zu Pfaden außerhalb des Gebäudes. Bei geschlossenen Strukturen können Lösungen des *connected-component-labeling*-Problems hier helfen, dies zu vermeiden. He et al. zeigen eine Übersicht über verschiedene Herangehensweisen [170]. Bei nicht geschlossenen Gebäudestrukturen könnte die konkave Hülle des Gebäudes extrahiert und so die Flächen außerhalb des Gebäudes erkannt werden. Ebert et al. zeigen eine Übersicht über mögliche Herangehensweisen [171]. Nach der Erosion könnten so die Graphstrukturen außerhalb des Gebäudes entfernt werden. Der Gebäudeplan zeigt außerdem Säulen in einem großen Raum, welche im Graph zu Schleifen um ebendiese herum und damit auch zu mehr Kanten führen, was in Abb. 5.8b deutlich wird. Das vermag eventuell für Positionsschätzung günstig zu sein, jedoch werden auch unwahrscheinliche Gehwege gefunden, wenn Säulen in der Nähe von Wänden stehen. In großen Gebäuden kann dadurch auch die benötigte Rechenzeit für eine Wegfindung in Navigationsanwendungen deutlich höher sein.

Die gemessenen Laufzeiten werden in Tab. 5.1 präsentiert. Der Algorithmus wurde in *C++* mit Intels Threading Building Blocks (TBB) implementiert<sup>3</sup>. Alle Experimente wurden auf einer Intel<sup>®</sup> Xeon<sup>®</sup> CPU (E3-1240 V2) mit 3,4 GHz, vier Kernen und 16 GB RAM unter Ubuntu Linux 14.04.5 LTS durchgeführt. Die Tabelle zeigt die Laufzeiten der seriellen Version (Alg. 2) in der Spalte  $T_S$  und die Laufzeiten  $T_n$  bzw. den Speed-Up  $S_n$  für  $n$  Threads in der parallelisierten Version (Alg. 3). Die Zeiten der seriellen Version wurden mit *gprof*<sup>4</sup> gemessen. Sie stellt die Dauer der

<sup>3</sup>GCC-Version 4.8.4 und TBB-Version 4.2

<sup>4</sup>GNU-gprof (GNU-Binutils for Ubuntu) 2.24

Erosion ohne Initialisierung dar. Da *gprof* bei mehreren Threads nicht direkt anwendbar ist, stellt der Wert die Dauer der Erosion inklusive Verwaltungsaufgaben dar. Der Zeitunterschied zwischen  $T_S$  und  $T_1$  beinhaltet also auch das Aufteilen der Arbeitspakete auf die Threads, das Thread-Management und die Initialisierung verschiedener Datenstrukturen.  $T_1$  wurde als Basis zur Berechnung des Speed-Ups von  $T_2$ ,  $T_4$  und  $T_8$  benutzt. Es werden drei verschiedene Karten mit jeweils zwei verschiedenen Auflösungen benutzt. Der Algorithmus wird zehn mal pro Karte ausgeführt und die Durchschnittslaufzeit berechnet. Es gab keine Ausreißer in den Messreihen.

Tabelle 5.1: CEA-Laufzeiten in [ms]

Dim. [px]	Weiß [px]	$T_S$	Parallelisierung						
			$T_1$	$T_2$	$S_2$	$T_4$	$S_4$	$T_8$	$S_8$
1000 × 727	681,561	80	241.9	167.3	1.44	122.4	1.97	89.2	2.71
1000 × 727	654,103	80	240.3	171.8	1.40	112.2	2.14	89.8	2.68
1900 × 820	1,518,456	250	633.0	459.9	1.38	311.7	2.03	216.9	2.92
1900 × 820	1,503,929	210	620.2	474.0	1.31	319.3	1.94	220.1	2.82
4000 × 2906	11,390,011	2510	7394.4	5657.1	1.31	4451.8	1.66	3252.3	2.27
4000 × 2906	11,229,996	2370	7182.9	6389.8	1.12	4893.7	1.47	3326.8	2.16

Die parallele Version benutzt bis zu acht Threads. Der Speed-Up variiert mit den Karten. Bei zwei Threads liegt dieser zwischen 1,12 und 1,44, bei vier bei 1,47 und 2,14 und bei acht Threads zwischen 2,16 und 2,71. Niedrigere Speed-Ups scheinen mit ungleichen Arbeitspaketen zu korrelieren. Der Speed-Up ist in der Anzahl der Threads limitiert, da Teile des Algorithmus nicht parallelisiert werden können, wie die Aufteilung der Pixel auf verschiedene Threads und die Synchronisierung nach jeder Iteration. Amdahls Gesetz besagt, dass der Speed-Up daher konvergieren wird [172]. Ein wichtiger Aspekt ist die abnehmende Größe der Arbeitspakete der Threads. Je fortgeschrittener die Erosion ist, desto weniger weiße Pixel bleiben einem Thread pro Iteration zur Abarbeitung. Daher wäre es sinnvoll, die Anzahl der Threads an die abzuarbeitende Arbeitslast anzupassen. Mit der statisch gewählten Anzahl an Threads wird an dieser Stelle nur die generelle Machbarkeit des Ansatzes gezeigt. Die derartige Verwendung mehrerer Threads produziert teilweise kammartige Artefakte, sprich zusätzliche Kanten im Graphen, siehe Abb. 5.9. Dies liegt daran, dass die Reihenfolge der Erosion derart nicht gewährleistet ist, dass Pixel nicht gleichberechtigt erodieren. Eine Lösung wäre die Synchronisierung an jedem Berührungspunkt. Die Anzahl der Artefakte konnte durch umgedrehte Sortierung der benachbarten Threads reduziert werden. Die horizontalen Streifen werden fortlaufend nummeriert. Gerade Streifen wer-

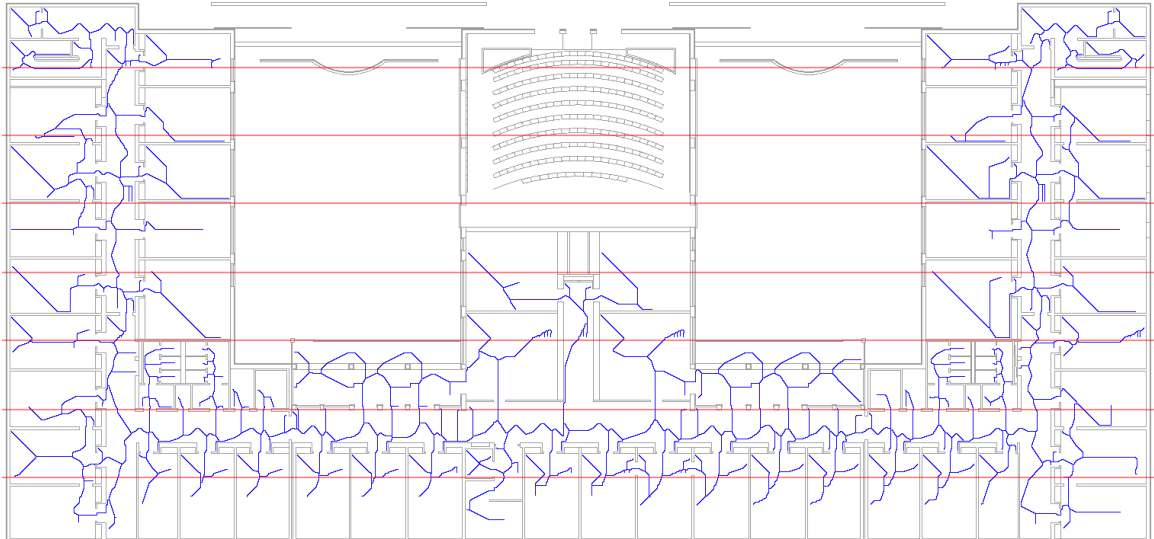


Abbildung 5.9: Ein erodierter Gebäudeplan. Der Routing-Graph (blau) wurde mit acht Threads parallel berechnet. Die Grenzen zwischen den Threads sind in rot angegeben.

den zeilenweise von oben links bis unten rechts abgearbeitet, ungerade entsprechend andersherum. Des Weiteren berühren sich zwar Kanten über Thread-Grenzen hinweg, jedoch resultiert ein signifikant unterschiedliches Ergebnis im Vergleich mit dem seriellen Algorithmus. Da Räume nicht mehr als Ganzes und, auf einzelne Pixel bezogen, insbesondere nicht mehr in der richtigen Reihenfolge bearbeitet werden, bilden Kanten nun nicht mehr zwangsläufig die Mitte eines Raumes ab. Allgemein formuliert ergeben sich Abhängigkeiten zum Ort der Thread-Grenze. Im konkreten Anwendungsfall muss daher separat entschieden werden, ob eine derartige Parallelisierung sinnvoll ist oder nicht.

Die Anzahl möglicher Belegungen der neuen Formel 5.10 ist offensichtlich geringer. Trotz der Entfernung von Klauseln aus der eigentlich bereits reduzierten Formel 5.9 durch Entfernung der Gruppe mit Repräsentant 5.4b, verbessert sich offenbar das Resultat. Bei Betrachtung der Häufigkeit der Versuche in Abb. 5.10, ist zunächst ersichtlich, dass offenbar nur wenige Pixel tatsächlich weniger schnell geschwärzt werden. Auch erkennbar ist allerdings, dass sich die Trefferrate bei Anwendung von nur vier Kernen (Formel 5.11) schon beim ersten Versuch deutlich erhöht, was direkt mit einer schnelleren Erosion korreliert.

