



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Johannes Meyer

Ortungsverbesserung durch hybride Verfahren

*Fakultät Technik und Informatik
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science
Department of Computer Science*

Johannes Meyer

Ortungsverbesserung durch hybride Verfahren

Masterarbeit eingereicht im Rahmen der Masterprüfung

im Studiengang Master of Science Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Birgit Wendholt
Zweitgutachter: Prof. Dr. Gunter Klemke

Eingereicht am: 2. Mai 2013

Johannes Meyer

Thema der Arbeit

Ortungsverbesserung durch hybride Verfahren

Stichworte

Ortungsverbesserung, Indoor-Lokalisierung, Pfadkorrektur, Trajektorienbewertung, Kartenabgleich, Ubisense, Gebäudemodell, IFC

Kurzzusammenfassung

Diese Masterarbeit beschäftigt sich mit der Präzisionssteigerung eines absoluten Ortungssystems zur Laufzeit auf Basis eines Bewegungs- und Gebäudemodells. Hierbei wird die aktuelle Position entsprechend der historischen Entwicklung der zurückgelegten Trajektorie in den Grundriss eingepasst (Map-Matching), um einem plausiblen Raumwechsel zu entsprechen. Anhand von einschränkenden Elementen, z. B. Wänden und Türen, wird die korrigierte Position zu dem wahrscheinlichsten und zugleich logisch erreichbaren Punkt verschoben. Zur Lösung einer mehrdeutigen Pfadentwicklung werden zwei selbst entwickelte Algorithmen vorgestellt und deren Korrekturverhalten verglichen.

Johannes Meyer

Title of the paper

Location improvements by hybrid methods

Keywords

Localization improvement, indoor localization, path correction, trajectory evaluation, map matching, Ubisense, building model, IFC

Abstract

This masterthesis tries to increase the precision of an absolute positioning system at runtime, based on a movement- and building model. The current position is fitted in the floor plan accordingly to the historical development of the traveled trajectory (map matching). The corrected position is shifted to the most likely and logical point, based on limiting elements like walls and doors. Two self-developed algorithms are presented to solve an ambiguous path development.

Inhaltsverzeichnis

1. Einführung	1
1.1. Motivation	1
1.2. Ziele	3
1.3. Gliederung	4
2. Vergleichbare Arbeiten	6
2.1. Ortungsverbesserung auf Grundlage von rein messwertbasierten Algorithmen	8
2.1.1. Sensorbereinigung durch Glättung und Plausibilitätsprüfung	8
2.1.2. Drift Compensation	12
2.1.3. Automatisierte RSSI-Berechnung auf Basis von Kartendaten	16
2.2. Map-Matching Verfahren	18
2.2.1. Historische Entwicklung	18
2.2.2. Matching Verfahren auf Basis von Knoten- und Kantenmodellen	19
2.2.3. Relative Ortungsverfahren	20
2.2.4. Absolute Ortungsverfahren	37
2.3. Fazit	45
3. Analyse	47
3.1. Szenarien	47
3.1.1. Ortung und Navigation	48
3.1.2. Activity Pattern Recognition	48
3.2. Anforderungsanalyse	49
3.2.1. Randbedingungen	50
3.2.2. Funktionale Anforderungen	51
3.2.3. Nicht-funktionale Anforderungen	56
3.3. Fazit	59
4. Design	61
4.1. Idee	61
4.2. Algorithmen	63
4.2.1. Gemeinsamkeiten	64
4.2.2. Spezialfälle	69
4.2.3. Berechnung der Wahrscheinlichkeiten	71
4.2.4. Restriktiver Ansatz	73
4.2.5. Fehlertoleranter Ansatz	76

4.3.	Architektur und Integration	78
4.3.1.	Living Place	79
4.3.2.	ISIS	79
4.3.3.	Komponenten und Schnittstellen	81
4.3.4.	Gebäudemodellierung	83
4.3.5.	Visualisierung	84
4.4.	Fazit	85
5.	Evaluation	87
5.1.	Testaufbau und Szenarien	87
5.2.	Messergebnisse	90
5.2.1.	Szenario 1	90
5.2.2.	Szenario 2	96
5.2.3.	Szenario 3	104
5.3.	Vergleichende Betrachtung der Korrekturergebnisse	108
5.4.	Fazit	110
6.	Schluss	112
6.1.	Zusammenfassung	112
6.2.	Bewertung	113
6.3.	Ausblick	114
A.	Anhang	117

1. Einführung

1.1. Motivation

Seit im Jahr 2000 die „selective availability“¹ für GPS-Empfänger (Global Positioning System) aufgehoben wurde, sind Positionsbestimmungen im Bereich von 15 m und kleiner möglich [DoD \(2008\)](#). Zuvor war eine Ortung auf 100 m begrenzt und lediglich für ungefähre Positionsschätzungen nutzbar, welche z. B. in der Nautik Anwendung fand. Ab diesem Zeitpunkt ist das GPS auch für viele andere Bereiche interessant geworden. So bildet die Autonavigation ein zentrales Standbein. Deren Nutzen sorgte schon sehr früh für eine breite Akzeptanz unter der Bevölkerung und ist für die vielfältigen Entwicklungen in diesem Bereich verantwortlich. In der heutigen Zeit ist die GPS-gestützte Ortung und Navigation selbstverständlich geworden und kaum noch wegzudenken. Sei es privat im alltäglichen Umgang mit dem Web 2.0 und Location Based Services oder im kommerziellen Sektor wie beispielsweise der Logistik. GPS bietet eine „out of the box“ Ortung, welche keiner Installation oder Kalibrierung bedarf und eine ausreichende Präzision für vielseitige Anwendungen im Outdoor-Bereich liefert. Die Anwendungsszenarien für eine präzise Ortung sind nicht nur auf den Außenbereich beschränkt und hören an einem Gebäudeeingang auf, denn gerade hier wird es für die Fußgängernavigation interessant. Begibt man sich jedoch in den Indoor-Bereich, stößt das bewährte GPS schnell an seine Grenzen. Aufgrund der Funktionsweise ist idealerweise eine freie Sicht zum Himmel, bzw. zu den Satelliten notwendig. Bauwerke führen zu einer Signalabschwächung, Interferenzen und Reflexionen, welche die Ortungspräzision negativ beeinflussen und unter Umständen zu einem Totalausfall der Ortung führen können [Koyuncu und Yang \(2010\)](#).

Da das Interesse an einer Indoorlokalisierung laut [Liu u. a. \(2007\)](#) stetig zunimmt, werden alternative Ortungssysteme entwickelt um diese Gebiete abzudecken. Die Problematik besteht hierbei in der globalen Verfügbarkeit für ein „Out of the Box“ System und der geforderten Präzision. Hochpräzise Ortungssysteme existieren bereits und basieren beispielsweise auf UWB² oder RFID³ [Mautz \(2011\)](#). Diese finden beispielsweise Anwendung in der Medizin,

¹künstliche Verschlechterung für zivile Nutzer

²Ultra Wideband

³Radio-frequency identification

Industrie oder Logistik. An der HAW wurde schon 2006 ein genaues Indoor-Ortungssystem (IMAPS) auf Basis von Ultraschall entwickelt [Gregor \(2006\)](#). Alle diese Systeme benötigen jedoch zusätzliche Infrastruktur, sind aufwändig in der Installation und meist sehr teuer [Koyuncu und Yang \(2010\)](#). Ein flächendeckender Einsatz für den öffentlichen Nutzen ist demnach in absehbarer Zukunft nicht zu erwarten.

Die Indoor-Ortung ist kein Selbstzweck. Ein genaues Wissen über die aktuelle Indoor-Position ermöglicht neue Entwicklungen im Bereich der Ambient Intelligence. So stattet Google seine Handys mit NFC⁴-Modulen aus, welche eine berührungslose Interaktion ermöglichen. Dies ist nicht nur auf das bargeldlose Bezahlen beschränkt, womit es meistens in Verbindung gebracht wird, sondern findet auch Anwendung im Bereich der intelligenten Wohnung [Fischer \(2009\)](#). Sei es für kontextabhängige Informationen oder die *Augmented Reality*, bei der über das Auslesen von getaggten Objekten der aktuelle Standort in Erfahrung gebracht wird. Diese Ortungen sind jedoch nur punktuell und ermöglichen keine kontinuierliche Lokalisierung, da die Tags lediglich im geringen Abstand erkennbar sind. Somit ist die *Location Awareness* nicht flächendeckend verfügbar und eine Navigation nur eingeschränkt möglich.

Es gilt eine Abwägung zwischen Präzision, Installationsaufwand, Ausfallsicherheit, Skalierbarkeit und Kosten zu treffen. Für großflächige verfügbare Ortungsdienste bietet es sich an, bestehende Infrastruktur zu verwenden. Hierfür kommen nicht viele Technologien in Frage und das Wireless Local Area Network (WLAN) spielt nach Ansicht mehrerer Studien ([Lin und Lin \(2005\)](#), [Koyuncu und Yang \(2010\)](#) und [Liu u. a. \(2007\)](#)) offensichtlich eine sehr wichtige Rolle in der Zukunft. WLAN-Hotspots sind inzwischen weit verbreitet und ermöglichen z. B. eine Lokalisierung mittels Signalstärkenauswertung. Hierbei werden verständlicherweise die Genauigkeiten der hochpräzisen Ortungssysteme nicht erreicht, da ihr ursprünglicher Verwendungszweck ein anderer ist. Die Leistungsfähigkeit wird oft überschätzt und die Präzisionsangaben von bis zu 1m meist nur unter Laborbedingungen erreicht.

Anhand eigener Versuche im Bereich der WLAN-Ortung wurde sehr bald ersichtlich, dass diese Angaben jedoch nicht durchgehend eingehalten werden. Nach der Kalibrierung führen schon kleinste Änderungen im Ortungsbereich zu extremen Abweichungen. Zu den Störungen zählen unter anderem Personen, welche aufgrund der enthaltenen großen Wassermenge die Funksignale dämpfen. Weiterhin führen Änderungen an der Infrastruktur sowie dynamische Signalstärkenanpassungen seitens der Access Points (Stichwort „Green IT“) zu Fehlmessungen. Es muss somit von größeren Abweichungen im Bereich von drei bis zehn Metern ausgegangen werden. Dies ist jedoch nicht ausreichend für eine Indoor-Ortung.

⁴Near Field Communication

Daher beschäftigt sich ein großer Forschungsbereich mit der algorithmischen Ortungsverbesserung dieser *einfachen* Systeme ohne den Anwender unnötig zu belästigen. Als Basis für die Algorithmen bietet es sich an Gebäudemodelle zu verwenden, mit deren Hilfe logische Auswertungen getroffen werden können. CAD-Systeme für die Modellierung von Gebäuden werden nicht mehr nur von Architekten genutzt, sondern sind auch für andere Disziplinen interessant. So verwendet Google schon seit längerer Zeit 3D-Gebäudemodelle für ihre Straßenkarten als zusätzliche Orientierungshilfe in Großstädten. Diese werden beispielsweise in CityGML konstruiert, stellen meist jedoch nur die äußeren Strukturen der Gebäude dar (Level of Detail 3) [CityGML](#). In naher Zukunft werden 3D-Modelle nicht nur Geometrien, sondern auch höhere Detaillierungsgrade wie Stockwerke, Türen und Ausstattungen anzeigen. „IndoorOSM“ bietet einen Ausblick welche Informationen uns diesbezüglich in Zukunft zu Verfügung stehen könnten [OpenStreetMap Wiki](#). Der *Open Source*-Gedanke bietet hier neue Möglichkeiten der frei verfügbaren und flächendeckenden Modellierung der Umgebung, da es keine hinderlichen proprietären Formate gibt, sondern diese standardisiert und austauschbar sind. Die Abbildung des Außenbereichs, wie z. B. der Straßen, Gebäude oder Points of Interests wird in absehbarer Zeit ausreichend erfasst sein. Der Innenbereich hingegen kaum - daher dringen die Informationssammler, sei es Google oder auch freie *Mapper*, in diese Bereiche vor. Sollte Hamburg zukünftig einmal komplett in 3D modelliert sein, könnte man erstmalig ernsthaft darüber nachdenken, wie man eine präzise Ortung auch in Gebäuden flächendeckend realisieren kann. Auf Grundlage dieser Daten sollen Nutzer zukünftig infrastrukturfrei auch innerhalb von Gebäuden präzise geortet werden können, indem die bestehenden Lokalisationsdaten durch das Gebäudemodell verbessert werden. Bis es soweit ist, kann die Forschung anhand von beispielhaften Modellversuchen, die Algorithmen für eine verbesserte Indoor-Ortung entwickeln.

1.2. Ziele

Ziel der Masterarbeit soll es sein, eine Präzisionssteigerung eines bestehenden Ortungssystems zu realisieren.

Hierfür werden zwei eigene Algorithmen entwickelt, die auf dem Map-Filtering Prinzip basieren (vgl. Map-Matching 2.2). Durch einen Abgleich mit einer zugrunde liegenden Karte wird die Ortungstrajektorie logisch überprüft, gegebenenfalls korrigiert und in das Gebäudemodell eingepasst.

Die Korrekturen finden somit ohne jegliches Wissen über die Ortungstechnologie statt. Dies schließt messungsbasierte Fehlerkorrekturen aus, bietet jedoch die Möglichkeit, unabhängig

vom Ortungssystem zu sein und dieses gegebenenfalls auszuwechseln.

Eine Aussage über die zu erreichende Präzisionssteigerung kann nicht pauschal getroffen werden, da diese eng mit der jeweiligen Umgebung und dem Gebäudemodell zusammen hängt. Die Korrektur erfolgt hauptsächlich mittels baulicher Restriktionen entlang von Wänden und Türen.

Ziel ist es, die Position in einem Gebäude mindestens auf den räumlichen und semantisch korrekten Kontext wiedergeben zu können. Basierend auf einem Startpunkt wird die Bewegung einer Person durch das Gebäude betrachtet und anhand von physikalisch erreichbaren Raumwechseln korrigiert. Diese Abfolge an Raumwechseln soll der tatsächlichen Bewegungstrajektorie entsprechen. Die Korrektur der Ortungspunkte innerhalb des Raumes sollen möglichst nahe an der tatsächlichen Position liegen und nicht nur eine generelle Aussage über den aktuell betretenen Raum geben.

Sollten begrenzende Faktoren fehlen, wie z. B. in einer großen Halle, kann nicht mit der oben beschriebenen Präzisionssteigerung gerechnet werden.

Einen weiteren Faktor für die Ortungsverbesserung stellt das Ortungssystem selber dar. Die Frequenz der Positionsupdates sollte nicht geringer sein als es einem Menschen möglich ist einen Raum zu durchqueren, sodass von einem Update zum Nächsten ein Raum übersprungen wird und der logische Zusammenhang der Trajektorie verloren geht. Der maximale Jitter, also die Abweichung von der tatsächlichen Position, darf ebenfalls nicht zu groß werden, um eine Tendenz der Aufenthaltsposition erkennen zu können.

Je höher die Frequenz und kleiner der Jitter wird, um so größer wird die Wahrscheinlichkeit für eine korrekte Bestimmung des aktuellen Raumes im Bezug auf die Ortungshistorie und führt somit zu einer verbesserten Präzision.

Um die Leistungsfähigkeit der Algorithmen zu überprüfen werden repräsentative Bewegungsszenarien erstellt und anhand mehrerer Testszenarien das Regelverhalten überprüft. Beide Algorithmen erhalten die gleichen Testdaten um deren unterschiedliches Korrekturverhalten theoretisch zu diskutieren. Die Daten bestehen einerseits aus künstlich generierten Trajektorien um Spezialfälle gezielt abfragen zu können, andererseits auch aus Testläufen mit einer zufälligen Streuung. Die Ergebnisse sollen in einer vergleichenden Analyse bewertet werden.

1.3. Gliederung

Die Masterarbeit gliedert sich in folgende Abschnitte.