Abbildung 5.11 zeigt das Ergebnis bei nur vier benutzten Kernen. Wie zuvor erwähnt,

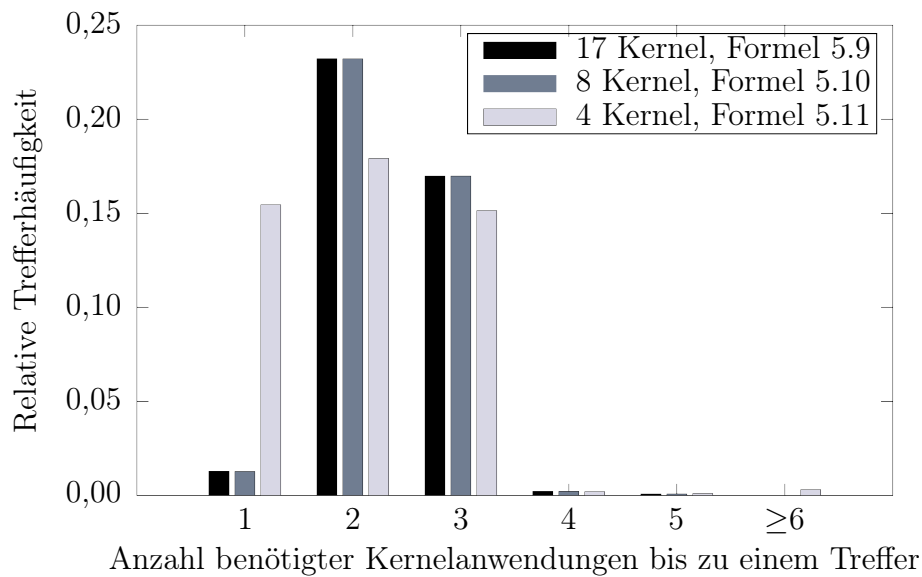


Abbildung 5.10: Durchschnittliche relative Häufigkeit von Schwärzungen beim ersten, zweiten usw. Versuch die Kernel auf einen bestimmten Pixel anzuwenden. Wurde ein Pixel bei der ersten (zweiten, dritten, ...) Überprüfung geschwärzt, war nur eine (zwei, drei, ...) Kernelanwendung nötig.

verletzt dieses Szenario die selbst auferlegten Kriterien für eine Repräsentation üblicher Gehwege im Gebäude. Gleichwohl werden hier verwertbare Informationen über das Gebäude berechnet, wie zum Beispiel das generelle Layout oder begehbare Strukturen. Da das Ergebnis deutlich weniger Kanten besitzt, ist es für eine weitere Verarbeitung, wie bspw. Routing in größeren Gebäuden, Raumzählung oder -verortung, unter Umständen besser geeignet. Auch die benötigte Rechenzeit mit weniger Kernen ist geringer, jedoch bleiben die Verwertungsmöglichkeiten für Positionierung im Sinne dieser Arbeit stark begrenzt, da keine üblichen Gehwege mehr abgebildet werden.

Wie zu vermuten war, besitzt der Algorithmus eine Abhängigkeit bezüglich der Rotation. Eine Änderung der Ausrichtung kann eine große Änderung im Resultat zur Folge haben. Um dies zu verdeutlichen, werden die Resultate zweier Eingaben gezeigt. Das Bild der zweiten Eingabe ist, bis auf eine geänderte Rotation und der damit verbundenen Interpolation der Pixel, identisch mit der ersten Eingabe. Abbildung 5.12 zeigt einen Ausschnitt beider Resultate auf einer Etage. Der rote Graph zeigt das Resultat bei intuitiver Ausrichtung der Eingabekarte, wie sie auch hier im Dokument dargestellt ist. Die Wände sind bei dieser Eingabe somit von oben nach unten bzw. links nach rechts eingetragen. Der blaue Graph zeigt das Resultat bei im Uhrzeigersinn um

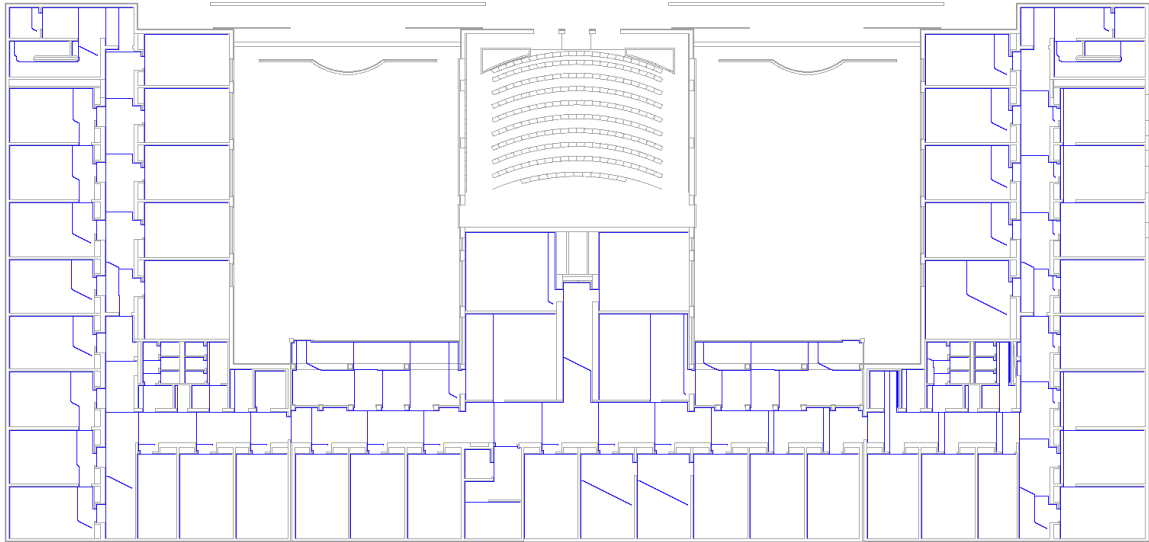


Abbildung 5.11: Ein erodierter Gebäudeplan (blau) bei Anwendung von nur vier Kernen. Wände wachsen hier von rechts unten nach links oben. Die verwendeten Kernel sind in Abb. 5.7 gezeigt.

45° gedrehter Eingabe, zur Präsentation zurückgedreht. Im gedrehten Resultat wird deutlich, dass die Grundstruktur fast identisch zur intuitiven Ausrichtung bleibt. Bei genauerem Betrachten lassen sich in vielen Räumen Ausreißer, in dieser Darstellung nach links unten, finden. Abbildung 5.13 zeigt die zurückgedrehte, rotierte Eingabe im Ganzen. Hier finden sich meist lange Fortsätze bei Räumen, welche auf der linken Seite von Fluren befinden, und meist kurze bei Räumen auf der rechten Seite eines Flurs. Dies liegt an der sequentiellen Abarbeitung der Pixel der Eingabe. Der Grund ist die Sortierung der Tupel in  $WPL$  und  $WPL_{next}$ . Wie zuvor beschrieben wird eine spaltenweise Sortierung von links oben nach rechts unten verwendet. Dieses inhärente Verhalten kann nicht ohne Weiteres bei sequentieller Abarbeitung der Pixel aufgrund einer solchen Sortierung geändert werden. Jedoch widerspricht das Ergebnis nicht dem generellen Verständnis davon, wie ein solcher Graph aussehen könnte oder sollte. Nach vorheriger Definition sind auch hier übliche Gehwege repräsentiert.

Der generierte Graph wird direkt aus der Rastergrafik eines Gebäudeplans extrahiert. Da dieser Schritt automatisiert durchgeführt wird, ist hier keine informierte Person involviert. Im Detail betrachtet könnte eine solche für bestimmte Areale (öffentliche bzw. offene Bereiche, bestimmte Nutzungsarten etc.) bessere Entscheidungen treffen. Weiterhin werden jegliche Metainformationen in der Karte (bspw. Bezeichnungen,

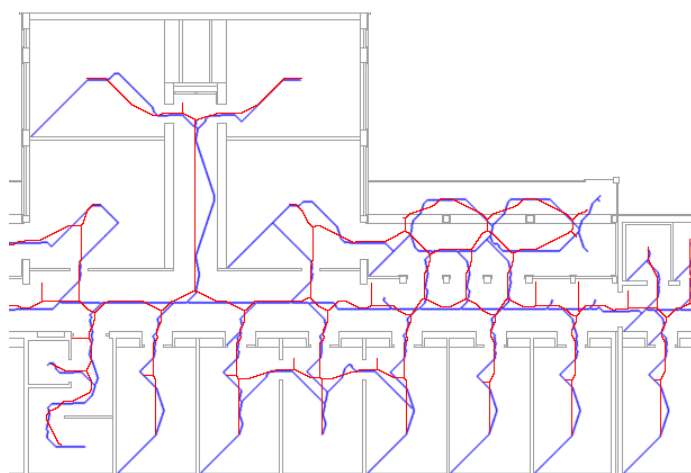


Abbildung 5.12: Erodierter Gebäudeplan. Der rote Graph repräsentiert das Ergebnis ohne rotierte Karte, also in intuitiver Orientierung wie hier abgebildet. Der blaue Graph zeigt das Ergebnis, wenn die selbe Karte um  $45^\circ$  im Uhrzeigersinn gedreht dem Algorithmus als Eingabe übergeben wird. Das Ergebnis wurde für eine sinnhafte Präsentation zurückgedreht.