Zunächst wird in den Vergleichbaren Arbeiten (Kapitel 2) ein Überblick auf Technologien der Indoor-Ortung und Verfahren zur Präzisionssteigerung gegeben. Dies beinhaltet die Einord-

1. Einführung

nung des Problems sowie die aktuelle Forschung („state of the art“) auf diesem Gebiet. Basierend darauf und der Motivation aus Kapitel 1.1 wird in der Analyse (siehe Kap. 3) eine Betrachtung bezüglich der Anforderungen an das zu erstellende System vorgenommen. Ein grundlegender eigener Lösungsansatz wird vorgestellt, welcher im anschließenden Kapitel 4 umgesetzt wird. Hier wird das Vorgehen für die Realisierung geschildert und im folgenden Kapitel 5 evaluiert.

Abschließend folgt in Kapitel 6 eine zusammenfassende Betrachtung der erreichten Ziele sowie ein Ausblick auf mögliche Weiterentwicklungen und Verbesserungsvorschläge.

2. Vergleichbare Arbeiten

Für die Indoor-Ortung existieren schon seit längerem vielfältige Technologien und Ausprägungen [Liu u. a. \(2007\)](#). Die Herausforderung besteht in der möglichst hohen Präzision [Koyuncu und Yang \(2010\)](#).

Realisierungen, die sich nur auf eine Ortungstechnik verlassen, stoßen in dieser Hinsicht schnell an ihre Grenzen. So ist z. B. *RADAR* von [Microsoft Research](#) ein WLAN-basiertes Ortungssystem, dessen Präzision im Bereich von 5 bis 10 Metern liegt [Bahl u. a. \(2000\)](#). Ähnliche Umsetzungen kommen auf vergleichbare Werte [Lin und Lin \(2005\)](#).

Schon früh gab es Bemühungen die Ortung zu verbessern (vgl. [Levi und Judd \(1996\)](#) und [Martin u. a. \(2006\)](#)). Die Anforderungen von aktuellen Anwendungen aus dem Bereich der Location Awareness, unter anderem der Indoornavigation, benötigen um so dringender eine genaue Lokalisierung [Hochschule Luzern \(2012\)](#).

Dies kann durch Neuentwicklungen und hochpräzise Sensoren erfolgen, welche hohe Investitionen erfordern. Daher beschäftigt sich ein Teil der aktuellen Forschung mit der Verbesserung bestehender und somit günstigeren Ortungsverfahren. Der Vorteil liegt hierbei - je nach zugrundeliegender Technologie - bei den geringeren Kosten, da die Infrastruktur (z. B. WLAN) meistens bereits vorhanden ist.

Innerhalb dieses Forschungsgebietes haben sich zwei Entwicklungsrichtungen etabliert. Einerseits kann die Ortsverbesserung durch *zusätzliche Hardware* erfolgen, z. B. durch ein zweites, günstiges und unabhängiges Lokalisierungssystem als Referenz bzw. zur Sensorfusionierung.

Andererseits kann die Präzision auch mittels *logischer Auswertung* auf Basis von Algorithmen gesteigert werden.

Die Verwendung dualer Ortungssysteme bietet Realisierungen wie die Kombination von WLAN-Signalstärke und optischer Repositionierung mittels *Quick Response (QR)-Codes* [Hattori u. a. \(2011\)](#), Inertial Navigation System (INS) und WLAN [Chai u. a. \(2011\)](#) sowie dem Navio Projekt [Retscher \(2007\)](#), welches GPS und Beschleunigungssensoren kombiniert. Inzwischen sind auch kommerzielle Lösungen verfügbar. So bietet [Broadcom](#) den BCM4752 Chip an, welcher anhand vieler Sensordaten (GPS, WLAN, Bluetooth, Near Field Communication (NFC))

usw.) eine unterbrechungsfreie Navigation im Outdoor- sowie Indoorbereich ermöglichen soll. Die Kombination unterschiedlicher Ortungstechnologien bietet den Vorteil, dass Messfehler mit der Sensorfusionierung abgeschwächt werden und systematische Fehler durch die jeweilige andere Technologie ausgeglichen werden. Als Beispiel kann man hierfür die Inertial Measurement Unit (IMU) heranziehen. Diese bestimmt die Position durch doppelte Integration der Beschleunigung sowie der Kumulation der Messwerte. Messfehler addieren sich somit über die Zeit und führen zu großen Abweichungen im Bereich von mehr als 100 Metern bei einer Laufzeit von einer Minute Woodman (2007). Eine zusätzliche Hardware, wie z. B. ein NFC-Beacon, kann die Position in bestimmten Abschnitten aktualisieren und einen neuen Start- bzw. Referenzpunkt für die IMU-basierte Ortung liefern. Ein Nachteil wird hier deutlich, da für viele Realisierungen zusätzliche Hardware in der Umgebung installiert und kalibriert werden muss.

Die Verwendung dualer Lokalisierungshardware stellt eine mögliche Lösung zur Präzisionssteigerung dar, indem die Stärken der jeweiligen Systeme kombiniert werden und systematische Fehler ausgeglichen werden können. Die Herausforderung besteht hierbei hauptsächlich in der korrekten Sensorfusionierung durch mathematische Verfahren.

Der Ansatz zur *logischen Auswertung* bestehender Ortungssysteme kann ebenfalls die Präzision steigern. Hierfür wird nicht nur der aktuelle Ortungspunkt bestimmt, sondern auch dessen zeitliche Abfolge als Ortungshistorie aufgezeichnet. Auf dieser Datengrundlage lassen sich Plausibilitätsüberprüfungen durch Algorithmen durchführen und die gesamte Trajektorie korrigieren.

Einige der folgenden Algorithmen sind rein messwertbasiert und nutzen allgemeine Annahmen über die Umgebung und das Bewegungsverhalten von Personen, womit eine gesteigerte Präzision erzielt werden kann. Der Vorteil dieser Verfahren liegt in ihrer schnellen Installation, Portierbarkeit und Hardwareunabhängigkeit.

Anderen Auswertungsmethoden aus dem Bereich des Map-Matching liegt ein Gebäudeplan zugrunde, welcher einen Abgleich mit der Trajektorie und möglichen Bewegungspfaden durch das Gebäude zulässt. Physikalisch unzulässige Raumwechsel führen zu einer Korrektur der Ortungshistorie und liefern eine deutliche Verbesserung gegenüber den rein messwertbasierten Korrektur-Algorithmen.

Allen Verfahren gemein ist die Anreicherung der Ortungspunkte um weiteres semantisches Wissen und der daraus resultierenden Aussage über die Aufenthaltswahrscheinlichkeit. Diese Herangehensweise bietet interessante Lösungsansätze und eigene Entwicklungsmöglichkeiten zur logischen Ortungsverbesserung.

Im folgenden Kapitel werden daher zunächst hybride Verfahren vorgestellt (Kapitel 2.1), welche ein bestehendes Ortungssystem allein anhand von Messwerten und logischer Auswertung verbessern. Weiterführend werden in Kapitel 2.2 vergleichbare Arbeiten aus dem Bereich des Map-Matching vorgestellt, welche eine Spezialisierung darstellen und gleichzeitig für eine verbesserte Ortungsgenauigkeit sorgen. Ein besonderes Augenmerk liegt hierbei auf den Algorithmen, welche für einen möglichen Einsatz im Framework untersucht werden.

2.1. Ortungsverbesserung auf Grundlage von rein messwertbasierten Algorithmen

Ein Teil des Forschungsbereiches hat sich auf die alleinige algorithmische Messwertverbesserung spezialisiert. Das bedeutet, dass weder zusätzliche Hardware, noch ein umfassendes Wissen über die Umgebung benötigt wird. Die Verfahren werten die Ortungshistorie aus, indem Bewegungsheuristiken darauf abgebildet werden. Ein klarer Vorteil liegt hierbei offensichtlich bei dem geringen Installationsaufwand, dieser muss aber auch gegenüber dem erreichbaren Nutzen abgewogen werden.

2.1.1. Sensorbereinigung durch Glättung und Plausibilitätsprüfung

Die Positionsbestimmung erfolgt mittels Sensoren und unterliegt gewissen Messtoleranzen. Hierbei ist es gleichgültig um welches Ortungssystem es sich handelt, sei es IMU- oder WLAN-basiert.

Die Messfehler liegen meist Gauß-verteilt um die tatsächliche Position und sorgen für den sogenannten Jitter. Einzelne Messwerte weisen jedoch eine extreme Abweichung auf, welche durch zufällige Störgrößen hervorgerufen werden.

In der Masterarbeit von Voskuhl (2012) wird eine Plausibilitätsprüfung vorgestellt. Hierbei wird vorgeschlagen, die Daten zu bewerten und im Vorfeld zu bearbeiten, bevor sie für weitere Analysen verwendet werden. Die Schwierigkeit besteht in der Erkennung von *Ausreißern*. Um fundierte Aussagen und Kategorisierungen zu treffen, behilft man sich mit Algorithmen aus der Statistik Hartung u. a. (2005).

In Abbildung 2.1 sieht man den Grundriss des *Living Place* sowie links die Rohdaten des Ortungssystems und rechts die daraus gebildeten Trajektorien (Splines). Die Farben geben das zur jeweiligen Trajektorie erkannte Bewegungsmuster aus der Activity Recognition Phase wieder. Nicht erkannte Pfade werden schwarz dargestellt. Diese gilt es anhand der Filter-Algorithmen zu minimieren.

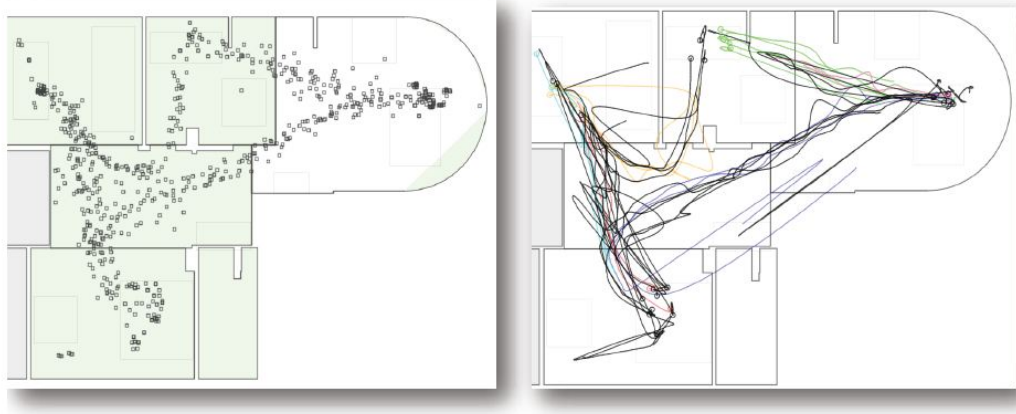


Abbildung 2.1.: Rohdaten und daraus gebildete Trajektorien Voskuhl (2012)

Ein einfacher Ansatz stellt das arithmetische Mittel (siehe Formel 2.1) dar. Auf dieser Basis wurde ein Filter für die Trajektorie realisiert, welcher einen Durchschnittswert über den aktuellen Ortungspunkt und seine unmittelbaren Nachbarn bildet. Die Ergebnisse zeigten eine leichte Abschwächung der Ausreißer, sorgten jedoch bei extremen Abweichungen für unzureichende Korrekturen.

Ein Extremwert-unabhängiger Algorithmus bietet hier Vorteile. So wird der Median (vgl. Formel 2.2) verwendet, um eine Maßzahl einer Beobachtungsreihe zu bestimmen. Dieses Lagemaß ist für die Erkennung von Ausreißern nicht aussagekräftig. Hierfür muss zusätzlich das Streuungsmaß mittels der Formel 2.3 bestimmt werden, womit die mittlere absolute Abweichung zum berechneten Zentrum beschrieben wird.

$$\text{Arithmetisches Mittel: } \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (2.1)$$

$$\text{Median: } \bar{x}_{med} = \begin{cases} x_{\frac{n+1}{2}} & \text{falls } n \text{ ungerade} \\ \frac{1}{2} (x_{\frac{n}{2}} + x_{\frac{n}{2}+1}) & \text{falls } n \text{ gerade} \end{cases} \quad (2.2)$$

$$d = \frac{1}{n} \sum_{i=1}^n |x_i - \bar{x}_{med}| \quad (2.3)$$

Durch diese Bestimmung ist es nun mittels eines Filters möglich, Ausreißer zu identifizieren. Hierbei werden jeweils fünf aufeinanderfolgende Punkte einer Trajektorie ausgewählt, der Median gebildet und gespeichert. Die Start- und Endpunkte werden dabei beibehalten. Für die

2. Vergleichbare Arbeiten

weiterführende Auswertung hat dies zur Folge, dass zwar Ausreißer entfernt werden, jedoch auch korrekte Messwerte verloren gehen. Je nach gewählter Wegstreckenlänge führt dies wie in Abb. 2.2 ersichtlich, zu erheblichen Abweichungen bezüglich dem tatsächlich zurückgelegten Pfad.

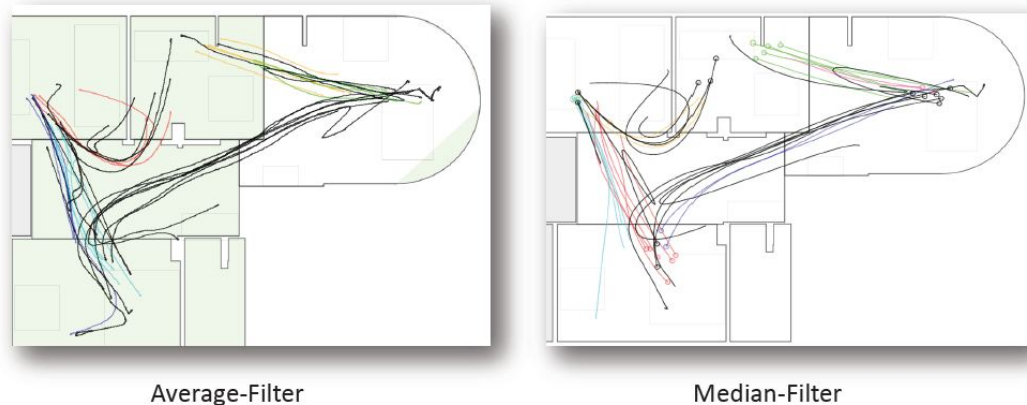


Abbildung 2.2.: Vergleich der beiden Filter-Algorithmen Voskuhl (2012)

Die Verwendung des Median-Filters stellt demnach keine optimale Lösung der Ausreißer-Problematik dar.

Grundlegend empfiehlt es sich daher, Sensordaten nicht gutgläubig zu vertrauen und eine alleinige Mittelung der Rohdaten durchzuführen. Für die Indoor-Ortung können einige Annahmen getroffen werden, welche eine Plausibilitätsprüfung der Messwerte erlauben. Hierbei werden beispielsweise alle Messwerte größer einer festgelegten Distanz als Ausreißer deklariert Grubbs (1969).

In den hier betrachteten Szenarien werden Personen geortet, welche sich zu Fuß fortbewegen. Bei einer Bewegung durch das Gebäude können diese nur eine definierte maximale Geschwindigkeit annehmen. Durch eine Betrachtung des Abstandes zwischen dem aktuellen Ortungspunkt und dessen Vorgänger sowie der Zeitdifferenz, kann die dafür benötigte Geschwindigkeit errechnet werden. Sollte diese über der für menschliche Verhältnisse möglichen Höchstgeschwindigkeit liegen, kann davon ausgegangen werden, dass der aktuelle Ortungspunkt einem großen Messfehler unterliegt und muss dementsprechend bewertet werden.