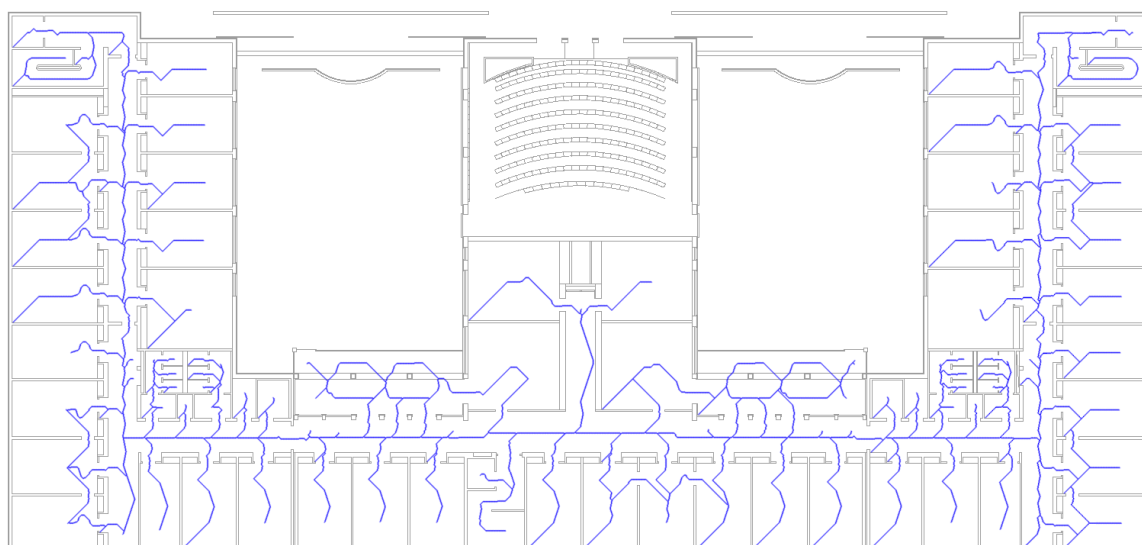


Abbildung 5.13: Erodierter Gebäudeplan. Der Graph zeigt das Ergebnis, wenn die Karte um  $45^\circ$  im Uhrzeigersinn gedreht dem Algorithmus als Eingabe übergeben wird. Das Ergebnis wurde für eine sinnhafte Präsentation zurückgedreht.

Symbole, Türöffnungsrichtungen oder Treppen) außer Acht gelassen und bei ungenügender Vorverarbeitung fälschlicherweise als Hindernis bei der Erosion betrachtet. Da hier auch nur 2D-Repräsentationen betrachtet werden, führen im Allgemeinen keine Kanten in Richtung von oder auf Treppen. In solchen Fällen muss ein manueller Vorverarbeitungsschritt erfolgen.

Ein manuell erstellter Graph würde im Allgemeinen weniger Kanten als ein automatisch erstellter Graph besitzen. Nichtsdestotrotz ist die Struktur in den meisten öffentlichen Bereichen ähnlich. Wenn die Anzahl an Kanten zu hoch ist, kann hier eine Vektorisierung angewandt werden, um Kanten ggf. zu verschmelzen [72]. Da in dieser Arbeit eine Positionsschätzung fokussiert wird, sollten für einen solchen Prozess mehr Kanten günstiger sein.

Dieser Ansatz ist generell auf alle Arten von Gebäuden anwendbar. Allerdings können offene Bereiche nur durch eine Kante ohne weitere Substrukturen dargestellt sein, insbesondere wenn nur ein Eingang in diesen Raum führt. Übliche Gehwege in solchen Bereichen können ohne weiteres Wissen schlicht nicht vorhergesagt werden.

Da der Algorithmus jeden Pixel nur endlich oft prüft, steht das Ergebnis in linearer Zeit in Abhängigkeit zur Anzahl der Pixel in der Eingabe bereit. Eine Parallelisierung ist vielversprechend für höhere Auflösungen und größere Karten.

## 5.5 Zusammenfassung

Diese Arbeit zeigt die automatische Extraktion des Layouts aus Gebäudeplänen mit dem Ziel einer Verbesserung der Positionsschätzung und Navigationsfähigkeit von Lokalisierungssystemen. Der Algorithmus bedient sich aus Ideen der Bildverarbeitung. Es wird eine regelbasierte Erosion weißer Pixel benutzt. Diese Regeln ( $3 \times 3$  Pixel-Kernel) werden auf jeden weißen Pixel iterativ angewandt. Es wird eine Liste an genutzten Kernen präsentiert, welche durch eine Überführung in eine mathematische Lösung weiter reduziert wird. Abhängig von der konkreten Anwendung, reichen vier oder acht Kernel zur Extraktion eines Graphen aus.

Dieser Algorithmus, bzw. dessen Ausgabe, kann ein INS und auch visuelle Systeme unterstützen, um einen systeminhärenten Drift auszugleichen. Auch distanzbasierte Lokalisierungssysteme könnten davon profitieren. Hier könnten weniger Distanzmessungen als üblich nötig sein, um eine raumgenaue Schätzung zu erlauben. Das

Resultat des Algorithmus kann im Prozess der Positionsschätzung und auch der Navigation eingesetzt werden. Um Letztere zu intensivieren, könnten Raumannotationen und Kosten für die Kanten hinzugefügt werden.

Ein nächster Schritt könnte die semi-automatische Annotation von Räumen sein. Räume mit nur einem Eingang werden im Graphen durch Blätter repräsentiert. Durch die Erweiterung des Algorithmus zur Schätzung von Strukturbreiten, könnten in diesem Kontext auch bspw. Raumeingänge annotiert werden. Weiter könnten Flure durch die Erkennung von längeren, gleichmäßigen Strukturbreiten erkannt und annotiert werden.

Es wurde gezeigt, wie eine 2D-Extraktion möglich ist. Zur Verbindung mehrerer Etagen eines Gebäudes bedarf es einer informierten Person, sofern die Genauigkeit der dritten Dimension von Interesse ist. Die 2D-Pixel- bzw. Kantenkoordinaten können leicht um die dritte Dimension erweitert werden. Bei Verbindungen zwischen Etagen können entsprechende Kanten manuell eingefügt werden. Eine Automatisierung wäre mit der Integration weiterer Bildverarbeitung denkbar. Hierzu zählen die Erkennung von, bzw. die Verwertung von Information über, Treppen, Aufzügen, Rampen, Leitern etc. in der Karte. Auch die Reihenfolge der Etagen ist zu beachten. Grundsätzlich denkbar ist ebenfalls eine pixelbasierte Erweiterung des Algorithmus auf die dritte Dimension, falls 3D-Modelle des Gebäudes zur Verfügung stehen.



---

## KAPITEL 6

---

# Zusammenfassung und Ausblick

Diese Dissertation gibt einen Einblick in aktuelle Verfahren zur Lokalisierung von Menschen oder Objekten in Gebäuden und zeigt neue Erkenntnisse über Schwachstellen der Untersuchung von Lokalisierungslösungen auch bei internationalen Konferenzen. In Kapitel 2, 3 und 4 wurden derartige Schwachstellen und Lösungsmöglichkeiten vorgestellt. Kapitel 5 ist inhaltlich abgegrenzt und stellt eine Möglichkeit zum Abgleich von Distanz- oder Inertialdaten auf einem Graphen vor.

**Kapitel 2 (Evaluation von Lokalisierungssystemen)** zeigt die gegenwärtigen Methodiken der Evaluation und deren Entwicklung in der Wissenschaft über die letzten Jahre. Die Dokumentationstiefe und vor allem der Vergleich mit früheren Lösungen lässt vielfach zu wünschen übrig. Mit einem neuartigen RS wurde eine kostengünstige Möglichkeit gezeigt, viele verschiedene Technologien numerisch zu evaluieren. Es wird die Nutzung derartiger Systeme für Feldtests motiviert.

Offen ist die automatische Evaluation in unebenen oder in durch das RS schwer zugänglichen Bereichen, bspw. bei mehreren Stockwerken. Auch wurde die Höhe des SUT nicht verändert; zukünftige Untersuchungen sollten einen Fokus auf die dritte Dimension legen. Eine Spezialisierung auf konkrete Anwendung, bspw. Lagerhallen, U-Bahn Stationen etc., könnte neue Probleme aufzeigen.

**Kapitel 3 (Virtuelles Testbed für distanzbasierte Lokalisierung)** erweitert bisherige Testbeds und Feldtests um eine virtuelle Komponente. Während Feldtests meist einer aufwändigen Installation bedürfen und Testbeds meist statischen oder sehr begrenzt mobilen Charakter haben, erlaubt VirTIL die Nutzung einer breiten Datenbasis mit virtuell de-/aktivierbaren Ankern und der Auswahl beliebiger Pfade in dieser Datenbasis. Das neuartige Testbed erlaubt daher die

Analyse verschiedener Lokalisierungsalgorithmen in verschiedenen (virtuellen) Szenarien mit nur einer Datensammlung und kombiniert daher die Flexibilität einer Simulation mit der Realitätsnähe eines realen Experiments.

VirTIL kann um weitere Lokalisierungstechniken und weitere Gebäude erweitert werden. Es lässt sich vermuten, dass die Evaluation von kooperativen Verfahren zukünftig wichtiger wird. Die Unterstützung hierfür sollte ausgebaut werden.

**Kapitel 4 (Räumliche Fehlerverteilung bei distanzbasierter Lokalisierung)** untersucht Distanz- und Positionsfehler. Es existiert keine Technologie, welche in jedem Szenario einen homogenen Fehler bietet. Weiterhin korrelieren die Fehler an einem Evaluationspunkt bei verschiedenen Lösungen nicht zwangsläufig. Daher ist die Wahl der Punkte kritisch für die Bewertung. Es wurde eine Analyse der Positionierung von Ankern und verschiedene LOS- und NLOS-Effekte in einem typischen Bürogebäude gezeigt. Die Arbeit zeigt zuvor unbekannte physikalische Effekte und deren Auswirkungen.