In der Masterarbeit von Voskuhl (2012) wird das Problem der zu hohen Trajektorienglättung dadurch gelöst, dass zunächst ein Radius-Filter angewendet wird. Hierbei werden Messwerte, welche zu ihrem Vorgänger einen größeren Abstand als 1,5 m aufweisen, als Fehlmessungen deklariert und verworfen. Dies ergibt sich aus der Updaterate des Ortungssystems von 0,5 Sekunden und einer experimentell festgestellten maximalen Bewegungsgeschwindigkeit von

2. Vergleichbare Arbeiten

Personen im Raum. Nach dieser Bereinigung von Ausreißern, wird der zuvor beschriebene Median-Filter entsprechend abgeändert eingesetzt um die Trajektorien zu glätten. Das Ergebnis (vgl. Abb. 2.3) weist eine deutliche Verbesserung gegenüber der alleinigen Anwendung des Median-Filters auf.

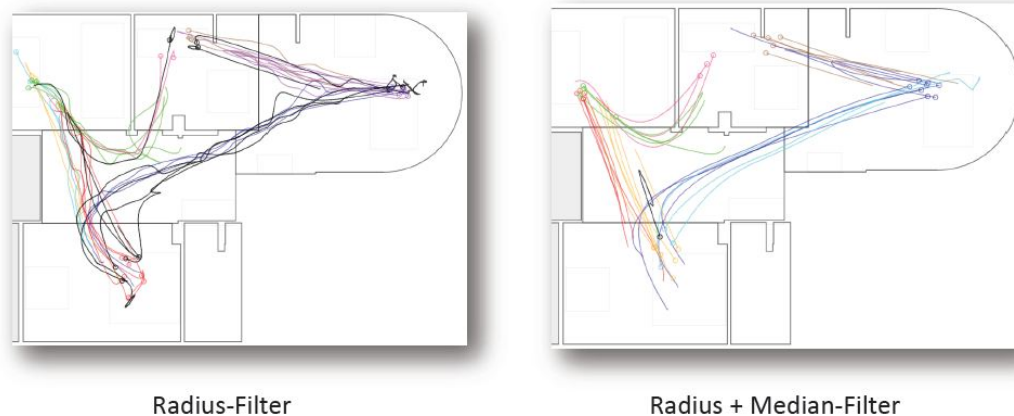


Abbildung 2.3.: Verbesserte Ergebnisse durch Kombination von Filtern Voskuhl (2012)



Abbildung 2.4.: Living Place Einrichtung Voskuhl (2012)

Die Anzahl erkannter Trajektorien (farbig) wurde gegenüber den Rohdaten erheblich verbessert. Vergleicht man die Pfade mit einem Grundrissplan mit Einrichtungsgegenständen (z. B. Küchenzeile und Tischen in Abb. 2.4), so werden diese noch an ein paar Stellen geschnitten, welches einer natürlichen Bewegung widerspricht. Für höher stehende Auswertungsschichten

bieten diese bereinigten Positionsdaten eine abstraktere Übersicht und verbesserte Erkennungsrate bezüglich Bewegungsmustern.

2.1.2. Drift Compensation

IMU-Sensoren messen die Beschleunigung, Rotation und Orientierung. Gerade bei der Bestimmung der Orientierung mittels Magnetometern kann es bauartbedingt zu Messabweichungen kommen. Da die Messwerte über die Zeit akkumuliert werden um auf die aktuelle Position zu schließen, sorgt dieser sogenannte Drift für Fehlpositionierungen, welche möglichst minimiert werden soll. Nachfolgend wird ein Algorithmus von [Dorveaux und Petit \(2011\)](#) vorgestellt, der laut Angaben der Entwickler dem Drift entgegen wirken kann, indem die Sensoren automatisiert kalibriert werden. Im Anschluss wird ein zweiter Algorithmus von [Jimenez u. a. \(2011\)](#) beschrieben, der ebenfalls eine Driftkompensation bietet und dessen Kalibrierung auf Annahmen über die Bewegungstrajektorien im Gebäude basiert.

Automatisierte Sensorkalibrierung

Die Schwierigkeit bei einem IMU-gestützten Ortungssystem stellt die Verschmelzung der einzelnen Systemkomponenten dar. Jeder Sensor hat seinen eigenen Betrachtungsrahmen in Bezug auf das Koordinatensystem. Für eine Verortung der Person müssen die Geschwindigkeits- und Lagesensoren kombiniert und in ein gemeinsames Koordinatensystem überführt werden. Durch bauliche Schwankungen können schon kleinste Abweichungen mit der Zeit zu großen Positionsfehlern führen. Dies wird in [Abbildung 2.5](#) deutlich. Dort werden mehrere geschlossene Pfade abgelaufen (grüne Trajektorie) und mittels einer IMU gemessen (rote Trajektorie). Durch eine leichte Pitch-Verschiebung¹ sind diese Zyklen nicht mehr geschlossen und führen nach mehreren Runden zu einer großen Abweichung.

Mittels einer Harmonisierungs-Matrix versuchen [Dorveaux und Petit \(2011\)](#) die unterschiedlichen Rahmen gegeneinander auszurichten. Diese Kalibrierung soll durch ein einfaches Verfahren ermöglicht werden, welches leicht vor Ort durchgeführt werden kann. Hierzu wird mit konstanter Geschwindigkeit ein geschlossener Pfad abgelaufen. Start- und Endpunkt sind identisch und lassen somit Rückschlüsse auf Messabweichungen zu.

$$P_r(T) - P(0) = \int_0^T R_{att}(t) V_{vel} \cdot dt \quad (2.4)$$

¹Konstante Abweichung auf der z- Achse

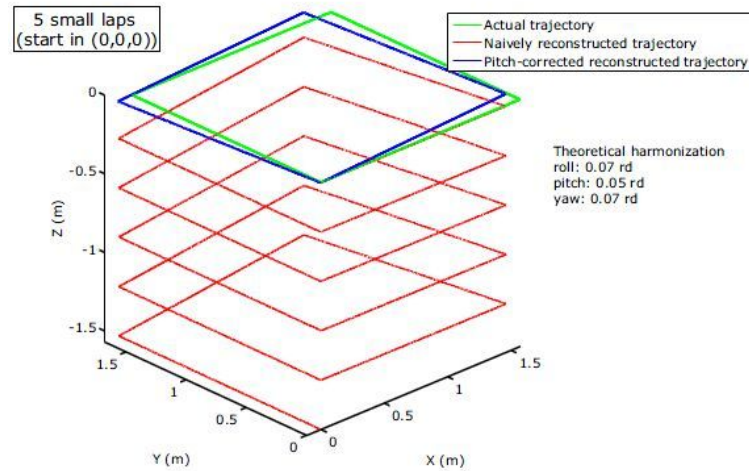


Abbildung 2.5.: Kein geschlossener Zyklus (rot) durch einen konstanten Pitch-Fehler in der z-Achse [Dorveaux und Petit \(2011\)](#)

Die Gleichung 2.4 liefert die zurückgelegte Trajektorie, wobei $P(0)$ der Start- und $P_r(T)$ der Endpunkt ist. Desweiteren ist hier R_{att} die Rotationsmatrix zwischen IMU-Sensor und inertialem Betrachtungsrahmen und V_{vel} die Geschwindigkeit, über welche integriert wird.

Um den Pitchwinkel θ zu bestimmen (unter der Annahme, dass es sich um eine geschlossene Trajektorie handelt) wird Gleichung 2.5 verwendet.

$$\sin(\theta) = \| P_r(T) - P(0) \| \cdot \frac{-1}{V_0 \cdot T} \quad (2.5)$$

Anhand dieser Formeln ist es möglich, den Drift der Trajektorie zu minimieren (vgl. 2.5, blaue Trajektorie), indem die Harmonisierungs-Matrix die Messwerte der Sensoren in Einklang bringt. Zusätzlich lässt sich diese Kalibrierung leicht durchführen. Dieser Algorithmus wird für die initiale Einrichtung benötigt um konstante Abweichungen der Sensoren auszugleichen, welches eine Verbesserung der Ortungsgenauigkeit darstellt. Für die Verwendung im eigenen Framework bietet sich diese Umsetzung in zweierlei Hinsicht nicht an:

Einerseits ist der Algorithmus für relative Ortungssysteme ausgelegt und somit nicht nutzbar für absolute Realisierungen (z. B. WLAN-basiert, oder [Ubisense \(2013\)](#)). Andererseits stellt dieser Algorithmus wie bereits angesprochen, nur eine Art der Sensorkalibrierung dar und gehört zu einer vorgelagerten Korrekturphase.

Improved Heuristic Drift Elimination

Einen weiteren Ansatz zur Driftkompensierung bieten [Jimenez u. a. \(2011\)](#) an.

Ähnlich dem Map-Matching aus Kapitel 2.2, werden bei diesem Verfahren Annahmen über die Umgebung genutzt, um die Messwerte der Sensoren zu überprüfen und zu korrigieren. Diese Annahmen basieren jedoch nicht auf Karten oder einem konkreten Gebäudemodell, sondern auf dem typischen Aufbau von Gebäuden sowie Bewegungsmustern von Menschen.

Ihr iHDE²-Algorithmus nutzt die Tatsache, dass die meisten Korridore senkrecht bzw. in einem 45° Winkel zueinander ausgerichtet sind. Es wird angenommen, dass sich Personen ebenfalls geradlinig entlang dieser Gänge bewegen. Die IMU-basierten Trajektorien weisen durch ihre akkumulierten Orientierungsfehler eine Krümmung auf, welche durch den Algorithmus „geglättet“ werden, indem sie in rechtwinklige sowie diagonale Strecken in das Framework von [Jimenez u. a. \(2010\)](#) eingepasst werden. In Abbildung 2.6 wird die Präzisionssteigerung anhand eines Beispiels verdeutlicht.

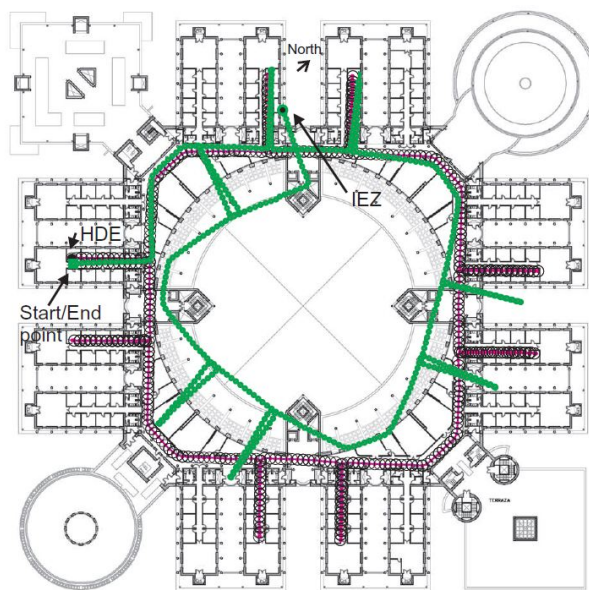


Abbildung 2.6.: Ohne iHDE (grün), mit iHDE (lila) [Jimenez u. a. \(2011\)](#)

Das IMU-basierte Ortungssystem leidet mit der Zeit unter einer extremen Abweichung zur tatsächlichen Position (grüne Trajektorie). Mit Hilfe der iHDE wird basierend auf einer dominierenden Hauptbewegungsachse, die Trajektorie zu den entsprechend erlaubten Bewegungsrichtungen (0°, 45°, 90°, usw.) ausgerichtet und korrigiert (lila Trajektorie).

²Improved Heuristic Drift Elimination

2. Vergleichbare Arbeiten

Der Algorithmus basiert auf den HDE-Ansätzen von [Abdulahim u. a. \(2010\)](#), welche diese Korrekturen auf alle Trajektorien angewendet hat. So werden z. B. geradlinige Bewegungen entlang einer abweichenden Bewegungsachse mit dem HDE-Algorithmus fälschlicherweise korrigiert (vgl. Abb. 2.7 a), oder gekrümmte Trajektorien (Kreisbahnen) verzerrt (vgl. Abb. 2.7 b).

Dieses Fehlverhalten wird mit dem improved HDE-Algorithmus von [Jimenez u. a. \(2011\)](#) behoben, indem die Korrekturen nur in bestimmten Szenarien angewendet werden (minimale Schrittlänge und fünf aufeinander folgende Schritte in gleiche Richtung). Bei einer gekrümmten Bewegung findet somit keine Korrektur oder Repositionierung statt, wie in Abbildung 2.8 zu sehen ist. Ebenfalls werden Trajektorien ab einer definierter Länge, entlang einer nicht kategorisierbaren Bewegungsrichtung, unverändert übernommen.

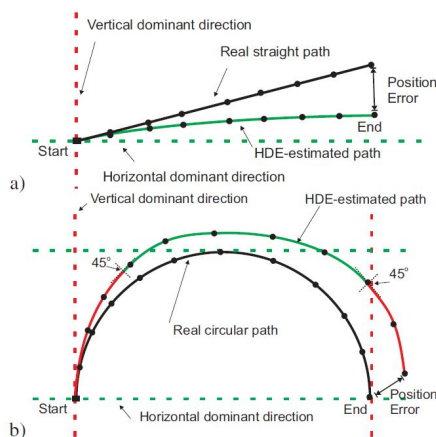


Abbildung 2.7.: HDE Algorithmus [Borenstein und Ojeda \(2010\)](#) Problemfälle

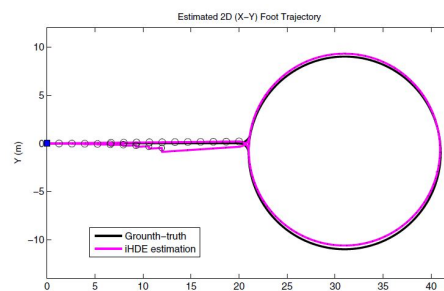


Abbildung 2.8.: iHDE Korrektur (lila) auf einer kreisförmigen Trajektorie (schwarz) [Jimenez u. a. \(2011\)](#)

In einem idealen orthogonalen Gebäude mit engen Korridoren fallen die Ergebnisse beider Algorithmen (HDE = 0,33% TTD und iHDE = 0,2% TTD) nahezu identisch aus. Für nicht ideale Trajektorien wie z. B. gekrümmte Pfade, übertrifft der iHDE-Algorithmus seinen Vorgänger (vgl. Abb. 2.8 gegenüber 2.7). Dennoch basiert dieser Ansatz auf stark einschränkenden Annahmen bezüglich der Umgebung und der voraussichtlichen Bewegung von Personen. Der iHDE-Algorithmus besitzt zwar eine höhere Präzision als sein Vorgänger, stößt aber an seine Grenzen, sobald die Annahme über begrenzende enge Korridore nicht mehr zutrifft (z. B. große Freiflächen oder Flughäfen) und geradlinige Bewegungen nicht zwingend zu erwarten sind.

Weitere hybride Lokalisierungssysteme sind IMU-basiert, wie z. B. [Krach und Robertson \(2008\)](#), [Woodman und Harle \(2008\)](#) oder [Beauregard u. a. \(2008\)](#). Die Algorithmen sind auf die zuvor beschriebenen besonderen Eigenschaften von IMU-Sensoren spezialisiert und benötigen relative Ortungsinformationen. Sie lassen sich nicht ohne Weiteres auf absolute Ortungssysteme übertragen, welche unabhängige Positionsangaben liefern und in der eigenen Umsetzung verwendet werden sollen. Daher wird im folgenden Kapitel ein Algorithmus zur Präzisionssteigerung für ein absolutes Ortungssystem vorgestellt.

2.1.3. Automatisierte RSSI-Berechnung auf Basis von Kartendaten

Im Gegensatz zu relativen IMU-Ortungssystemen, liefern WLAN-basierte Realisierungen eine von den vorherigen Messungen unabhängige, absolute Position. Hierbei stützt man sich entweder auf das Fingerprinting-Verfahren oder die Analyse der Signalausbreitung.

Bei dem ersteren Verfahren wird während einer Trainingsphase (Offline-Phase) an möglichst vielen Stellen die Signalstärke (RSSI³) zu mehreren Access Points gemessen und in einer Karte verortet. Erst nach dieser initialen Kalibrierung kann man in die Online-Phase wechseln und das Ortungssystem nutzen. Für eine Lokalisierung wird die aktuelle Signalstärke der empfangbaren Access Points bestimmt und mit den zuvor gespeicherten Messwerten verglichen. Die grundlegende Funktionsweise basiert darauf, dass die größte Übereinstimmung mit den gesammelten Daten als aktueller Standpunkt angenommen wird.

Es existieren diverse Verbesserungsansätze zum optimierten Abgleich mit den Offline-Messwerten, wie z. B. der Interpolation einer Position zwischen mehreren Messwerten [Liu u. a. \(2007\)](#) oder der automatisierten Berechnung der Datenbasis [Wang \(2011\)](#).

Ein weiterer Ansatz versucht die Signalausbreitung und insbesondere deren Abschwächung durch Hindernisse zu berechnen und anhand der aktuell gemessenen Signalstärken auf den aktuellen Standort zu schließen.

Das folgende Verfahren von [Mestre u. a. \(2011\)](#) kombiniert beide Ansätze, um eine vereinfachte Installation und erhöhte Ortungsgenauigkeit zu erzielen. Die Arbeitsgruppe um Pedro Mestre der UTAD⁴ hat ein Verfahren entwickelt, mit dem es möglich ist, die zeitintensive Kalibrierung zu vereinfachen. Eine Karte dient hierbei als Grundlage für die Berechnung der Signalausbreitung. Dabei werden die dämpfenden Eigenschaften von Wänden und sonstigen Hindernissen berücksichtigt. Die daraus resultierende „Heatmap“ kann für eine automatische Generierung der Fingerprintingtabelle verwendet werden. Eine zeitaufwändige manuelle Ein-

³Received Signal Strength Indication

⁴Universidade de Trás-os-Montes e Alto Douro

messungsphase entfällt und ermöglicht eine schnelle Anpassung an bauliche Veränderung, sobald diese auf der Karte eingetragen werden.

$$L = L_{FS} + L_C + \sum_{i=0}^I k_{wi} L_{wi} + L_f^{\frac{k_f+2}{k_f+1}-b} \quad (2.6)$$

$$L_{FS} = L_0 + 10N \log d \quad (2.7)$$

Für die Berechnung der Signalstärkenabschwächung durch ein System bestehend aus mehreren Wänden und Türen, wird Gleichung 2.6 aus dem COST231 Modell von Action (1999) verwendet. L_C steht für den konstanten Verlust in dB, k_{wi} für die Anzahl zwischenliegender Wände vom Typ i mit dem Verlustfaktor L_{wi} , k_f die Anzahl von Geschossdecken mit dem Verlustfaktor L_f , b ist ein empirischer Parameter und I ist die Anzahl unterschiedlicher Wandtypen. L_{FS} ist der Verlust im freien Raum. Wobei L_{FS} mittels Gleichung 2.7 berechnet wird. Hierbei ist L_0 der Verlust auf einen Meter, N ein Index für die Verluststärke und d die Entfernung zwischen Sender und Empfänger in Metern. Für eine Steinwand wird eine Dämpfung von 9,00 dB und für eine Holztür 3,04 dB angenommen.

Auf Grundlage der Karte, stationären Access Points und Referenzpunkten zur Kalibrierung, kann mit Hilfe von Gleichung 2.6 für jeden beliebigen Punkt im Gebäude der entsprechende Signalstärkeneintrag in der Tabelle berechnet werden. Die Anzahl der dazwischenliegenden Wände und deren jeweiliger Dämpfungsfaktor fließt in die Berechnung mit ein, um ein möglichst realistisches Abbild der Signalstärkenausbreitung im Raum zu erhalten.

Auf Grundlage dieser Daten werden unterschiedliche Algorithmen⁵ zur Ortungsabschätzung getestet, um Aussagen über die Qualität der erzeugten Heatmap machen zu können. Die durchgeführten Experimente und Messreihen von Mestre u. a. (2011) ergeben die beste Genauigkeit von 3,249 m für das k-Nearest Neighbour Verfahren und die beste Standardabweichung von 1,964 m für den Fuzzy Logic Algorithmus.

Dieses Verfahren ist bereits nicht mehr ohne Weiteres an einen beliebigen Ort portierbar, da für die Berechnung der Heatmap eine möglichst detaillierte Umgebungskarte nötig ist. Des Weiteren müssen bauliche Veränderungen umgehend in die digitalen Karten übertragen werden und eine erneute Berechnung angestoßen werden, um Fehlpositionierungen zu verhindern. Es wird ersichtlich, dass eine effektive Ortungsverbesserung einhergeht mit dem gesteigerten Wissen über die Umgebung.

Da die hier verwendete Karte nur für die initiale Erstellung benötigt wird, stellt dieses Verfahren

⁵Nearest Neighbour, k-Nearest Neighbour, Weighted k-Nearest Neighbour und Fuzzy Logic Serodio u. a. (2011)

einen Übergang zu den folgenden Map-Matching-Verfahren dar, lässt sich aber noch nicht als selbiges kategorisieren.

2.2. Map-Matching Verfahren

Wie in Kapitel 2.1.3 angesprochen, kann eine Karte als zusätzliches Hilfsmittel verwendet werden. In der Fachliteratur wird dieses Verfahren unter dem Begriff des Map-Matching bzw. Map-Filtering zusammengefasst, welches Karteninformationen als a priori-Wissen verwendet um Positionsschätzungen durch Sensorfusionierung zu verbessern [Angermann u. a. \(2008\)](#).

Die Ortungspositionen werden auf einem 2D-Gebäudeplan mit Wänden und Türen abgebildet. Aus den sich daraus ergebenden Einschränkungen für die mögliche Bewegungsrichtung wird eine Korrektur des Pfades vorgenommen.

Viele Realisierungen sind IMU-basiert, wie z. B. „PeLoTe“ [Saarinen und Heikkila \(2005\)](#), „wearIT@WORK“ [Beauregard \(2007\)](#), „DLR Advanced Navigation Applications“ [Krach und Roberston \(2008\)](#) oder SLAM⁶-basierte Kartenerstellung und Navigation [Driewer u. a. \(2004\)](#). Die Map-Matching Verfahren sind ähnlich und werden daher in den Unterkapiteln bezüglich der jeweiligen Hauptunterschiede dargestellt.

Zunächst folgt in Kapitel 2.2.1 eine kurze historische Einführung, gefolgt von einem generellen Verfahren zum Map-Matching (Kap. 2.2.2).

Anschließend werden die Map-Matching Algorithmen in zwei Kategorien aufgeteilt - die relativen (siehe Kap. 2.2.3) und absoluten (siehe Kap. 2.2.4) Ortungssysteme - da sich deren Eigenschaften grundlegend unterscheiden.

In einem Fazit (siehe Kapitel 2.3) werden die Verfahren bewertet und verglichen, um ein geeignetes Referenzsystem für die eigene Realisierung zu finden.

2.2.1. Historische Entwicklung

Die Idee des Map-Matching ist nicht neu, denn schon vor Jahren wurden ähnliche Verfahren für die Outdoornavigation verwendet. Diese stammen aus dem Sektor des Terrain-Matchings, wie z. B. die Cruise-Missile [Hicks \(1993\)](#). Es existieren aber auch neuere Entwicklungen wie der *Lunar Lander* [Johnson und Montgomery \(2008\)](#) oder die Navigationshilfe für Unterwasserfahrzeuge [Strauss u. a. \(1999\)](#). Ein Kalman Filter sorgt für die nötige Sensorfusionierung sowie die Korrelation mit der hinterlegten Karte.

Für den Indoorbereich gibt es ebenfalls seit längerer Zeit die Idee des Kartenabgleichs. Dies erfolgt bei NavMote [Fang u. a. \(2005\)](#) jedoch noch manuell durch einen Operator.

⁶Simultaneous Localization and Mapping

Für die Straßennavigation in Verbindung mit absoluten GPS-Ortungspunkten gibt es mehrere Algorithmen zum Kartenabgleich [White u. a. \(2000\)](#) und [Eisner u. a. \(2011\)](#). Diese machen sich den allgemeinen Straßenaufbau zu Nutze, indem Straßen und Kreuzungen zu einem Knoten- und Kantenmodell abstrahiert werden und die geschätzte Position möglichst logisch korrekt auf eine nahegelegene Kante verschoben wird.

Im Gegensatz zur restriktiven Bewegungsmöglichkeit auf der Straße, ist die Indoor- Fußgänger- navigation komplexer und lässt sich nicht ohne Einschränkungen auf ein Knoten- und Kantenmodell abstrahieren, da eine freie Bewegung durch den Raum möglich ist.

In folgendem Kapitel werden die Grundlagen dieser Herangehensweise näher betrachtet und die daraus resultierenden Einschränkungen erörtert.

2.2.2. Matching Verfahren auf Basis von Knoten- und Kantenmodellen

Zunächst wird ein prinzipielles Map-Matching Verfahren zum Abgleich einer Soll- zu einer Ist-Trajektorie vorgestellt. Anhand des Gebäudemodells, dem Start sowie dem Zielpunkt der Navigation wird die Soll-Trajektorie berechnet. Die Ist-Trajektorie ergibt sich aus der Aneinanderreihung der gemessenen Ortungspunkte.

Die naheliegendste Herangehensweise zur Verbesserung der Indoorlokalisierung ist ein einfaches „Point to Point“ bzw. „Point to Curve“ Matching. [Gilliéron u. a. \(2004\)](#) verwenden als Grundlage für ihren Algorithmus einen vorgesehenen Navigationspfad, um den aktuellen Ortungsmesspunkt daran auszurichten und somit zu verbessern.

$$\sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \qquad c = \sqrt{\frac{\left((y_1 - y_2)^2 x_p + (x_2 - x_1)^2 y_p + (x_1 y_2 - x_2 y_1) \right)^2}{(y_1 - y_2)^2 + (x_2 - x_1)^2}}$$

Abbildung 2.9.: Point to Point Matching [Gilliéron u. a. \(2004\)](#)

Abbildung 2.10.: Point to Curve Matching [Gilliéron u. a. \(2004\)](#)

Das Point to Point Matching (vgl. Abb. 2.9) bestimmt den Abstand zwischen zwei Raumkoordinaten, basierend auf dem Satz des Pythagoras. Wohingegen das Point to Curve Matching (vgl. Abb. 2.10) eine Erweiterung darstellt und eine Distanz zwischen einem Punkt und einer Geraden berechnet.

Um eine Berechnung der minimalen Entfernung zwischen einem Ortungspunkt und dem Navigationspfad zu ermöglichen, ist es nötig zuvor einen Graph - bestehend aus Knoten und Kanten - aus dem Gebäudemodell zu extrahieren. Die Mittellinien der Korridore bilden die Hauptachsen (Kanten), angrenzende Türen werden auf die Kanten projiziert und bilden Knotenpunkte zu den Räumen (vgl. Abb. 2.11).

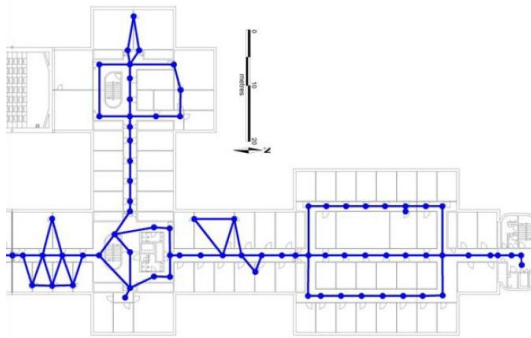


Abbildung 2.11.: Routenpfade innerhalb eines Gebäudes **Gilliéron u. a. (2004)**

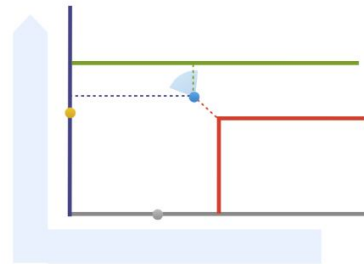


Abbildung 2.12.: Wahl des richtigen Pfades **Manh-Hung Vo Le (2009)**

Der vom Ortungssystem bestimmte Punkt wird nun anhand des geringsten Abstandes zu einer Kante hin verschoben (Abb. 2.10). Zusätzlich ist es nötig eine Gewichtung der infrage kommenden Kanten vorzunehmen, um eine Fehlpositionierung zu minimieren. Hierbei spielen mehrere Faktoren, u. a. die Bewegungsrichtung, der ursprünglich geplante Navigationspfad, der zurückgelegte Pfad sowie der Abstand zum Ziel, für den Bewertungsfaktor eine entscheidende Rolle.

In Abbildung 2.12 wird dies anhand eines Beispiels verdeutlicht. Der graue Punkt ist die letzte Ortungsposition, der gelbe Punkt ist die tatsächliche aktuelle Position und blau die fehlerbehaftete Schätzung vom Ortungssystem. Die graue Kante stellt den geplanten Navigationspfad dar, rot und grün sind abzweigende Gänge. Sollte die Position zur nächstgelegenen Kante (grün und rot) korrigiert werden, ist dies nicht die beste Wahl, da eine Verschiebung hin zum geplanten Navigationspfad (grau) logischer wäre.

Dieses Verhalten muss anhand von Gewichtungsfaktoren in die Bewertung einfließen.

Bei diesem Algorithmus ist ein zu Beginn vorgegebener Navigationspfad nötig, um die Ortungsgenauigkeit zu verbessern.

Start- und Zielpunkt sind festgelegt, wodurch eine freie Bewegung im Raum durch diesen Algorithmus nicht korrigiert werden kann, da wichtige Informationen für den Bewertungsfaktor nicht zur Verfügung stehen und eine Fehleinschätzung somit wahrscheinlicher wird und zudem eine logische Überprüfung auf illegale Raumübergänge nicht durchgeführt wird.

2.2.3. Relative Ortungsverfahren

Für das Map-Matching ist das zugrunde liegende Ortungsverfahren ein entscheidender Faktor für die Wahl des passenden Algorithmus. In diesem Kapitel werden Algorithmen vorgestellt,

die auf relative Ortungssysteme zugeschnitten sind. Unter *relativ* wird in diesem Zusammenhang verstanden, dass die Position ausgehend von dessen direkten Vorgänger angegeben wird. Die meisten IMU⁷-Systeme basieren auf dieser Art von Ortsangabe, da sie von einem bekannten Ausgangspunkt die folgenden Positionsänderungen relativ zu diesem aufsummieren. Somit akkumulieren die Systeme über die Zeit ihre Messfehler und müssen von den darauf zugeschnittenen Map-Matching Algorithmen entsprechend ausgeglichen werden.

In den folgenden Abschnitten werden unterschiedliche relative Ortungsverfahren und deren Korrekturalgorithmen vorgestellt.

„Footpath“ (Kap. 2.2.3) versucht die Bewegungstrajektorie in einen zuvor festgelegten Navigationspfad einzupassen, um so auf die aktuelle Position zu schließen. Hierfür werden Start- und Zielpunkt benötigt, wobei eine freie Bewegung durch das Gebäude nicht korrigiert werden kann was eine erhebliche Einschränkung der Nutzungsmöglichkeiten darstellt.

Basierend auf diesem Nachteil wird in 2.2.3 ein Verfahren dargestellt, welches es ermöglicht, eine aufgezeichnete Trajektorie in einen Gebäudeplan einzupassen. Es werden keine absoluten Start- und Endpunkte mehr benötigt, die zurückgelegte Trajektorie muss jedoch lang genug und eindeutig bezüglich des Grundrisses sein, damit die Startposition konvergiert und eine einzelne Lösung geliefert wird.

„CADMS“ (Kap. 2.2.3) korrigiert die zurückgelegte Bewegungstrajektorie anhand eines Gebäudemodells, indem Raumwechsel nur durch Türen erlaubt und an diesen ausgerichtet werden. Sollte die Trajektorie Wände kreuzen, wird erstere segmentweise verschoben, um innerhalb des Raumpolygons zu verbleiben.