Es ist nicht auszuschließen, dass bei Betrachtung weiterer Daten (Systeme, Gebäude) weitere unerwartete physikalische Effekte auftreten. Der Vergleich mit Simulationen zeigte signifikante Diskrepanzen. Eine genauere Betrachtung dieser könnte zu Verbesserungen von Simulationen führen.

**Kapitel 5 (Erzeugung von Routing Graphen)** zeigt eine Lösung zum Zurücksetzen des systemimmanenten Drifts bei INS mit aus Gebäudeplänen extrahierten Routing-Graphen. Es wurde ein neuartiger Algorithmus entworfen, welcher derartige Graphen aus gegebenen Gebäudeplänen extrahiert und sich dabei Verfahren aus der Bildverarbeitung zunutze macht.

Derartige Graphen können auch für den Abgleich von Funkdistanzmessungen genutzt werden und bieten daher ein großes Potential, falls nur wenige oder ungenaue Distanzwerte zur Verfügung stehen. Weiterhin sollte die Anwendung auf weitere Probleme untersucht werden, bspw. die Straßenerkennung auf Satellitenbildern oder Messung von Strukturweiten, wie verstopfte Arterien.

Die Lokalisierung innerhalb von Gebäuden wird auch in absehbarer Zukunft genauer werden, wodurch eine gewissenhafte Dokumentation der Evaluation noch an Relevanz gewinnt. Dezimeter- oder gar zentimetergenaue Techniken erfordern eine Anpassung der vorgestellten Evaluation. Mit einer höheren Auflösung der Auswertungen können ggf. auch bei neuen Techniken weitere unvorhergesehene Probleme im Detail verstanden und Lösungen erarbeitet werden.

---

# Literaturverzeichnis

- [1] R. Zekavat und R. M. Buehrer, *Handbook of position location: Theory, practice and advances*, Ser. IEEE Series on Digital & Mobile Communication. John Wiley & Sons, 2011, Bd. 27, ISBN: 9781118104767.
- [2] A. Bensky, *Wireless Positioning Technologies and Applications, Second Edition*: Ser. GNSS technology and applications series. Artech House, 2016, ISBN: 9781608079520.
- [3] R. Mautz, “Indoor positioning technologies”, Habilitation, ETH Zürich, Department of Civil, Environmental, Geomatic Engineering, Institute of Geodesy und Photogrammetry, 2012.
- [4] T. Hillebrandt und H. Will, “Distanzbasierte Indoorlokalisierung in mobilen Netzwerken”, Dissertation, Freie Universität Berlin, 2014.
- [5] A. Lindemann, B. Schnor, J. Sohre und P. Vogel, “Indoor Positioning: A Comparison of WiFi and Bluetooth Low Energy for Region Monitoring.”, in *HEALTHINF*, 2016, S. 314–321.
- [6] R. Bruno und F. Delmastro, “Design and analysis of a bluetooth-based indoor localization system”, in *IFIP International Conference on Personal Wireless Communications*, Springer, 2003, S. 711–725.
- [7] L. M. Ni, Y. Liu, Y. C. Lau und A. P. Patil, “LANDMARC: indoor location sensing using active RFID”, *Wireless networks*, Jg. 10, Nr. 6, S. 701–710, 2004.
- [8] A. Youssef, J. Krumm, E. Miller, G. Cermak und E. Horvitz, “Computing location from ambient FM radio signals [commercial radio station signals]”, in *Wireless Communications and Networking Conference*, IEEE, Bd. 2, 2005, S. 824–829.

- 
- [9] J. Krumm, G. Cermak und E. Horvitz, “Rightspot: A novel sense of location for a smart personal object”, in *International Conference on Ubiquitous Computing*, Springer, 2003, S. 36–43.
  - [10] C. Beder und M. Klepal, “Fingerprinting based localisation revisited: A rigorous approach for comparing RSSI measurements coping with missed access points and differing antenna attenuations”, in *Indoor Positioning and Indoor Navigation (IPIN), International Conference on*, IEEE, 2012.
  - [11] S. Sen, R. R. Choudhury, B. Radunovic und T. Minka, “Precise indoor localization using PHY layer information”, in *Proceedings of the 10th ACM Workshop on hot topics in networks*, ACM, 2011.
  - [12] S. Sen, B. Radunovic, R. R. Choudhury und T. Minka, “You are facing the Mona Lisa: spot localization using PHY layer information”, in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, ACM, 2012, S. 183–196.
  - [13] S. P. Tarzia, P. A. Dinda, R. P. Dick und G. Memik, “Indoor localization without infrastructure using the acoustic background spectrum”, in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, ACM, 2011, S. 155–168.
  - [14] L. Li, P. Hu, C. Peng, G. Shen und F. Zhao, “Epsilon: A Visible Light Based Positioning System”, in *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*, Ser. NSDI’14, Seattle, WA, 2014, S. 331–343, ISBN: 978-1-931971-09-6.
  - [15] T. E. Abrudan, Z. Xiao, A. Markham und N. Trigoni, “Distortion rejecting magneto-inductive three-dimensional localization (MagLoc)”, *IEEE Journal on Selected Areas in Communications*, Jg. 33, Nr. 11, S. 2404–2417, 2015.
  - [16] G. Pirkel und P. Lukowicz, “Robust, low cost indoor positioning using magnetic resonant coupling”, in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, ACM, 2012, S. 431–440.
  - [17] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai und M. Wiseman, “Indoor location sensing using geo-magnetism”, in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, ACM, 2011, S. 141–154.

- [18] C. Zhang, J. Luo und J. Wu, “A dual-sensor enabled indoor localization system with crowdsensing spot survey”, in *Distributed Computing in Sensor Systems (DCOSS), IEEE International Conference on*, IEEE, 2014, S. 75–82.
- [19] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef und R. R. Choudhury, “No need to war-drive: Unsupervised indoor localization”, in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, ACM, 2012, S. 197–210.
- [20] H. Hellmers, Z. Kasmi, A. Norrdine und A. Eichhorn, “Accurate 3D Positioning for a Mobile Platform in Non-Line-of-Sight Scenarios Based on IMU/-Magnetometer Sensor Fusion”, *Sensors*, Jg. 18, 2018, ISSN: 1424-8220. DOI: 10.3390/s18010126.
- [21] A. Matic, A. Popleteev, V. Osmani und O. Mayora-Ibarra, “FM radio for indoor localization with spontaneous recalibration”, *Pervasive and Mobile Computing*, Jg. 6, Nr. 6, S. 642–656, 2010.
- [22] A. Popleteev, “Indoor positioning using FM radio signals”, Dissertation, University of Trento, 2011.
- [23] A. Popleteev, V. Osmani und O. Mayora, “Investigation of indoor localization with ambient FM radio stations”, in *Pervasive Computing and Communications (PerCom), IEEE International Conference on*, IEEE, 2012, S. 171–179.
- [24] S.-H. Fang, J.-C. Chen, H.-R. Huang und T.-N. Lin, “Metropolitan-scale location estimation using fm radio with analysis of measurements”, in *International Wireless Communications and Mobile Computing Conference (IWCMC)*, IEEE, 2008, S. 171–176.
- [25] V. Moghtadaiee, A. G. Dempster und B. Li, “Accuracy indicator for fingerprinting localization systems”, in *Position Location and Navigation Symposium (PLANS), IEEE/ION*, IEEE, 2012, S. 1204–1208.
- [26] V. Moghtadaiee, A. G. Dempster und S. Lim, “Indoor positioning based on fm signals and wi-fi signals”, in *Proceedings of IGNSS Symposium*, 2011.
- [27] —, “Indoor localization using fm radio signals: A fingerprinting approach”, in *Indoor Positioning and Indoor Navigation (IPIN), International Conference on*, IEEE, 2011.

- [28] Y. Chen, D. Lymberopoulos, J. Liu und B. Priyantha, “Fm-based indoor localization”, in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, ACM, 2012, S. 169–182.
- [29] —, “Indoor localization using fm signals”, *IEEE Transactions on Mobile Computing*, Jg. 12, Nr. 8, S. 1502–1517, 2013.
- [30] A. Varshavsky, E. De Lara, J. Hightower, A. LaMarca und V. Otsason, “GSM indoor localization”, *Pervasive and Mobile Computing*, Jg. 3, Nr. 6, S. 698–720, 2007.
- [31] V. Otsason, A. Varshavsky, A. LaMarca und E. De Lara, “Accurate GSM indoor localization”, in *Ubiquitous Computing (UbiComp), International conference on*, Springer, 2005, S. 141–158.
- [32] C.-L. Li, C. Laoudias, G. Larkou, Y.-K. Tsai, D. Zeinalipour-Yazti und C. G. Panayiotou, “Indoor geolocation on multi-sensor smartphones”, in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, ACM, 2013, S. 503–504.
- [33] M. Youssef und A. Agrawala, “The Horus WLAN location determination system”, in *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, ACM, 2005, S. 205–218.
- [34] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach und L. E. Kavradi, “Practical robust localization over large-scale 802.11 wireless networks”, in *Proceedings of the 10th annual international conference on Mobile computing and networking*, ACM, 2004, S. 70–84.
- [35] T. Willemsen, “Fusionsalgorithmus zur autonomen Positionsschätzung im Gebäude, basierend auf MEMS-Inertialsensoren im Smartphone”, Dissertation, HafenCity Universität Hamburg, 2016.
- [36] Y. Sun, “An Indoor Positioning Strategy based on Inertial Sensors”, Dissertation, Freie Universität Berlin, 2016.
- [37] Z. Xiao, H. Wen, A. Markham und N. Trigoni, “Lightweight map matching for indoor localisation using conditional random fields”, in *Proceedings of the 13th international symposium on Information processing in sensor networks*, IEEE Press, 2014, S. 131–142.
- [38] E. Köppe, “Lokalisierung sich bewegender Objekte innerhalb und außerhalb von Gebäuden”, Dissertation, Freie Universität Berlin, 2014.