In Abschnitt 2.2.3 wird ein Algorithmus zur Pfadglättung vorgestellt, da die meisten Korrekturen zu Sprüngen und extremen Krümmungen der Trajektorie führen, was nicht einer natürlichen Bewegung entspricht.

Partikelfilter werden häufig für IMU-basiertes Map-Matching herangezogen und in 2.2.3 gesammelt vorgestellt. Sie erstellen *Cluster* von möglichen Aufenthaltsorten, welche gegenüber dem Gebäudemodell auf Plausibilität überprüft und bewertet werden.

Als spezielles Beispiel wird in Kapitel 2.2.3 ein Backtracking Partikelfilter vorgestellt. Dieser verfügt über zusätzliches Wissen über seine Historie, das ihm ermöglicht, alternative Routen rückwirkend zu erschließen, sobald der aktuelle Pfad z. B. durch eine Sackgasse unwahrscheinlich wird.

Im Anschluss wird in Kapitel 2.2.3 ein Algorithmus vorgestellt, der auf Grundlage des menschlichen Bewegungs- und Gebäudemodells eine Wahrscheinlichkeitsfunktion für den zukünftigen

⁷Inertial Measurement Unit

Aufenthaltort bietet und somit eine Bewertung und Korrektur des aktuellen Ortungspunktes zulässt.

FootPath: Abgleich von Soll- und Ist-Trajektorien

Footpath ist ein kartenbasiertes Navigationssystem und bietet zwei wählbare Map-Matching Algorithmen. Es basiert auf der Annahme, dass Start- und Zielpunkt bekannt sind und der Navigationspfad mit Hilfe des Kartenmaterials berechnet wurde. Der Pfad wird auf eine Linienrepräsentation vereinfacht und ausschließlich nur diese für den Algorithmus verwendet. Eine Korrektur kann nur anhand des Navigationspfades erfolgen, somit ist eine Abweichung von der Route nicht vorgesehen.

Ungenauigkeiten der IMU-Ortung in Bezug auf Erkennung der einzelnen Schritte und Schrittweite sowie der Bewegungsrichtung mittels Kompass, werden durch einen Abgleich mit dem zuvor berechneten Soll-Pfad (Map-Matching) korrigiert und verbessert (vgl. Abb. 2.13).

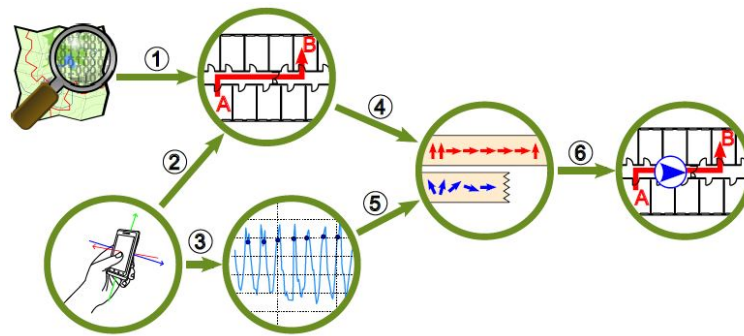


Abbildung 2.13.: Footpath Funktionsweise Wehrle (2011)

Die von der IMU gelieferte Trajektorie wird über den First Fit, bzw. Best Fit Algorithmus (vgl. Vazirani (2004)) mit der geplanten Route verglichen und die aktuelle Position entsprechend angepasst (siehe Abb. 2.14).

Der *First Fit* Algorithmus berechnet aus einer angenommenen Schrittweite die aktuelle Position auf dem Navigationspfad. Sobald die Bewegungsrichtung von der erwarteten Richtung abweicht - falls die angenommene Schrittweite zu kurz gewählt wurde - greift der „Lookahead Matching Mode“ und es wird entlang des Pfades nach einer entsprechenden Änderung der Bewegungsrichtung gesucht und die angenommene Position entsprechend dorthin verschoben. Sollte die Schrittweite zu lang geschätzt worden sein, wartet der Algorithmus an der nächsten Ecke, bis eine entsprechende Richtungsänderung festgestellt wurde Wehrle (2011).

Der *Best Fit* Algorithmus ist aus der Bioinformatik abgeleitet, welche ähnliche Verfahren für

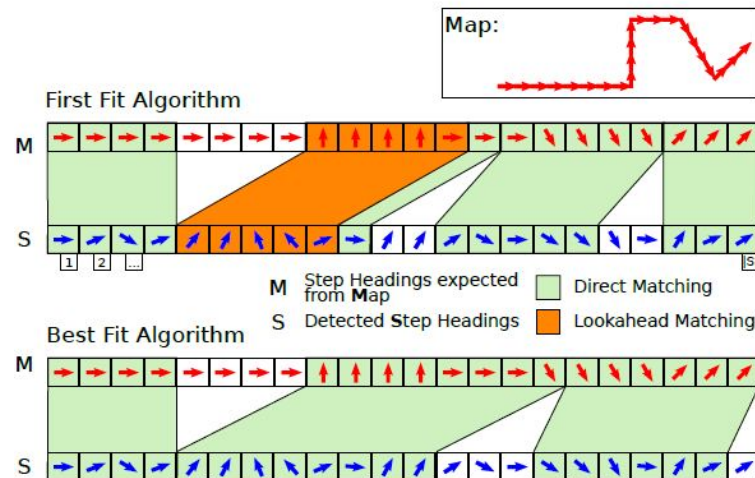


Abbildung 2.14.: Footpath Algorithmus Wehrle (2011)

den Abgleich von DNS-Sequenzen verwendet. Hierbei wird das Matching-Verfahren der dynamischen Programmierung entsprechend, iterativ der Minimierung eines Bestrafungswertes genähert. Somit kann die angenommene Position auf dem Pfad vor-, bzw. zurückspringen, je nachdem wo der momentane Scoringwert die besten Ergebnisse liefert Wehrle (2011).

Einen limitierenden Faktor dieser Realisierung stellen die verwendeten Sequenzalignment-Algorithmen dar, welche eine eindimensionale Sequenz in eine vorgegebene Folge einpassen. Dies setzt voraus, dass sich der zurückgelegte Pfad an einem zuvor geplanten Routenpfad entlang bewegt und nicht davon abweicht, da die aktuell angenommene Position nur entlang dieses Pfades korrigiert wird.

Weiterhin basiert der Algorithmus auf der Grundlage der IMU. Diese Sensoren lassen unter anderem eine direkte Auswertung von konkret erkennbaren Schritten und der Bewegungsrichtung zu. Abweichungen bezüglich der Schrittlänge und Richtung auf Grundlage des geplanten Navigationspfades werden durch den Algorithmus ausgeglichen.

Eine direkte Übertragung auf ein funkbasiertes (absolute) Ortungssystem mit zufälligen Fehlern ist nicht ohne Weiteres möglich, da sich die Funktionsweise des Korrektur-Algorithmus auf den systematischen Fehler des relativen Ortungssystems spezialisiert hat. Es gilt daher, Teilkomponenten und Ideen des Algorithmus auf Verwendbarkeit zu prüfen und gegebenenfalls Anpassungen vorzunehmen.

Autonome Einpassung von Trajektorien in ein Gebäudemodell

Bei relativen Ortungssystemen wird die aktuelle Position basierend auf einem definierten Ausgangspunkt angegeben (vgl. Kap. 2.2.3). Dies ist in aller Regel der Start der Navigation und wird durch eine manuelle Eingabe oder ein zweites absolutes Ortungsverfahren (vgl. Kap. 2.2.4) geliefert.

Um die Abhängigkeit von anderen Systemen zu minimieren, entwickelten Spassov u. a. (2007) ein Verfahren zur Einpassung der Bewegungstrajektorie in ein Gebäudemodell, um autonom auf die Startposition zu schließen und im Umkehrschluss die aktuelle Position ohne Hilfe von zusätzlichen Referenzsystemen zu erhalten.

Die Trajektorie entspricht der Bewegungshistorie, deren letztes Segment die aktuelle Position des Benutzers darstellt. Mögliche Bewegungspfade aus dem Gebäudemodell werden auf ein Knoten- und Kantenmodell vereinfacht. In diesem Graphen kann die Position bestimmt werden, indem die Kanten des Linienzuges mit ersterem abgeglichen werden. Ziel ist es im Graphen eine Reihe von Kanten zu finden, welche dem Linienzug am besten entsprechen.

Abhängig vom Gebäudegraphen können symmetrische Strukturen zu einer mehrdeutigen Lösung führen, wie in Abbildung 2.15 zu sehen. Zu einem bestimmten Zeitpunkt t kann der Linienzug an mehreren Stellen des Graphen eingepasst werden.

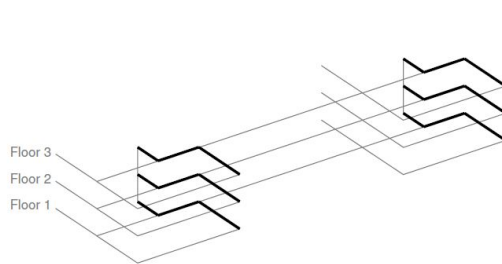


Abbildung 2.15.: Mehrdeutige Lösung Spassov u. a. (2007)

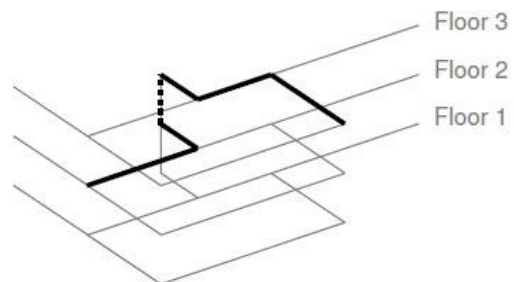


Abbildung 2.16.: Eindeutige Lösung Spassov u. a. (2007)

Erst mit einer längeren und eindeutigen Trajektorie kann die Aufenthaltswahrscheinlichkeit auf eine Kante konvergieren (siehe Abb. 2.16). Das bedeutet, dass die korrekte Position im Gebäude erst nach einiger Zeit bestimmt werden kann, bis ausreichend Informationen vorliegen.

Der Linienzug wird aus fehlerbehafteten Messwerten aufgebaut. Daher ist es unwahrscheinlich, einen perfekten Abgleich mit dem Graphen zu erzielen. Stattdessen wird ein „best match“-Verfahren mit einem statistischen Ansatz verfolgt.

Zu Beginn der Ortung ($t = 0$) ist die Aufenthaltswahrscheinlichkeit auf allen Kanten gleichmäßig verteilt. Dies entspricht der Annahme, dass die Person überall sein kann, da noch keine zu vergleichende Trajektorie vorliegt.

Sobald sich die Person bewegt und mittels der IMU eine Trajektorie aufgezeichnet wird, erfolgt die algorithmische Auswertung wie in [Abbildung 2.17](#) dargestellt.

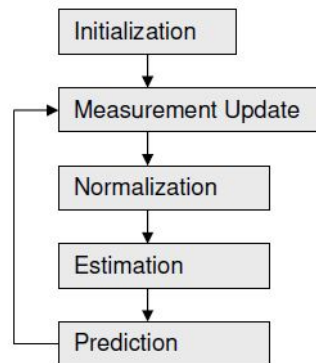


Abbildung 2.17.: Ablauf der Initialen Lokalisierung [Spasov u. a. \(2007\)](#)

Für jede Kante des Graphen wird die Wahrscheinlichkeit berechnet, mit der sich die Person darauf befindet. Nachdem in der Initialisierungsphase alle Wahrscheinlichkeiten gleichmäßig verteilt wurden, wird der Algorithmus so lange ausgeführt, bis eine eindeutige Position bestimmt wurde, welche als Ausgangspunkt für die zukünftige Ortung im Gebäudekontext verwendet wird.

Der Algorithmus wird bei jedem neuen Messwert ausgeführt und berechnet die neuen Wahrscheinlichkeiten, welche aus den gewichteten Kanten zum Zeitpunkt t , den Eingabedaten der IMU und der Historiendaten bestehen (vgl. [Abb. 2.18](#)). Eine Normalisierung über alle Kanten lässt in der Estimation-Phase eine Analyse bezüglich der wahrscheinlichsten Kante zu. In der nächsten Phase (prediction) werden die Gewichte der benachbarten Kanten des Graphen bewertet, welche im zukünftigen Schritt betreten werden könnten, um einen topologisch zusammenhängenden Trajektorienabgleich zu gewährleisten.

Eine Einschränkung dieser Realisierung stellt die Abstraktion des Gebäudemodells auf ein Knoten- und Kantenmodell dar. Das Matchen der Trajektorie erfolgt nur anhand dieser restriktiven Bewegungspfade. Die Leistungsfähigkeit hängt stark von der Beschaffenheit des Grundrisses ab. Des Weiteren werden die Wahrscheinlichkeiten nur bezüglich einer Kante angegeben. Unabhängig ihrer Länge wird die Person auf selbiger nicht näher definiert angenommen und stellt eine weitere Abstraktion dar.

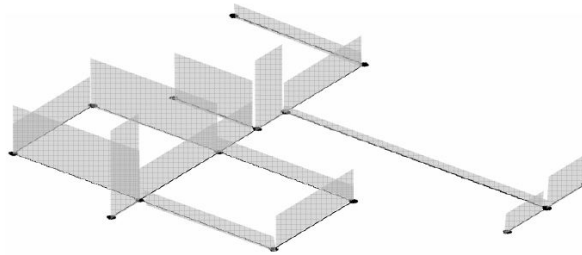


Abbildung 2.18.: Gewichtung der Kanten des Gebäudemodells [Spasov u. a. \(2007\)](#)

CADMS: Einpassung von Trajektorien in ein Gebäudemodell

Bei dem Computer Aided Disaster Management System (CADMS) Framework der [TU Graz \(2009\)](#) handelt es sich um eine Kombination aus Beschleunigungssensoren (IMU) und dem Abgleich mit Kartendaten (Map-Matching).

Ausgehend von einer bekannten Position, werden mit Hilfe der Sensormessdaten der IMU die Bewegungsrichtung und Geschwindigkeit berechnet. Dies geschieht durch doppelte numerische Integration der Beschleunigung um Entfernungswerte zu erhalten.

Die Sensoren haben eine endliche Genauigkeit und werden durch lokale Anomalien des Erdmagnetfeldes beeinflusst, wodurch es mit der Zeit zu erheblichen Abweichungen kommt, da die relativen Entfernungsangaben auch den Fehler aufsummieren. Hierbei handelt es sich um einen systematischen Fehler, welcher nicht durch einen numerischen Lösungsansatz korrigiert werden kann.

Um diesem Drift logisch entgegen zu wirken, werden Informationen aus dem Gebäudemodell zur Hilfe genommen, indem der von der IMU gelieferte Pfad in die umgebende Karte eingepasst wird. Das Ablaufdiagramm [2.19](#) beschreibt die generelle Funktionsweise des Algorithmus.

In einem ersten Schritt werden die rohen Sensormessdaten auf ihre physikalische Korrektheit überprüft (z. B. maximale Höchstgeschwindigkeit von Personen im Gebäude) und in Koordinaten überführt. Die darauf folgende automatisierte Plausibilitätsprüfung verwendet den Gebäudegrundriss und zusätzliche Informationen, wie z. B. Raumübergänge in einem standardisierten XML⁸ Format, welches einzelne Räume als 2D-Polygone darstellt, um illegale Raumwechsel zu erkennen (vgl. [Wießflecker u. a. \(2008\)](#) und [Bernoulli u. a. \(2008\)](#)). Hierunter wird verstanden, dass die fehlerbehaftete Bewegungstrajektorie Wände durchschneidet oder Türen und Treppenhäuser nicht exakt trifft. Räume können demnach nur über speziell markierte Übergangsbereiche betreten und verlassen werden. Dem Plausibilitätslevel entsprechend

⁸Extensible Markup Language

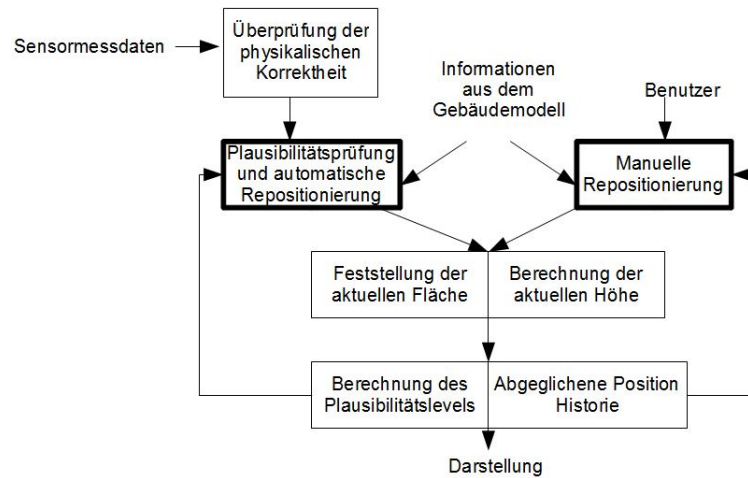


Abbildung 2.19.: CADMS Funktionsweise

wird dem Positionspunkt eine logische Fläche zugewiesen, um später einen logischen Bezug zur aktuellen Position im Gebäude zu erhalten **Wießflecker (2009)**. Sollte die automatisierte Repositionierung scheitern oder eine unzureichende Datengrundlage haben, besteht die Möglichkeit der manuellen Repositionierung mittels grafischer Benutzerinteraktion.

Wird ein Fehlerfall entdeckt, greifen die Korrekturmechanismen und die Trajektorie wird korrigiert. Dies kann auf zweierlei Weise geschehen:

Einerseits indem die vorhergehenden Daten und naheliegenden Türen auf einen Raumwechsel schließen lassen. Andererseits kann die Trajektorie auch entsprechend ihres Einfallwinkels bezüglich der Wand entlang korrigiert werden.

Im Fall der Türkorrektur wird das letzte Segment des Pfades abgetrennt und an dem Türschwerpunkt neu ausgerichtet (vgl. Abb. 2.20). Sollte in der Nähe einer Tür die Bewegungskurve eine Wand durchschneiden, wird der darauf folgende Messwert $2'$ entsprechend der lateralen Entfernung zum Türschwerpunkt, zur neuen Position 2 verschoben (vgl. Abb. 2.21), da er keinem physikalisch möglichen Positionspunkt entspricht **von Both und Koch (2009)**.

Die Korrektur an Wänden (z. B. entlang von Gängen) führt zu einer weiteren Verbesserung der Ortungsgenauigkeit und Einpassung der Bewegungstrajektorie. In Abbildung 2.22 wird die IMU Trajektorie nicht durch das CADMS-Framework korrigiert und verläuft aufgrund des Sensor driftes quer durch die Räume. Mit der Plausibilitätsprüfung (vgl. Abb. 2.23) wird das jeweils letzte Segment der Trajektorie korrigiert, sobald ein Wandabschnitt durchkreuzt wird.

Durchgeführte Testreihen der Arbeitsgruppe und statistische Auswertungen ergeben eine Ortungspräzision von ein bis zwei Metern für 95% der Bewegungspfade in einem Gebäude mit

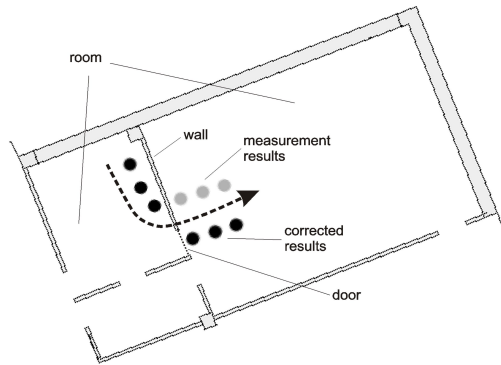


Abbildung 2.20.: CADMS Tür-Korrektur TU Graz (2009)

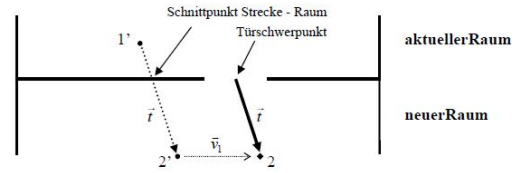


Abbildung 2.21.: Verschiebung der Trajektorie von Both und Koch (2009)



Abbildung 2.22.: IMU Pfad ohne Map-Matching TU Graz (2009)



Abbildung 2.23.: IMU Pfad mit Map-Matching TU Graz (2009)

ausreichend restriktivem Grundriss Walder (2011). Ein ähnliches IMU-basiertes Map-Matching Verfahren wird von Vildjiounaite u. a. (2002) vorgestellt.

IMU-basierte Partikelfilter

Ein Großteil der IMU-basierten Realisierungen zur Ortungsverbesserung verwenden Partikelfilter und werden in diesem Abschnitt gemeinsam vorgestellt. Partikelfilter implementieren rekursive Bayes-Filter indem sie sequenzielle Monte-Carlo Methoden nutzen. Sie eignen sich für nicht-lineare und nicht Gauß-verteilte Abschätzungsprobleme Daum (2005).

Die generelle Funktionsweise besteht in der Generierung einer Ansammlung (Wolke) von gewichteten Aufenthaltspunkten, welche durch die Wahrscheinlichkeitsbeschreibung des Ortungssystems begrenzt wird. Für den nächsten erkannten Bewegungsschritt wird für jedes Partikel aus dem vorherigen Schritt, mehrere Folgezustände berechnet und deren Wahrscheinlichkeit in die Bewertung einbezogen.

Eine entstehende Partikelwolke für den nächsten Schritt ist in Abbildung 2.24 dargestellt. x_{k-1}^i beschreibt den Ausgangspunkt (vorheriger Schritt) des i -ten Partikels, x_k^i beinhaltet i -Partikel des k -ten Schrittes. Neben der Unschärfe für die Schrittweite sorgt die Varianz der Bewegungsrichtung für eine Verbreiterung der Partikelwolke. In diesem Fall werden die Partikel noch nicht bewertet, da kein einschränkendes Gebäudemodell verwendet wird.

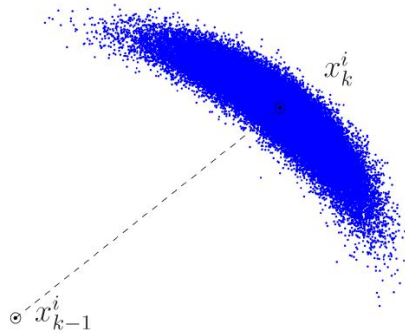


Abbildung 2.24.: Partikelwolke für den nächsten Schritt [Ascher u. a. \(2010\)](#)

Partikelfilter werden zur Abschätzung eines statistischen Systemzustandes verwendet und deren Wahrscheinlichkeitsdichte⁹ numerisch genähert. Im Fall des Map-Matching findet dies Anwendung, da die Dichtefunktion nicht Gauß-verteilt ist.

Für den nächsten Ortungspunkt werden auf Grundlage eines Bewegungsmodells neue Partikel erzeugt. Diese sind durch einen zufällig generierten Vektor definiert, welcher die PDF der Systemunschärfe abdeckt.

Nimmt man ein Gebäudemodell zur Bewertung der Partikel hinzu, lassen sich über die entstehende Dichteverteilung Aussagen über den wahrscheinlichen Aufenthaltsort treffen sowie Unsicherheiten über den Verlauf einer Trajektorie verringern. Partikel die eine Wand durchschreiten (magenta) und eine physikalisch unmögliche Position einnehmen, werden entsprechend niedrig bewertet (im einfachsten Falle mit einem binären Faktor) und ggf. verworfen (vgl. Abb. 2.25) [Widyawan u. a. \(2007\)](#).

Physikalisch wahrscheinliche Punkte bleiben erhalten und dienen als Grundlage für den nächsten Iterationsschritt nach Eintreffen neuer Sensordaten.

Sollte durch die Eliminierung ungültiger Partikel die Gesamtanzahl zu niedrig werden, erfolgt in der Resampling-Phase eine zufällige Duplikation der bestehenden Punkte.

In Testläufen von [Woodman und Harle \(2008\)](#) wird auf Basis handelsüblicher IMU-Systeme, mit Hilfe dieser Partikelfilter und einem detaillierten Gebäudeplan, dem typischerweise auf-

⁹Probability Density Function (PDF)

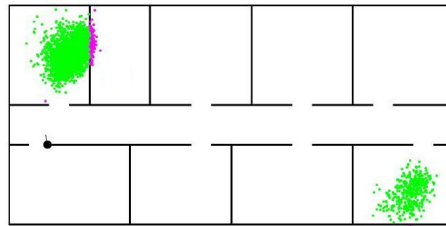


Abbildung 2.25.: Eliminierung ungültiger Partikel [Ascher u. a. \(2010\)](#)

trehenden Drift entgegen gewirkt. Ihre Realisierung ermöglicht es eine Person mit einer Genauigkeit von 0,73 m in 95% der Fälle, in einem 8725 m^2 großen Gebäude zu orten, was einer erheblichen Verbesserung der Ortungsgenauigkeit gegenüber einem unkorrigierten System entspricht.

Im Gegensatz zu der eigenen geplanten Realisierung verfolgen Partikelfilter einen anderen Ansatz, da sie eine Vielzahl von Partikeln in die Umgebung aussetzen und das Prinzip des *Survival of the fittest* verfolgen, wohingegen der eigene Ansatz eine gezielte Suche nach dem physikalisch logischen Pfad darstellt.

Das Map-Matching Verfahren mit Hilfe eines Partikelfilters ist nicht auf ein relatives Ortungssystem beschränkt, sondern lässt sich auch auf absolute Systeme anwenden, welche in Kapitel 2.2.4 näher betrachtet werden.

Zunächst folgt jedoch eine Weiterentwicklung des Partikelfilters auf Basis eines relativen Ortungssystems.

Backtracking Particle Filter

Der sogenannte „Backtracking Particle Filter“ (BPF) von [Beauregard u. a. \(2008\)](#) bietet eine Möglichkeit zur Ortungsverbesserung auf Grundlage einer Weiterentwicklung des Partikelfilters.

Ein gewöhnlicher Partikelfilter berechnet nur auf Grundlage der aktuellen Partikelwolke und des Gebäudemodells die nächste Fläche der möglichen Aufenthaltspositionen (vgl. Abb. 2.26).

Sollte sich ein Cluster in eine Sackgasse bewegt haben, die Bewegungstrajektorie jedoch weiter durch die Wand verlaufen, werden alle Partikel dieses Clusters ungültig und die Bewegung ist nicht mehr nachvollziehbar.

Das Szenario wird in Abbildung 2.27 a) anhand von vier PDF-Zeitschritten veranschaulicht. „Ground Truth“ entspricht der tatsächlich zurückgelegten Trajektorie. Die Position wird anhand der größten Aufenthaltswahrscheinlichkeit durch den Partikelfilter bestimmt. Parallel dazu wird geprüft, ob die neue Position über den bisher zurückgelegten Pfad erreichbar ist.

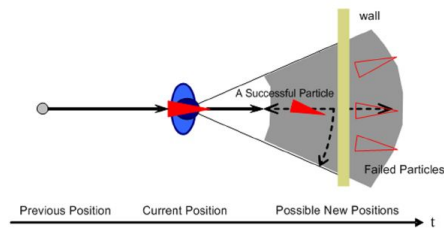


Abbildung 2.26.: Niedrige Partikelbewertung nach Wanddurchschreitung [Widyawan u. a. \(2008b\)](#)

Sollte dies nicht der Fall sein (vgl. b), wird der Pfad zurückverfolgt, die nun ungültigen Partikel gelöscht und eine alternative Zustandsabschätzung entsprechend der verbleibenden Partikel berechnet, welche einen Pfad zum neuen Ortungspunkt bilden. Der Bewegungspfad wird somit rückwirkend korrigiert (vgl. c).

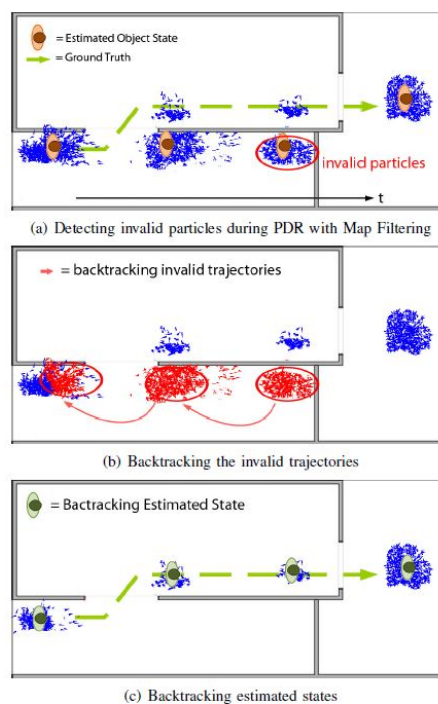


Abbildung 2.27.: Backtracking Particle Filter [Beauregard u. a. \(2008\)](#)

Der BPF verhält sich zunächst wie der in Kapitel 2.2.3 beschriebene reguläre Partikelfilter, erweitert diesen jedoch in der Bewertungsphase um zusätzliche Schritte, wodurch eine verbesserte Zustandsabschätzung möglich werden soll. Es werden alle berechneten Partikel erhalten,

eine Bewertung findet zwar statt, führt jedoch nicht zu ihrer Löschung, wie es in Kap. 2.2.3 vorgeschlagen wird. Dies lässt eine spätere Neubewertung der verbleibenden Partikel nach einem Backtracking-Fall zu, was zu einer alternativen Routenentscheidung führen kann (vgl. Abb. 2.27 b) und c)).

Zusätzlich wird eine Historie eines jeden Partikels und somit der zurückgelegten Trajektorie gespeichert, um die Rückverfolgung zu einem früheren Bewegungsschritt zu ermöglichen.

Im Falle eines Backtracking-Ereignisses werden diejenigen Partikel verworfen, die aufgrund ihrer Dichtefunktion in Kombination mit dem Map-Matching die vorherige Position definiert haben. Ausgehend von den verbleibenden Partikeln findet eine erneute Bewertung statt und es wird geprüft ob diese Partikelwolke durch das Map-Matching nicht ausgelöscht wird und zum Ziel führt. Dies basiert auf der Annahme, dass die ungültigen Partikel auf Grundlage von Partikeln erzeugt wurden, welche einer ungültigen Trajektorie folgen und die Neuberechnung der PDF ohne jene Partikel zu einer verbesserten Zustandsschätzung führen [Widyawan u. a. \(2008b\)](#). Die Rückverfolgung findet solange statt, bis entweder das Ziel oder die maximal zulässige Anzahl an Historiensritten erreicht ist, und der Algorithmus endet.

Der Algorithmus basiert auf dem Partikelfilter von [Widyawan u. a. \(2007\)](#), welcher später im Kapitel der Absoluten Verfahren (2.2.4) separat behandelt wird. Der BPF ist spezialisiert auf die Verwendung in Brandbekämpfungs- und Rettungsszenarien, um eine robuste und präzise Ortung innerhalb von Gebäuden zu gewährleisten. Die Anforderungen verlangen eine möglichst autonome Ortung ohne zeitaufwändige Kalibrierung sowie eine Unterstützung unterschiedlicher Detailgrade¹⁰ des assistierenden Gebäudemodells, welches u. U. erst vor Ort anhand der Rettungspläne erstellt wird.