- [39] M. Edel und E. Köppe, “An advanced method for pedestrian dead reckoning using BLSTM-RNNs”, in *Indoor Positioning and Indoor Navigation (IPIN), International Conference on*, IEEE, 2015.
- [40] Z. Zhao, T. Braun, Z. Li, A. Neto u. a., “A real-time robust indoor tracking system in smartphones”, *Computer communications*, 2017.
- [41] J.-O. Nilsson, J. Rantakokko, P. Handel, I. Skog, M. Ohlsson und K. Hari, “Accurate indoor positioning of firefighters using dual foot-mounted inertial sensors and inter-agent ranging”, in *Position, Location and Navigation Symposium (PLANS), IEEE/ION*, IEEE, 2014, S. 631–636.
- [42] P. Lazik und A. Rowe, “Indoor pseudo-ranging of mobile devices using ultrasonic chirps”, in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, ACM, 2012, S. 99–112.
- [43] N. B. Priyantha, A. Chakraborty und H. Balakrishnan, “The cricket location-support system”, in *Proceedings of the 6th annual international conference on Mobile computing and networking*, ACM, 2000, S. 32–43.
- [44] Y. Yang, “Sample-based Probabilistic Estimation for Indoor Positioning and Tracking Under Ranging Uncertainty”, Dissertation, Freie Universität Berlin, 2015.
- [45] Y. Zhao, “Adaptive Particle Filters for Wireless Indoor Target Tracking”, Dissertation, Freie Universität Berlin, 2014.
- [46] R. Reimann, A. Bestmann und M. Ernst, “Locating technology for aal applications with direction finding and distance measurement by narrow bandwidth phase analysis”, in *International Competition on Evaluating AAL Systems through Competitive Benchmarking*, Springer, 2012, S. 52–62.
- [47] P. Bahl und V. N. Padmanabhan, “RADAR: An in-building RF-based user location and tracking system”, in *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Ieee, Bd. 2, 2000, S. 775–784.
- [48] H. Will, T. Hillebrandt und M. Kyas, “The Geo-n localization algorithm”, in *Indoor Positioning and Indoor Navigation (IPIN), International Conference on*, Nov. 2012. DOI: 10.1109/IPIN.2012.6418867.

- [49] R. Want, A. Hopper, V. Falcao und J. Gibbons, “The active badge location system”, *ACM Transactions on Information Systems (TOIS)*, Jg. 10, Nr. 1, S. 91–102, 1992.
- [50] A. Pogo, “Gemma Frisius, his method of determining differences of longitude by transporting timepieces (1530), and his treatise on triangulation (1533)”, *Isis*, S. 469–506, 1935.
- [51] K. Römer, “The lighthouse location system for smart dust”, in *Proceedings of the 1st international conference on Mobile systems, applications and services*, ACM, 2003, S. 15–30.
- [52] P. Rong und M. L. Sichitiu, “Angle of arrival localization for wireless sensor networks”, in *Sensor and Ad Hoc Communications and Networks (SECON), 3rd Annual IEEE Communications Society on*, IEEE, Bd. 1, 2006, S. 374–382.
- [53] R. B. Ertel und J. H. Reed, “Angle and time of arrival statistics for circular and elliptical scattering models”, *Selected Areas in Communications, IEEE Journal on*, Jg. 17, Nr. 11, S. 1829–1840, 1999.
- [54] Q. H. Spencer, B. D. Jeffs, M. A. Jensen und A. L. Swindlehurst, “Modeling the statistical time and angle of arrival characteristics of an indoor multipath channel”, *Selected Areas in Communications, IEEE Journal on*, Jg. 18, Nr. 3, S. 347–360, 2000.
- [55] P. Kułakowski, J. Vales-Alonso, E. Egea-López, W. Ludwin und J. García-Haro, “Angle-of-arrival localization based on antenna arrays for wireless sensor networks”, *Computers & Electrical Engineering*, Jg. 36, Nr. 6, S. 1181–1186, 2010.
- [56] R. Kuang, H. Song und G. Wang, “Target localization via correlated link inference”, in *Mechatronics and Automation (ICMA), International Conference on*, IEEE, 2011, S. 1010–1014.
- [57] O. Kaltiokallio, M. Bocca und N. Patwari, “Enhancing the accuracy of radio tomographic imaging using channel diversity”, in *the 9th IEEE International Conference on Mobile Ad hoc and Sensor Systems*, 2012.
- [58] Y. Zhao und N. Patwari, “Histogram distance-based radio tomographic localization”, in *Proceedings of the 11th international conference on Information Processing in Sensor Networks*, ACM, 2012, S. 129–130.



- [59] S. Adler, S. Schmitt und M. Kyas, “Device-free indoor localisation using radio tomography imaging in 800/900 MHz band”, in *Indoor Positioning and Indoor Navigation (IPIN), International Conference on*, Okt. 2014, S. 544–553. DOI: 10.1109/IPIN.2014.7275527.
- [60] J. Hightower und G. Borriello, “A Survey and Taxonomy of Location Systems for Ubiquitous Computing”, *IEEE Computer, Techn. Ber.*, 2001.
- [61] H. Liu, H. Darabi, P. Banerjee und J. Liu, “Survey of wireless indoor positioning techniques and systems”, *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Jg. 37, Nr. 6, S. 1067–1080, 2007.
- [62] Y. Gu, A. Lo und I. Niemegeers, “A survey of indoor positioning systems for wireless personal networks”, *Communications Surveys Tutorials, IEEE*, Jg. 11, Nr. 1, S. 13–32, 2009, ISSN: 1553-877X. DOI: 10.1109/SURV.2009.090103.
- [63] D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, V. Handziski, S. Sen, F. Lemic, J. Busch, Z. Jiang, H. Zou, H. Jiang, A. Ashok, C. Xu, A. Ghose, N. Ahmed, C. Zhang, Z. Xiao, H. Wen, T. Schmid, D. Lee, M. Klepal, C. Beder, M. Nikodem, S. Szymczak, P. Hoffmann, L. Selavo, D. Giustiniano, V. Lenders, M. Rea, A. Marcaletti, C. Laoudias, D. Zeinalipour-Yazti, Y.-K. Tsai, T. E. Abrudan, A. Markham, A. Bestmann, R. Reimann, L. Li, C. Zhao, S. Adler, S. Schmitt, P. Lazik, N. Rajagopal, A. Rowe, V. Dentamaro, D. Colucci, P. Ambrosini, A. S. Ferraz, L. Q. L. Martins, P. Bello, A. Alvino, V. Sark, G. Pirkl und P. Hevesi, “A Realistic Evaluation and Comparison of Indoor Location Technologies: Experiences and Lessons Learned”, in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks (IPSN)*, Seattle, Washington: ACM, Apr. 2015, S. 178–189, ISBN: 978-1-4503-3475-4. DOI: 10.1145/2737095.2737726.
- [64] D.-P. Yuan, Z.-h. Li und M. Li, “Research on Indoor Firefighter Positioning Based on Inertial Navigation”, in *Fire Science and Technology*, Springer, 2017, S. 473–483.
- [65] S. Adler, S. Schmitt, K. Wolter und M. Kyas, “A survey of experimental evaluation in indoor localization research”, in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Okt. 2015. DOI: 10.1109/IPIN.2015.7346749.

- [66] S. Schmitt, H. Will, B. Aschenbrenner, T. Hillebrandt und M. Kyas, “A reference system for indoor localization testbeds”, in *Indoor Positioning and Indoor Navigation (IPIN), International Conference on*, Nov. 2012. DOI: 10.1109/IPIN.2012.6418865.
- [67] S. Schmitt, H. Will, T. Hillebrandt und M. Kyas, “A virtual indoor localization testbed for Wireless Sensor Networks”, in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 10th Annual IEEE Communications Society Conference on*, Juni 2013, S. 239–241. DOI: 10.1109/SAHCN.2013.6644985.
- [68] S. Adler, S. Schmitt, H. Will, T. Hillebrandt und M. Kyas, “Virtual testbed for indoor localization”, in *Indoor Positioning and Indoor Navigation (IPIN), International Conference on*, Okt. 2013. DOI: 10.1109/IPIN.2013.6817880.
- [69] S. Adler, S. Schmitt und M. Kyas, “Experimental evaluation of the spatial error distribution of indoor localization algorithms”, in *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS)*, Nov. 2014, S. 44–53. DOI: 10.1109/UPINLBS.2014.7033709.
- [70] —, “Path loss and multipath effects in a real world indoor localization scenario”, in *Positioning, Navigation and Communication (WPNC), 11th Workshop on*, Best Paper Award, März 2014. DOI: 10.1109/WPNC.2014.6843300.
- [71] S. Adler, S. Schmitt, Y. Yang, Y. Zhao und M. Kyas, “Experimental evaluation of indoor localization algorithms”, in *Indoor Positioning and Indoor Navigation (IPIN), International Conference on*, Okt. 2014, S. 291–299. DOI: 10.1109/IPIN.2014.7275495.
- [72] S. Schmitt, L. Zech, T. Willemsen, H. Sternberg und M. Kyas, “Conditional erosion to estimate routing graph out of floor plans”, in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Okt. 2016. DOI: 10.1109/IPIN.2016.7743602.
- [73] S. Schmitt, L. Zech, K. Wolter, T. Willemsen, H. Sternberg und M. Kyas, “Fast Routing Graph Extraction from Floor Plans”, in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sep. 2017.
- [74] *International Conference on Indoor Positioning and Indoor Navigation, IPIN 2010, Zürich, Schweiz, September 15-17*, IEEE, 2010, ISBN: 978-1-4244-5864-6. Adresse: <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=5637226>.