Messreihen von [Widyawan u. a. \(2008a\)](#) und [Beauregard \(2009\)](#) haben eine Ortungsgenauigkeit von 1,32 m mittlerer 2D-Abweichung auf Basis eines detaillierten Gebäudeplans ergeben. Der Algorithmus leistet auf Grundlage von reduzierten Informationen eines Gebäudemodells (nur äußere Wandinformationen) eine Präzision von 1,89 m mittlerer 2D-Abweichung. Dies stellt eine erhebliche Ortungsverbesserung gegenüber den unkorrigierten IMU-Daten ohne Map-Matching oder Backtracking von 8,04 m mittlerer 2D-Abweichung dar. Gegenüber einem regulären Partikelfilter (PF) verbessern sich die Messergebnisse nicht erheblich (BPF: 1,32 m gegenüber IMU+PF: 1,55 m). Dies liegt darin begründet, dass das Backtracking nur unter speziellen Umständen zum Tragen kommt und die Präzision hauptsächlich auf dem Partikelfilter basiert.

¹⁰Level of Detail (LoD)

Bestimmung der Aufenthaltswahrscheinlichkeit auf Basis von Bewegungsmodellen

Entgegen anderer Partikelfilter, welche Wände zur Gewichtung der Partikel heranziehen (Beauregard u. a. (2008), Woodman und Harle (2008), Ascher u. a. (2010), siehe Kap. 2.2.3), wird bei diesem Pedestrian Dead Reckoning (PDR) Verfahren eine Wahrscheinlichkeitsfunktion zur Ortungsverbesserung verwendet, welche auf dem menschlichen Bewegungsmodell beruht Kaiser u. a. (2011a). Der für diese Realisierung verwendete Partikelfilter stammt von Krach und Robertson (2008), wobei die Bewertung der Partikel anhand des „Angular PDF“ Algorithmus stattfindet, was die Ortungsgenauigkeit zusätzlich verbessern soll.

Wie Beauregard (2009) in Kapitel 2.2.3 festgestellt hat, stoßen reguläre Partikelfilter in speziellen Szenarien (komplette Eliminierung aller Partikel in Sackgassen) an ihre Grenzen. Ein weiteres Szenario besteht in der Multimodalität, in welche sich die Partikelwolke aufteilt und unterschiedliche Pfade verfolgt. So lange das Gebäudemodell gleichmäßig dicht aufgebaut ist und ausreichend einschränkende Wände besitzt, kann das abdriftende Cluster eliminiert werden.

Sollte jedoch die zweite Partikelwolke (blauer Pfeil) in einen Gebäudeteil (B) ohne begrenzende Hindernisse gelangen (vgl. Abb. 2.28), so werden die Partikel nicht mehr entfernt, und zusätzlich die eigentlich korrekte Partikelwolke (entlang der grünen Trajektorie) in Gebäudeteil A aufgrund der Einschränkungen schneller eliminiert.

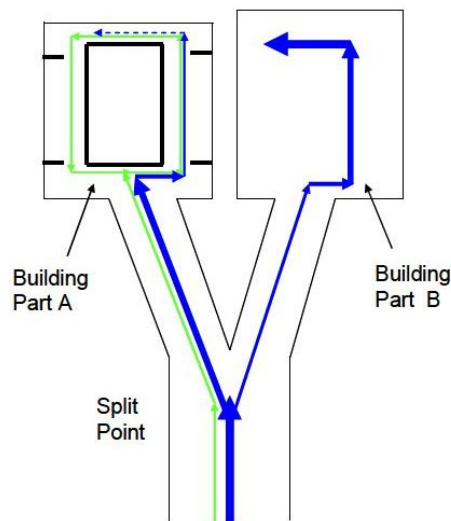


Abbildung 2.28.: Entstehung einer multimodalen Ortung Kaiser u. a. (2011a)

Der Lösungsvorschlag von Kaiser u. a. (2011a) verwendet eine winkelabhängige PDF (kartenbasierter Diffusions-Algorithmus nach Khider u. a. (2008) sowie Kaiser u. a. (2011b)) zur

Bewertung der Partikel anstatt eine binäre Gewichtung anhand von Wänden zu nutzen. Mit Hilfe des Diffusions-Algorithmus wird ein kontinuierlich ausströmendes Gas simuliert, welches sich im Raum frei ausbreitet und von den Wänden absorbiert wird (vgl. Abb. 2.29 a). Die Quelle stellt die aktuelle Ortungsposition dar. Basierend auf der Gasverteilung wird die winkelabhängige PDF berechnet, indem ein Grenzwert der Gasdichte eine Konturlinie bildet (vgl. Abb. 2.29 b). Sollte eine Wand die Ausbreitung vorzeitig verhindern, so verläuft die Konturlinie an der Wand entlang. Unter der Annahme, dass sich die mögliche Bewegungsrichtung anhand der Dichteverhältnisse orientiert, lässt sich die PDF direkt aus der Distanz zwischen dem Gasursprung und der Konturlinie bestimmen.

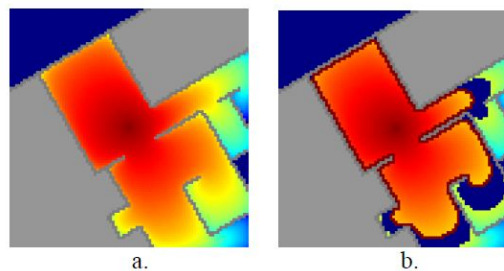


Abbildung 2.29.: a) Diffusionsmatrix b) Konturlinie (dunkelrot) Kaiser u. a. (2011a)

Die Schrittweite fließt nicht in die Gewichtung ein, sondern nur die Distanz zur Konturlinie unter dem relativen Winkel α zwischen dem vorherigen und aktuellen Ortungspunkt. Sollte eine Wand zwischen dem Ursprung und der Konturlinie liegen, wird dieser Wert nicht für eine Bewertung der PDF verwendet, da dies nicht dem natürlichen Bewegungsmodell entspricht. Zusätzlich wird die bisherige Bewegungsrichtung berücksichtigt, indem nur Winkel zwischen $\alpha - 90^\circ$ und $\alpha + 90^\circ$ betrachtet werden.

Die größte Entfernung zur Konturlinie stellt somit die wahrscheinliche Bewegungsrichtung dar und fließt in die Bewertung der Partikel ein.

Messreihen von Kaiser u. a. (2011a) haben einen durchschnittlichen Positionsfehler von 2,85 m für ihren Partikelfilter mit winkelabhängiger PDF ergeben, wohingegen ein normaler PF (ohne dem angular PDF) auf einen mittleren Positionsfehler von 8,59 m kommt. Das oben dargestellte Multimodalitätsproblem kann durch den Algorithmus verhindert werden und verdeutlicht, dass ein Gebäudeplan mehr leisten kann als nur für eine binäre Bewertung genutzt zu werden, indem ein Bewegungsmodell in die Bewertung der Partikel mit einfließt.

Bubbleband: Pfadglättung zwischen IMU und Partikelfilter

Ein generelles Problem das durch die Pfadkorrektur entsteht, ist die resultierende unnatürliche Trajektorie, welche starke Krümmungen oder sogar Sprünge aufweisen kann. Dieses Verhalten ist bei obigen erwähnten CADMS (Kap. 2.2.3) in Abbildung 2.23 deutlich zu erkennen. Die resultierende Trajektorie liegt zwar innerhalb von Gängen und wechselt die Räume über erlaubte Übergänge, entspricht von ihrem Aussehen her aber nicht einer natürlichen Bewegung eines Menschen durch ein Gebäude (wie es bei dem von der IMU aufgezeichneten Pfad der Fall ist).

Es wird demnach ein Verfahren gesucht, das einen Kompromiss zwischen der natürlichen Bewegung (basierend auf den IMU-Daten) und der logischen Raumkorrektur (in diesem Fall einen Partikelfilter) trifft.

Für die Lösung der oben genannten Problemstellung wird zunächst ein Ortungssystem samt Kartenabgleich benötigt. Das Map-Matching erfolgt bei dieser Realisierung durch einen Partikelfilter, welcher auf einem dualen inertialen Ortungssystem basiert [Sternberg \(2010\)](#). Der hier erwähnte Partikelfilter ist vergleichbar mit den in Kapitel 2.2.3 vorgestellten System und wird hier nicht näher thematisiert. Um die resultierende Trajektorie des Partikelfilters zu glätten, verwenden [Ascher u. a. \(2010\)](#) einen „balanced bubble band smoother“, der einen optimalen Abgleich zwischen Karte und aufgezeichneten IMU-Daten erlaubt (vgl. Abb. 2.30).

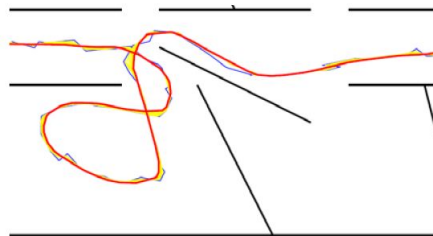


Abbildung 2.30.: Bubbleband Algorithmus [Ascher u. a. \(2010\)](#)

Der Glättungseffekt des Algorithmus basiert auf den unterschiedlichen Kräften, welche auf die einzelnen Stützstellen der IMU-Messdaten wirken und diese verschieben. Die Positionsschätzungen der einzelnen Schritte des Partikelfilters werden als Kugeln betrachtet, die untereinander mit Federn verbunden sind. Es wirken drei Kräfte, bestehend aus der Anziehung, der Abstoßung und dem Erhalt des relativen Winkels zueinander.

$$\vec{b}_{new} = \vec{b}_{old} + \alpha * \vec{f}_{1,2,3} \quad (2.8)$$

Summiert man diese Kräfte für die Kugel und multipliziert diese mit einem Skalierungsfaktor (vgl. Gleichung 2.8), so erhält man eine neue Positionsschätzung, welche die benachbarten Kugeln mit einbezieht und für eine Glättung der Trajektorie sorgt [Ascher u. a. \(2010\)](#).

Die zusammenziehende Kraft (contraction) \vec{f}_c spiegelt die Anziehung zwischen dem aktuellen Punkt b_i , seinem Vorgänger b_{i-1} und Nachfolger b_{i+1} wider (Gleichung 2.9). k_c stellt einen Gewichtungsfaktor dar, um den Einfluss der zusammenziehenden Kraft für ein optimales Korrekturverhalten zu variieren.

$$\vec{f}_c = k_c * \left(\frac{b_{i-1} - b_i}{\|b_{i-1} - b_i\|} + \frac{b_{i+1} - b_i}{\|b_{i+1} - b_i\|} \right) \quad (2.9)$$

Die abstoßende Kraft (rejection) \vec{f}_r (2.10) stellt das Streben dar, sich von Wänden fern zu halten. Dies soll verhindern, dass sich die Punkte aufgrund der anderen beiden Kräfte in Wände oder darüber hinaus verschieben.

Nur innerhalb eines Abstandes ρ_0 zu einer Wand soll diese Kraft wirken und ansonsten 0 werden. k_r ist ähnlich k_c ein Gewichtungsfaktor für die abstoßende Kraft, jedoch mit negativem Vorzeichen. Die Kraft wird zusätzlich radial zu ρ größer.

$$\vec{f}_r = \begin{cases} k_r(\rho_0 - \rho) \frac{\partial \rho}{\partial \vec{b}} & \rho < \rho_0 \\ 0 & \rho \geq \rho_0 \end{cases} \quad (2.10)$$

Die Kräfte \vec{f}_c und \vec{f}_r würden kreisförmige Trajektorien (im freien Raum) zu einer Geraden zusammen ziehen. Daher wird die dritte Kraft \vec{f}_I benötigt, welche dafür sorgt, dass die ursprüngliche Orientierung der IMU-Trajektorie, abweichenden neuen Winkeln entgegen wirkt. Gleichung 2.11 stellt eine Kraft in Richtung ψ dar, wobei die Gewichtung durch k_I geregelt wird.

$$\vec{f}_I = k_I \begin{bmatrix} \cos(\psi) \\ \sin(\psi) \end{bmatrix} \quad (2.11)$$

Die Verwendung von Korrekturverfahren wie beispielsweise CADMS (vgl. Kap. 2.2.3) oder Partikelfiltern (vgl. Kap. 2.2.3) führen zu einer veränderten Trajektorie, die unnatürliche Charakteristika besitzt. Durch die Glättung des Bubbleband Algorithmus wird der zurückgelegte Pfad logisch nachvollziehbarer, da eine Person keine sprunghaften Bewegungen macht, sondern eine gewisse Ähnlichkeit bezüglich der vorhergegangenen Bewegungsphase besteht.

2.2.4. Absolute Ortungsverfahren

In folgendem Abschnitt werden absolute Ortungsverfahren vorgestellt. Hierunter fallen funkbasierte Systeme, welche die Position über Multilateration oder Triangulation bestimmen, z. B. WLAN-Fingerprinting oder Ultra Wide Band (UWB) wie [Ubisense \(2013\)](#). Ihre Eigenschaft besteht darin, dass die Ortungspunkte unabhängig von den jeweiligen Vorgängern sind. *Absolut* bedeutet in diesem Zusammenhang *nicht*, dass die Position exakt sein muss und somit einer gewissen Unschärfe unterliegen kann.

Im Outdoorbereich kann mit Hilfe von GSM¹¹-Funkzellen und Map-Matching mit hoher Wahrscheinlichkeit eine straßengenaue Ortung realisiert werden ([Thiagarajan u. a. \(2011\)](#)). Für die Indoornavigation sind Präzisionen im Submeter-Bereich nötig. Da viele absolute Ortungssysteme, ähnlich den relativen Verfahren (vgl. Kap. 2.2.3), auf nur einem System beruhen, werden diese Zielvorgabe nicht erreicht und ebenfalls hybride Verfahren benötigt.

Zunächst wird in Kapitel 2.2.4 ein hybrides Ortungsverfahren vorgestellt, welches eine Kombination aus absoluten und relativen Ortungssystemen nutzt und einen Übergang zum vorherigen Kapitel darstellt. Im Anschluss wird in Kapitel 2.2.4 ein WLAN-basiertes Verfahren beschrieben, das ähnlich zu Kapitel 2.2.3 Map-Matching und Partikelfilter zur Präzisionssteigerung verwendet. Den Abschluss des Kapitels bildet das MagicMap Framework (vgl. Kap. 2.2.4), welches eine Vielzahl von Ortungssystemen unterstützt, dessen hauptsächlicher Fokus jedoch auf der WLAN-Ortung auf Basis von Fingerprinting- sowie Propagation-Verfahren liegt.

Die dargestellten hybriden Verfahren zeigen, dass eine Präzisionssteigerung für absolute Ortungssysteme möglich ist und bilden die Grundlage für eigene Entwicklungsansätze sowie Referenzwerte für eine vergleichende Betrachtung.

Hybrider WLAN- und IMU-Partikelfilter

Das Framework zur Indoor-Ortungsverbesserung von [Wang u. a. \(2007\)](#) basiert auf der Kombination von IMU-Sensoren, WLAN-Signalstärken und Karteninformationen durch einen Partikelfilter und stellt einen hybriden Übergang von den relativen (siehe Kapitel 2.2.3) zu den absoluten Ortungssystemen dar.

Die grundlegende Funktionsweise ist in Abbildung 2.31 ersichtlich. Die WLAN-Ortung erfolgt über den Signalstärkenabgleich (RSS) mit zuvor aufgenommenen Referenzwerten

¹¹Global System for Mobile Communications (früher Groupe Spécial Mobile)

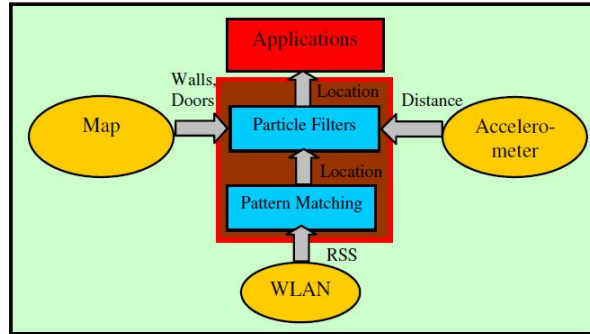


Abbildung 2.31.: Systemarchitektur Wang u. a. (2007)

(Fingerprinting bzw. Pattern Matching) durch den K-Nearest-Neighbor (KNN) Algorithmus aus Gleichung 2.12 (vgl. Bahl und Padmanabhan (2000)).

$$d(x) = \frac{1}{Q} \sum_{q=1}^Q \{(P_q^m - P_q(x))^2\} \quad (2.12)$$

Gesucht ist die Entfernung $d(x)$ zum Kalibrierungspunkt. P_q^m ist die gemessene Signalstärke des Access Points q , $P_q(x)$ ist das q^{te} Element aus der Fingerprinting Tabelle von Punkt x und Q ist die Gesamtanzahl der APs.