- [75] *International Conference on Indoor Positioning and Indoor Navigation, IPIN 2011, Guimaraes, Portugal, September 21-23*, IEEE, 2011, ISBN: 978-1-4577-1805-2. Adresse: <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6062621>.
- [76] *International Conference on Indoor Positioning and Indoor Navigation, IPIN 2012, Sydney, Australien, November 13-15*, IEEE, 2012, ISBN: 978-1-4673-1955-3. Adresse: <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6409516>.
- [77] *International Conference on Indoor Positioning and Indoor Navigation, IPIN 2013, Montbeliard, Frankreich, Oktober 28-31*, IEEE, 2013, ISBN: 978-1-4799-4043-1. Adresse: <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6811041>.
- [78] *International Conference on Indoor Positioning and Indoor Navigation, IPIN 2014, Busan, Südkorea, Oktober 27-30*, IEEE, 2014, ISBN: 978-1-4673-8054-6. Adresse: <https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7269216>.
- [79] W. F. Tichy, P. Lukowicz, L. Prechelt und E. A. Heinz, “Experimental Evaluation in Computer Science: A quantitative study”, *Journal of Systems and Software*, Jg. 28, Nr. 1, S. 9–18, Jan. 1995. DOI: 10.1016/0164-1212(94)00111-Y.
- [80] W. F. Tichy, “Should Computer Scientists Experiment More?”, *Computer*, Jg. 31, Nr. 5, S. 32–40, Mai 1998, ISSN: 0018-9162. DOI: 10.1109/2.675631.
- [81] J. Wainer, C. Barsottini, D. Lacerda und L. de Marcom, “Experimental evaluation in Computer Science II: a quantitative study, 12 years later”, Institute of Computing, University of Campinas, Techn. Ber. IC-09-02, 2009. Adresse: <http://www.ic.unicamp.br/~reltech/2009/09-02.pdf>.
- [82] J. Wainer, C. G. N. Barsottini, D. Lacerda und L. R. M. de Marco, “Empirical evaluation in Computer Science research published by ACM”, *Information and Software Technology*, Jg. 51, Nr. 6, S. 1081–1085, Juni 2009. DOI: 10.1016/j.infsof.2009.01.002.
- [83] C. Andujar, V. Schiaffonati, F. A. Schreiber, L. Tanca, M. Tedre, K. van Hee und J. van Leeuwen, “The Role and Relevance of Experimentation in Informatics”, *8th European Computer Science Summit of Informatics Europe*, 2013.

- 
- [84] M. Tedre und N. Moisseinen, “Experiments in computing: A survey”, *The Scientific World Journal*, Jg. 2014, 2014.
- [85] K. Popper, *The logic of scientific discovery*. Routledge, 2005.
- [86] J. B. Buckheit und D. L. Donoho, “WaveLab and Reproducible research”, in *Wavelets and Statistics*, Ser. Lecture Notes in Statistics, A. Antoniadis und G. Oppenheim, Hrsg., Bd. 103, New York: Springer, 1995, S. 55–81. DOI: 10.1007/978-1-4612-2544-7\_5.
- [87] C. Collberg, T. Proebsting und A. M. Warren, “Repeatability and Benefaction in Computer Science Research”, University of Arizona, Arizona, Techn. Ber. TR 14-05, Feb. 2015. Adresse: <http://reproducibility.cs.arizona.edu/v2/RepeatabilityTR.pdf>.
- [88] R. Gentleman und D. T. Lang, “Statistical Analyses and Reproducible Research”, *J. Computational and Graphical Statistics*, Jg. 16, Nr. 9, 2007. DOI: 10.1198/106186007X178663.
- [89] R. A. Fisher, *The Design of Experiments*, 8th. Edinburgh: Oliver und Boyd, 1966.
- [90] *ISO 5725-1 : 1994 "Accuracy (trueness and precision) of measurement methods and results — Part 1: General principles and definitions"*, International Organization for Standardization, Geneva, Schweiz, 1994.
- [91] R. Jain, *The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling*. Ser. Wiley professional computing. Wiley, 1991, ISBN: 978-0-471-50336-1.
- [92] C. J. Willmott und K. Matsuura, “Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance”, *Climate Research*, Jg. 30, Nr. 1, 2005. DOI: 10.3354/cr030079.
- [93] S. Schmitt, “Design und Implementierung eines mobilen Referenzsystems für die Indoorlokalisierung unter Verwendung des Robot Operating Systems”, Bachelorarbeit, Freie Universität Berlin, Jan. 2012.
- [94] —, “Aufbau und Evaluation eines virtuellen Testbeds zur Indoor Lokalisierung in drahtlosen Sensornetzwerken”, Masterarbeit, Freie Universität Berlin, Nov. 2013.

- [95] S. Schmitt, S. Adler und M. Kyas, “The effects of human body shadowing in RF-based indoor localization”, in *Indoor Positioning and Indoor Navigation (IPIN), International Conference on*, Okt. 2014, S. 307–313. DOI: 10.1109/IPIN.2014.7275497.
- [96] M. Piras und A. Cina, “Indoor positioning using low cost GPS receivers: Tests and statistical analyses”, in *Indoor Positioning and Indoor Navigation (IPIN), International Conference on*, IEEE, 2010.
- [97] M. Bal, H. Xue, W. Shen und H. Ghenniwa, “A testbed for localization and tracking in wireless sensor networks”, in *Systems, Man and Cybernetics (SMC), IEEE International Conference on*, IEEE, 2009, S. 3581–3586.
- [98] D. Johnson, T. Stack, R. Fish, D. M. Flickinger, L. Stoller, R. Ricci und J. Lepreau, “Mobile Emulab: A Robotic Wireless and Sensor Network Testbed”, in *25th IEEE International Conference on Computer Communications (INFOCOM)*, 2006. DOI: 10.1109/INFOCOM.2006.182.
- [99] A. Prorok, A. Arfire, A. Bahr, J. Farserotu und A. Martinoli, “Indoor Navigation Research with the Khepera III Mobile Robot: An Experimental Baseline with a Case-study on Ultra-wideband Positioning”, in *Proceedings of the IEEE International Conference on Indoor Positioning and Indoor Navigation*, ETH Zürich, 2010. DOI: 10.1109/IPIN.2010.5647880.
- [100] J. Rydell und E. Emilsson, “(Positioning evaluation)<sup>2</sup>”, in *Indoor Positioning and Indoor Navigation (IPIN), International Conference on*, IEEE, 2011.
- [101] M. Segura, H. Hashemi, C. Sisterna und V. Mut, “Experimental demonstration of self-localized Ultra Wideband indoor mobile robot navigation system”, in *Indoor Positioning and Indoor Navigation (IPIN), International Conference on*, Sep. 2010. DOI: 10.1109/IPIN.2010.5647457.
- [102] F. Lemic, J. Büsch, M. Chwalisz, V. Handziski und A. Wolisz, “Infrastructure for benchmarking rf-based indoor localization under controlled interference”, in *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPIN-LBS)*, IEEE, 2014, S. 26–35.
- [103] T. V. Haute, E. D. Poorter, J. Rossey, I. Moerman, V. Handziski, A. Behboodi, F. Lemic, A. Wolisz, N. Wirström, T. Voigt, P. Combez, P. Verhoeve und J. J. de las Heras, “The EVARILOS Benchmarking Handbook: Evaluation of RF-based Indoor Localization Solutions”, in *2nd International Workshop*

- on Measurement-based Experimental Research, Methodology and Tools*, 2013.  
Adresse: <http://soda.swedish-ict.se/5536/>.
- [104] Willow Garage. (Mai 2018). TurtleBot, Adresse: <http://www.willowgarage.com/turtlebot>.
- [105] K. Khoshelham, “Accuracy analysis of kinect depth data”, in *ISPRS workshop laser scanning*, Bd. 38, 2011.
- [106] (Mai 2018). Robot Operating System, Adresse: <http://www.ros.org/>.
- [107] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler und A. Y. Ng, “ROS: an open-source Robot Operating System”, in *ICRA Workshop on Open Source Software*, 2009.
- [108] D. Fox, “KLD-Sampling: Adaptive Particle Filters and Mobile Robot Localization”, in *In Advances in Neural Information Processing Systems (NIPS)*, 2001.
- [109] I. Guvenc, C.-C. Chong und F. Watanabe, “Analysis of a Linear Least-Squares Localization Technique in LOS and NLOS Environments”, in *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, 2007, S. 1886–1890. DOI: 10.1109/VETECS.2007.391.
- [110] S. Venkatesh und R. Buehrer, “A linear programming approach to NLOS error mitigation in sensor networks”, in *5th International Conference on Information Processing in Sensor Networks (IPSN)*, ACM, 2006, S. 301–308.
- [111] A. Savvides, H. Park und M. B. Srivastava, “The bits and flops of the n-hop multilateration primitive for node localization problems”, in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, Ser. WSNA '02, Atlanta, Georgia, USA: ACM, 2002, S. 112–121, ISBN: 1-58113-589-0. DOI: 10.1145/570738.570755.
- [112] K. Langendoen und N. Reijers, “Distributed localization in wireless sensor networks: a quantitative comparison”, *Comput. Netw.*, Jg. 43, Nr. 4, S. 499–518, Nov. 2003, ISSN: 1389-1286. DOI: 10.1016/S1389-1286(03)00356-6.
- [113] T. Kumpuniemi, M. Hämäläinen, K. Y. Yazdandoost und J. Inatti, “Human body shadowing effect on dynamic UWB on-body radio channels”, *IEEE Antennas and Wireless Propagation Letters*, Jg. 16, S. 1871–1874, 2017.

- [114] S. Adler, S. Schmitt, Y. Yang, Y. Zhao und M. Kyas, “FUBLoc - Accurate Range-based Indoor Localization and Tracking”, Microsoft Indoor Localization Competition Award, 13th ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN), 3. Platz (Infrastructure-based Approaches), Apr. 2014.
- [115] F. Lemic, V. Handziski, A. Wolisz, T. Constancebeys, C. Laoudias, D. Zeinalipour-Yazti, C. Panayiotou, S. Adler, S. Schmitt, H. Will, T. Hillebrandt, M. Kyas, Y. Yang und Y. Zhao, “Systematic Objective Evaluation of RF-based Indoor Localization Algorithms (EVARILOS Open Challenge: Track 3)”, Telecommunication Networks Group, Technische Universität Berlin, TKN Technical Report Series TKN-14-005, Okt. 2014. Adresse: [http://www.tkn.tu-berlin.de/fileadmin/fg112/Papers/2014/lemic14\\_open\\_challenge.pdf](http://www.tkn.tu-berlin.de/fileadmin/fg112/Papers/2014/lemic14_open_challenge.pdf).
- [116] S. Adler, S. Schmitt, H. Will, T. Hillebrandt und M. Kyas, “Geo-n Localization and 3CoM (3 Centers of Mass) Indoor Localization Algorithms”, Evaluation of RF-based Indoor Localization Algorithms, EVARILOS Open Challenge, 2. Platz, Mai 2014.
- [117] F. Lemic, V. Handziski, A. Wolisz, T. Constancebeys, C. Laoudias, S. Adler, S. Schmitt und Y. Yang, “Experimental evaluation of RF-based indoor localization algorithms under RF interference”, in *Localization and GNSS (ICL-GNSS)*, *International Conference on*, Juni 2015. DOI: 10.1109/ICL-GNSS.2015.7217149.
- [118] H. Will, S. Pfeiffer, S. Adler, T. Hillebrandt und J. Schiller, “Distance Measurement in Wireless Sensor Networks with Low Cost Components”, in *Indoor Positioning and Indoor Navigation (IPIN 2011)*, 2011.
- [119] Fraunhofer-Institut für Integrierte Schaltungen IIS. (Mai 2018). Test- und Anwendungszentrum L.I.N.K., Adresse: <https://www.iis.fraunhofer.de/de/profil/standorte/linkhalle.html>.
- [120] F. Lemić, “Enhanced Location Information and its Application for Improving Wireless Networks”, Dissertation, Technische Universität Berlin, 2017.
- [121] F. Lemic, V. Handziski, N. Wirström, T. V. Haute, E. D. Poorter, T. Voigt und A. Wolisz, “Web-based platform for evaluation of RF-based indoor localization algorithms”, in *IEEE International Conference on Communication Workshop (ICCW)*, Juni 2015, S. 834–840. DOI: 10.1109/ICCW.2015.7247280.

- [122] F. Lemic, V. Handziski, N. Wirström, T. Van Haute, E. De Poorter, T. Voigt und A. Wolisz, “Demo abstract: virtual experimental evaluation of RF-based indoor localization algorithms”, in *12th European Conference on Wireless sensor networks (EWSN)*, 2015.
- [123] N. Moayeri, M. O. Ergin, F. Lemic, V. Handziski und A. Wolisz, “PerfLoc (Part 1): An extensive data repository for development of smartphone indoor localization apps”, in *Personal, Indoor, and Mobile Radio Communications (PIMRC), IEEE 27th Annual International Symposium on*, IEEE, 2016.
- [124] M. Angermann, P. Robertson, T. Kemptner und M. Khider, “A high precision reference data set for pedestrian navigation using foot-mounted inertial sensors”, in *Indoor Positioning and Indoor Navigation (IPIN), International Conference on*, IEEE, 2010.
- [125] H. Choset, E. Acar, A. A. Rizzi und J. Luntz, “Exact cellular decompositions in terms of critical points of morse functions”, in *Robotics and Automation, ICRA, IEEE International Conference on*, IEEE, Bd. 3, 2000, S. 2270–2277.
- [126] S. C. Wong und B. A. MacDonald, “A topological coverage algorithm for mobile robots”, in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, IEEE, Bd. 2, 2003, S. 1685–1690.
- [127] E. Gonzalez, O. Alvarez, Y. Diaz, C. Parra und C. Bustacara, “BSA: a complete coverage algorithm”, in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2005, S. 2040–2044.
- [128] Y.-H. Choi, T.-K. Lee, S.-H. Baek und S.-Y. Oh, “Online complete coverage path planning for mobile robots based on linked spiral paths using constrained inverse distance transform”, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2009, S. 5788–5793.
- [129] T.-K. Lee, S.-H. Baek, Y.-H. Choi und S.-Y. Oh, “Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation”, *Robotics and Autonomous Systems*, Jg. 59, Nr. 10, S. 801–812, 2011.
- [130] K. Thiayagarajan und C. G. Balaji, “Traversal algorithm for complete coverage”, *Journal of Computer Science*, Jg. 8, Nr. 12, 2012.
- [131] D. E. Knuth, “A generalization of Dijkstra’s algorithm”, *Information Processing Letters*, Jg. 6, Nr. 1, 1977.



- [132] L. Zech, “Entwicklung eines Pfadplanungsalgorithmus zur vollständigen Abdeckung mit mobilen Robotern”, Bachelorarbeit, Freie Universität Berlin, Mai 2016.
- [133] M. Baar, H. Will, B. Blywis, T. Hillebrandt, A. Liers, G. Wittenburg und J. Schiller, “The ScatterWeb MSB-A2 Platform for Wireless Sensor Networks”, Freie Universität Berlin, Department of Mathematics, Computer Science, Institute for Computer Science, Telematics und Computer Systems group, Takustraße 9, 14195 Berlin, Germany, Techn. Ber. TR-B-08-15, Sep. 2008. Adresse: <ftp://ftp.inf.fu-berlin.de/pub/reports/tr-b-08-15.pdf>.
- [134] H. Will, T. Hillebrandt und M. Kyas, “Wireless sensor networks in emergency scenarios: the FeuerWhere deployment”, in *Proceedings of the 1st ACM international workshop on Sensor-Enhanced Safety and Security in Public Spaces*, Ser. SESP '12, Hilton Head, South Carolina, USA: ACM, 2012, S. 9–14, ISBN: 978-1-4503-1288-2. DOI: 10.1145/2248356.2248360.
- [135] Nanotron Technologies GmbH. (Mai 2018). nanoPAN 5375 RF Module, Adresse: [https://nanotron.com/EN/PR\\_ic\\_modules.php/](https://nanotron.com/EN/PR_ic_modules.php/).
- [136] S. Simic und S. Sastry, “Distributed localization in wireless ad hoc networks”, UC Berkley, Techn. Ber. UCB/ERL M02/26, 2001.
- [137] T. Hillebrandt, H. Will und M. Kyas, “Quantitative and Spatial Evaluation of distance-based Localization Algorithms”, in *Progress in Location-Based Services*, Springer, 2013, S. 173–194.
- [138] A. Bahillo, S. Mazuelas, R. M. Lorenzo, P. Fernández, J. Prieto, R. J. Durán und E. J. Abril, “Hybrid RSS-RTT localization scheme for indoor wireless networks”, *EURASIP J. Adv. Signal Process*, Feb. 2010, ISSN: 1110-8657. DOI: 10.1155/2010/126082.
- [139] D. Liu, P. Ning und W. K. Du, “Attack-resistant location estimation in sensor networks”, in *Proceedings of the 4th international symposium on Information processing in sensor networks (IPSN)*, Los Angeles, California: IEEE Press, 2005, ISBN: 0-7803-9202-7. Adresse: <http://dl.acm.org/citation.cfm?id=1147685.1147704>.
- [140] G. Zanca, F. Zorzi, A. Zanella und M. Zorzi, “Experimental comparison of RSSI-based localization algorithms for indoor wireless sensor networks”, in *Proceedings of the workshop on Real-world wireless sensor networks (RE-*

- ALWSN*), Glasgow, Scotland: ACM, 2008, ISBN: 978-1-60558-123-1. DOI: 10.1145/1435473.1435475.
- [141] N. Bulusu, J. Heidemann und D. Estrin, “GPS-less low-cost outdoor localization for very small devices”, *IEEE Personal Communications*, Jg. 7, Nr. 5, S. 28–34, Okt. 2000, ISSN: 10709916. DOI: 10.1109/98.878533.
- [142] K. Whitehouse, C. Karlof, A. Woo, F. Jiang und D. Culler, “The effects of ranging noise on multihop localization: an empirical study”, in *Proc. 4th international symposium on Information processing in sensor networks*, IEEE Press, 2005, S. 10.
- [143] R. Crepaldi, P. Casari, A. Zanella und M. Zorzi, “Testbed implementation and refinement of a range-based localization algorithm for wireless sensor networks”, in *Mobility '06: Proceedings of the 3rd international conference on Mobile technology, applications & systems*, ACM, New York, NY, USA: ACM, 2006, ISBN: 1-59593-519-3. DOI: <http://doi.acm.org/10.1145/1292331.1292401>.
- [144] J. Wendeberg, J. Muller, C. Schindelbauer und W. Burgard, “Robust tracking of a mobile beacon using time differences of arrival with simultaneous calibration of receiver positions”, in *Indoor Positioning and Indoor Navigation (IPIN), International Conference on*, 2012. DOI: 10.1109/IPIN.2012.6418919.
- [145] N. B. Priyantha, A. Chakraborty und H. Balakrishnan, “The Cricket location-support system”, in *Proceedings of the 6th annual international conference on Mobile computing and networking (MobiCom)*, Boston, Massachusetts, USA: ACM, 2000, S. 32–43, ISBN: 1-58113-197-6. DOI: 10.1145/345910.345917.
- [146] D. Moore, J. Leonard, D. Rus und S. Teller, “Robust distributed network localization with noisy range measurements”, in *Proc. 2nd international conference on Embedded networked sensor systems*, ACM, 2004, S. 50–61.
- [147] N. Amiot, M. Laaraiedh und B. Uguen, “Improved mobility modeling for indoor localization applications”, in *Positioning, Navigation and Communication (WPNC), 11th Workshop on*, März 2014. DOI: 10.1109/WPNC.2014.6843298.
- [148] “Effects of building materials and structures on radiowave propagation above about 100 MHz”, International Telecommunication Union, Techn. Ber. ITU-R P.2040-1, 2015. Adresse: <https://www.itu.int/rec/R-REC-P.2040-1-201507-I/en>.

- [149] “Propagation data and prediction methods for the planning of indoor radio-communication systems and radio local area networks in the frequency range 300 MHz to 100 GHz”, International Telecommunication Union, Techn. Ber. ITU-R P.1238-9, 2017. Adresse: <https://www.itu.int/rec/R-REC-P.1238/en>.
- [150] J. He, K. Pahlavan, S. Li und Q. Wang, “A Testbed for Evaluation of the Effects of Multipath on Performance of TOA-based Indoor Geolocation”, *Instrumentation and Measurement, IEEE Transactions on*, Jg. PP, Nr. 99, 2013, ISSN: 0018-9456. DOI: 10.1109/TIM.2013.2255976.
- [151] P. Teunissen, “The least-squares ambiguity decorrelation adjustment: a method for fast GPS integer ambiguity estimation”, English, *Journal of Geodesy*, Jg. 70, Nr. 1-2, S. 65–82, 1995, ISSN: 0949-7714. DOI: 10.1007/BF00863419.
- [152] G. S. Kuruoglu, M. Erol und S. Oktug, “Localization in Wireless Sensor Networks with Range Measurement Errors”, *Advanced International Conference on Telecommunications*, Jg. 0, S. 261–266, 2009. DOI: <http://doi.ieeecomputersociety.org/10.1109/AICT.2009.51>.
- [153] L. Haiyong, L. Hui, Z. Fang und P. Jinghua, “An Iterative Clustering-Based Localization Algorithm for Wireless Sensor Networks”, *China Communications*, Jg. 8, Nr. 1, 58, S. 58–64, 2011.
- [154] H. Will, T. Hillebrandt und M. Kyas, “The FU Berlin Parallel Lateration-Algorithm Simulation and Visualization Engine”, in *Proc. 9th Workshop on Positioning, Navigation and Communication (WPNC)*, 2012.
- [155] M. Kyas, “High-precision position estimation in indoor environments”, in *International Conference on Advanced Computer Science and Information Systems (ICACISIS)*, Sep. 2013, S. 9–14. DOI: 10.1109/ICACISIS.2013.6761545.
- [156] K. Whitehouse, C. Karlof und D. Culler, “A practical evaluation of radio signal strength for ranging-based localization”, *ACM SIGMOBILE Mobile Computing and Communications Review*, Jg. 11, Nr. 1, S. 41–52, 2007.
- [157] S. Venkatraman, J. Caffery Jr und H. You, “A novel ToA location algorithm using LoS range estimation for NLoS environments”, *IEEE Trans. Vehicular Technology*, Jg. 53, Nr. 5, S. 1515–1524, 2004.

- [158] L. Lazos und R. Poovendran, “SeRLoc: Robust localization for wireless sensor networks”, *ACM Transactions on Sensor Networks (TOSN)*, Jg. 1, Nr. 1, S. 73–100, 2005.
- [159] A. Srinivasan und J. Wu, “A survey on secure localization in wireless sensor networks”, *Encyclopedia of wireless and mobile communications*, 2007.
- [160] J. E. Dennis Jr. und R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations (Classics in Applied Mathematics, 16)*. Soc for Industrial & Applied Math, 1996, ISBN: 0898713641.
- [161] J. J. Robles, J. S. Pola und R. Lehnert, “Extended Min-Max algorithm for position estimation in sensor networks.”, in *9th Workshop on Positioning, Navigation and Communication (WPNC)*, IEEE, 2012, S. 47–52, ISBN: 978-1-4673-1437-4. DOI: 10.1109/WPNC.2012.6268737.
- [162] H. Will, T. Hillebrandt, Y. Yuan, Z. Yubin und M. Kyas, “The Membership Degree Min-Max localization algorithm”, in *2nd International Conference and Exhibition on Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS)*, 2012. DOI: 10.1109/UPINLBS.2012.6409781.
- [163] J. Blankenbach, H. Sternberg und S. Tilch, “Indoor-Positionierung”, in *Handbuch der Geodäsie: 5 Bände*, W. Freeden und R. Rummel, Hrsg. Springer, 2016, S. 1–36, ISBN: 978-3-662-46900-2. DOI: 10.1007/978-3-662-46900-2\_24-1.
- [164] M. Peter, D. Fritsch, B. Schäfer, A. Kleusberg, J. Á. B. Link und K. Wehrle, “Versatile geo-referenced maps for indoor navigation of pedestrians”, in *Indoor Positioning and Indoor Navigation (IPIN)*, *International Conference on*, Bd. 13, 2012.
- [165] H. Sternberg, F. Keller und T. Willemsen, “Precise Indoor Mapping as a Basis for Coarse Indoor Navigation”, *Journal of Applied Geodesy*, Jg. 7, Nr. 4, S. 231–246, Nov. 2013. DOI: 10.1515/jag-2013-0015.
- [166] T. Willemsen, F. Keller und H. Sternberg, “A topological approach with MEMS in smartphones based on routing-graph”, in *Indoor Positioning and Indoor Navigation (IPIN)*, *International Conference on*, IEEE, 2015. DOI: 10.1109/IPIN.2015.7346952.

- [167] K. Halldórsson und Y. Björnsson, “Automated Decomposition of Game Maps”, *Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, Jg. 15, S. 122–127, 2015. Adresse: <https://www.aaai.org/ocs/index.php/AIIDE/AIIDE15/paper/view/11522>.
- [168] A. Uriarte und S. Ontañón, “Improving Terrain Analysis and Applications to RTS Game AI”, in *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2016.
- [169] A. Sironi, E. Türetken, V. Lepetit und P. Fua, “Multiscale Centerline Detection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Jg. 38, Nr. 7, S. 1327–1341, Juli 2016, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2015.2462363.
- [170] L. He, X. Ren, Q. Gao, X. Zhao, B. Yao und Y. Chao, “The connected-component labeling problem: A review of state-of-the-art algorithms”, *Pattern Recognition*, Jg. 70, S. 25–43, 2017.
- [171] T. Ebert, J. Belz und O. Nelles, “Interpolation and extrapolation: Comparison of definitions and survey of algorithms for convex and concave hulls”, in *IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, Dez. 2014, S. 310–314. DOI: 10.1109/CIDM.2014.7008683.
- [172] G. M. Amdahl, “Validity of the single processor approach to achieving large scale computing capabilities”, in *Spring Joint Computer Conference (SJCC)*, ACM, 1967, S. 483–485. DOI: 10.1145/1465482.1465560.

---

# Zusammenfassung

Die Lokalisierung von Menschen oder Objekten ist heute besonders im Freien durch Navigationssatellitensysteme allgegenwärtig. Innerhalb von Gebäuden ergeben sich durch eine deutlich komplexere Signalausbreitung bei funkbasierten Systemen weitaus mehr Probleme als außerhalb. Zudem stellt sich der Evaluationsaufwand im Allgemeinen als äußerst aufwändig heraus. In dieser Dissertation wird daher der Aspekt der Evaluation derartiger Systeme betrachtet. Von besonderer Bedeutung ist dabei die räumliche Analyse von Distanz- und Positionsfehlern auf breiter Fläche. Dazu wird ein mobiles Referenzsystem und ein sog. virtuelles Testbed zur Durchführung von virtuellen Experimenten mit zuvor gesammelten Daten eingeführt. Des Weiteren wird die Extraktion von Graphen aus Gebäudeplänen als weitere Informationsquelle für Lokalisierungssysteme vorgestellt.



---

# Abstract

Today, localization of humans or objects is very ubiquitous, especially in open space due to navigation satellite systems. Indoors we find a more complex signal propagation with radio-frequency based systems, and thus more problems than outdoors. Moreover, the effort for evaluation is utmost expensive. This dissertation therefore addresses the problem of evaluating such systems. Of particular importance is the broad spatial analysis of distance and position errors. A mobile reference system and a so-called virtual testbed for implementing virtual experiments with previously gathered data is introduced. Furthermore, the extraction of graphs from floor plans as a further source of information for localization systems is presented.





*Thus I steer my bark, and sail  
On even keel, with gentle gale.*

– Matthew Green (1696–1737)