Mit Hilfe der Beschleunigungssensoren werden einzelne Schritte und weiterführend die Schrittweite erkannt. Der Partikelfilter kombiniert diese Werte und gewichtet die Partikel zusätzlich durch einen Kartenabgleich (Map-Matching).

Der RSS Partikelfilter nutzt die Positionsbestimmungen aus Gleichung 2.12. Für jedes Partikel $\mathbf{x}^i(k) = [x^i(k) \ y^i(k) \ v_x^i(k) \ v_y^i(k)]^T$ wird ein Nachfolger $\mathbf{x}^i(k+1)$ bestimmt, wobei angenommen wird, dass Personen eine Trägheit besitzen und durch Gauß-verteilte Beschleunigungen beeinflusst werden.

Die Gewichtung $w^i(k)$ der Partikel, des IMU-PFs (vgl. Kapitel 2.2.3) sowie RSS-PFs, erfolgt nach Gleichung 2.13. Es wird davon ausgegangen, dass die über RSS bestimmte Position Gauß-verteilt um die tatsächliche Position liegt.

$$w^i(k) = \begin{cases} 0 & \text{if new particle crosses walls} \\ \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x^i(k)-x_{RSS(k)})^2+(y^i(k)-y_{RSS(k)})^2}{2\sigma^2}} & \text{otherwise} \end{cases} \quad (2.13)$$

Zunächst erfolgt der Abgleich mit den Karteninformationen, indem die Gewichtung 0 gesetzt wird, sobald ein Partikel eine Wand durchschreitet, was einer Löschung des Partikels entspricht. Anhand der Standardabweichung σ werden jeweils die x- und y-Komponenten der unterschiedlichen Positionsschätzungen $x^i(k)$ der IMU sowie $x_{RSS}(k)$ des RSSI-Fingerprintings kombiniert.

Wang u. a. (2007) haben unterschiedliche Algorithmen gegenüber gestellt und Messungen durchgeführt. Als Vergleich wurde eine unkorrigierte Positionsschätzung auf Grundlage der RSSI-Werte und einem KNN-Algorithmus aufgenommen. Durch einen Kalman Filter Ristic u. a. (2004) konnte eine Verbesserung von 10% erreicht werden. Eine weitere Verbesserung von 25% konnte durch den oben beschriebenen Partikelfilter erzielt werden, welcher RSSI, IMU und Map-Matching kombiniert.

Die Autoren stellen in Aussicht, dass nicht nur WLAN-Signalstärken für den Algorithmus in Frage kommen können, sondern auch andere netzwerkbasierte Systeme wie UWB oder GSM, bzw. TDOA¹², TOA¹³ oder AOA¹⁴ Verfahren genutzt werden können.

RSSI basierte Sensorfusionierung mittels Partikelfilter

Ähnlich wie im vorherigen Kapitel 2.2.4 wird in der Realisierung von Widyawan u. a. (2007) ein Partikelfilter auf Basis von funkbasierten Signalstärken (RSSI) und Map-Matching verwendet. In diesem Fall wird jedoch keine IMU verwendet, um die Hardwareanforderung möglichst gering zu halten und eine reine Softwarelösung zu bieten.

Der hier vorgestellte Algorithmus wird in abgewandelter Form (zur Verarbeitung relativer Ortungsinformationen) in Kapitel 2.2.3 von Widyawan u. a. (2008b) verwendet. Wie das vorherige Kapitel 2.2.4 gezeigt hat, ist es möglich mit Hilfe von Partikelfiltern auch absolute Ortungssysteme anhand zusätzlicher Karteninformationen zu verbessern. In diesem Abschnitt soll der Fokus speziell auf dem RSSI-basierten PF liegen, dessen Grundlagen auf Bayes Algorithmen und sequentiellen Monte-Carlo Methoden beruhen (siehe Chen (2003) sowie Oh u. a. (2004)).

In Abbildung 2.32 sind die Schnittstellen der unterschiedlichen Eingabemöglichkeiten veranschaulicht. Die RSSI Auswertung kann auf Grundlage unterschiedlicher funkbasierter Systeme erfolgen, z. B. WLAN oder WSN¹⁵. Letzteres wird zunächst kurz durch einen RSSI Filter geglättet und gepuffert, um unterschiedliche Messfrequenzen auszugleichen und zu synchronisieren. Der Abgleich der WLAN- und WSN-Signalstärken mit den hinterlegten Fingerprintingdaten

¹²Time Difference of Arrival

¹³Time of Arrival

¹⁴Angle of Arrival

¹⁵Wireless Sensor Network

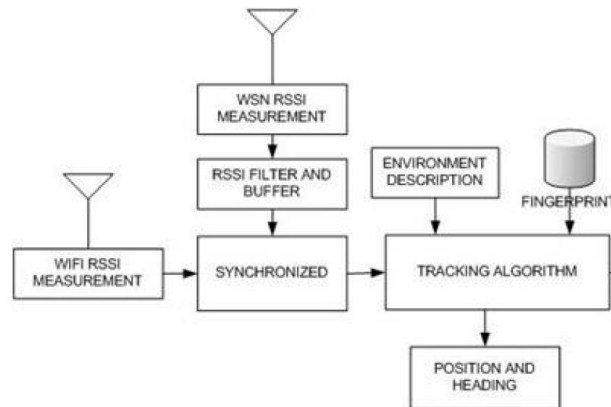


Abbildung 2.32.: Sensorfusionierung mittels Partikelfilter Widyawan u. a. (2007)

erfolgt mittels eines Tupels bestehend aus MAC¹⁶-Adresse des APs und dem RSSI Messwert. Die Fusionierung erfolgt durch den Partikelfilter, welcher zur Gewichtung der Partikel einen Gebäudeplan nutzt.

Zu Beginn der Ortung ($t = 0$) findet eine Initialisierung statt (vgl. Abb. 2.33). Es werden N Partikel ausgehend der initialen Dichte erzeugt und gleiche Gewichte w_0^i verteilt (vgl. Gleichung 2.14).

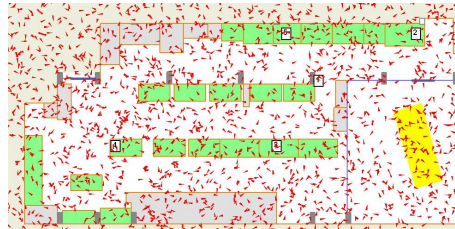


Abbildung 2.33.: Initialisierung der Partikelwolke zum Zeitpunkt $t = 0$ Widyawan u. a. (2007)

$$\{x_0^i\}_{i=1}^N = p_{x0} , w_0^i = \frac{1}{N} \quad (2.14)$$

In der Prediction-Phase wird für jedes Partikel x_t^i eine neue Position entsprechend des Bewegungsmodells $f()$ und eines zufälligen Rauschens n bestimmt (siehe Gleichung 2.15).

$$x_t^i = f(x_{t-1}^i) + n_{t-1}^i , i = 1, \dots, N \quad (2.15)$$

¹⁶Media Access Control

2. Vergleichbare Arbeiten

Die Update-Phase bestimmt die neue Gewichtung w_t^i der Partikel nach der Wahrscheinlichkeitsfunktion $p(z_t|x_t^i)$ entsprechend Gleichung 2.16 und normalisiert sie über die Gewichtung aller Partikel.

$$w_t^i = w_{t-1}^i p(z_t|x_t^i) \quad (2.16)$$

Während der Resampling-Phase werden entsprechend viele Partikel dupliziert um eine konstante Anzahl Partikel N zu erhalten. Die neue Zustandsabschätzung x_t erfolgt durch Mittelung über alle Partikel x_t^i (siehe Gleichung 2.17).

$$x_t = \frac{1}{N} \sum_{i=1}^N x_t^i \quad (2.17)$$

Das oben bereits angesprochene Bewegungsmodell beschränkt die Partikelausbreitung während der Prediction-Phase. Die Partikelbewegung kann durch Gleichung 2.18 beschrieben werden, indem aus der Geschwindigkeit v_t , der Orientierung α , einer Störgröße n und der vorherigen Position auf x_t^i geschlossen wird.

$$x_t^i = \begin{pmatrix} x_{t-1}^i + v_t \Delta t \cos(\alpha_t) + n_{t-1} \\ y_{t-1}^i + v_t \Delta t \sin(\alpha_t) + n_{t-1} \end{pmatrix} \quad (2.18)$$

Durch einen heuristischen Ansatz wird die Geschwindigkeit als $v = [0, 10 \frac{m}{s}]$ mit einer Streuung von $v_t = |N(v_{t-1}, 1 \frac{m}{s^2} \Delta t)|$ bezüglich dem Vorgänger angenommen. Für die Orientierung gilt $\alpha = [0, 2\pi]$, wobei die maximale Winkeländerung zum Vorgänger $\alpha_t = N(\alpha_{t-1}, 2\pi - \arctan(\frac{\sqrt{v_t}}{2}) \Delta t)$ beträgt.

Abbildung 2.34 zeigt die Update-Phase der Partikel anhand der Wahrscheinlichkeitsfunktion (blaue Kreise) und die darauf folgende Zustandsabschätzung zum Zeitpunkt $t = 1s$, $t = 20s$ und $t = 30s$.



Abbildung 2.34.: Zustandsabschätzung zum Zeitpunkt $t = 1s$, $t = 20s$ und $t = 30s$ (v.l.n.r)
Widyawan u. a. (2007)

2. Vergleichbare Arbeiten

Das Map-Matching bzw. Map-Filtering verhält sich wie in Kapitel 2.2.3 bereits beschrieben und wird nicht näher thematisiert.

Mit Hilfe des Partikelfilters konnten [Widyawan u. a. \(2007\)](#) auf Basis von WLAN-Signalstärken eine Standardabweichung von $\sigma = 1,37m$ und mittleren Fehler von $\mu = 2,24m$ erzielen. Durch die Kombination mit einem WSN wurde die Präzision zusätzlich gesteigert ($\sigma = 0,53m$ und $\mu = 0,97m$). Der Tradeoff besteht jedoch in dem hohen Installationsaufwand, der für ein zusätzliches WSN benötigt wird.

MagicMap

Das MagicMap Projekt der [Humboldt-Universität zu Berlin \(a\)](#), unter Leitung von Peter Ibach, beschäftigt sich ebenfalls mit der Positionsbestimmung sowie dessen Präzisionssteigerung. Zusätzlich zur Ortung per WLAN können andere funkbasierte Systeme integriert werden (ZigBee, Bluetooth, GPS, u.a. - siehe Abb. 2.35). Es existiert auch eine Schnittstelle um IMU-Daten auszuwerten [Humboldt-Universität zu Berlin \(b\)](#).

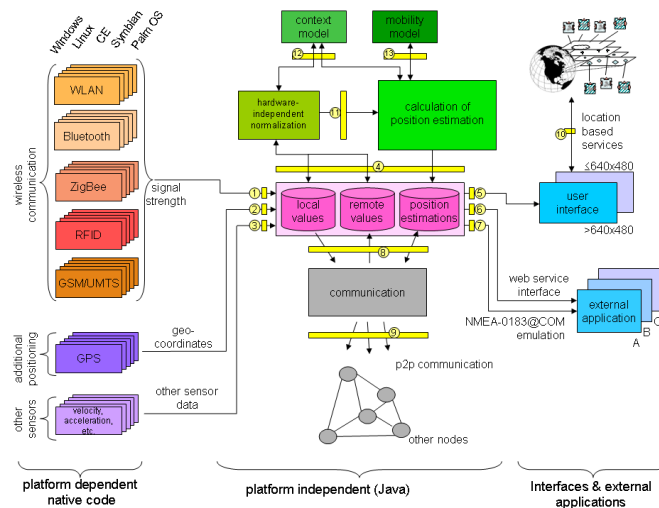


Abbildung 2.35.: MagicMap Schnittstellen [Humboldt-Universität zu Berlin \(b\)](#)

Das Verfahren ermöglicht ein hybrides Ortungskonzept, indem es mehrere Quellen für die Signalstärkenauswertung verwendet. Die gemittelte und normierte Signalstärke $\overline{RSSI}_{i,j}$ lässt Rückschlüsse auf die Distanz $\hat{\delta}_{i,j}$ zwischen Sender j und Empfänger i zu (vgl. Gleichung 2.19).

$\max \text{RSSI}_i$ beschreibt den höchstmöglichen RSSI-Wert und dient der Kalibrierung, während P_j die Sendeleistung widerspiegelt.

$$\hat{\delta}_{i,j} = \frac{\max \text{RSSI}_i - \overline{\text{RSSI}}_{i,j}}{P_j} \quad (2.19)$$

Auf Grundlage von mehreren Distanzschätzungen und bekannten Standorten der Sender (rote Punkte = APs) kann mittels Multilateration auf die aktuelle Position (grün) geschlossen werden (vgl. Abb. 2.36). Hierbei wird versucht, die Summe der quadrierten Abweichungen der Kantenlängen entsprechend $\sum(\tilde{\delta}_{i,j} - \hat{\delta}_{i,j})^2$ zu minimieren.

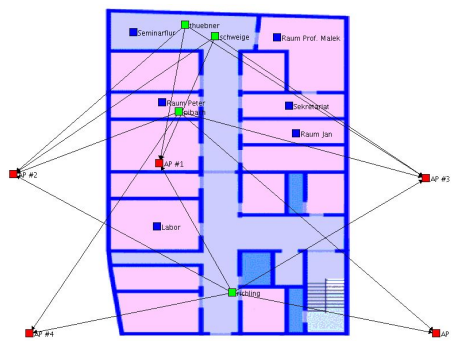


Abbildung 2.36.: MagicMap Multilateration **Ibach u. a. (2004)**

Die interessanteste Entwicklung von MagicMap ist jedoch die optionale Verwendung sogenannter „Plugins“ **Humboldt-Universität zu Berlin (d)**, welche die Ortungsgenauigkeit weiter verbessern sollen. Das Umgebungsmodell **Humboldt-Universität zu Berlin (e)** und die pfadbasierte Ortung **Humboldt-Universität zu Berlin (c)** verwenden Umgebungsinformationen, bestehend aus Wänden und Übergängen sowie Bewegungsinformationen um die Güte der Positionsdaten zu bewerten.

Hierdurch wird von einem punktuellen Ortungsverfahren auf eine pfadbasierte Ortung (Historie der Messdaten) gewechselt. Nach dem Prinzip des *Maximum Likelihood* Ansatzes wird versucht aus den Distanzschätzungen den am wahrscheinlichsten zurückgelegten Pfad zu bestimmen, indem ein genetisches Suchverfahren angewendet wird (Map-Matching).

Die Initialisierung erzeugt zunächst einen Pfad. Durch Mutation und Rekombination wird die Pfadpopulation variiert, Hypothesen generiert und anschließend evaluiert, indem die theoretischen und berechneten Distanzen ausgehend von ihren Varianzen überprüft werden. Basierend darauf wird eine neue Generation erzeugt.

Jeder neu eintreffende Messwert wird in die Berechnung mit einbezogen und kann zur Verwerfung und erneuten Initialisierung des Pfades führen **Humboldt-Universität zu Berlin (c)**.

Der zugrunde liegende Algorithmus ist nicht näher beschrieben (Closed Source), scheint jedoch auf einem Partikelfilter zu basieren.

Durch die Verwendung des Bewegungsmodells kann die pfadbasierte Ortung zusätzlich verbessert werden, da das natürliche Bewegungsverhalten von Menschen zur Bewertung der nächsten Positionsschätzung herangezogen wird. Abbildung 2.37 beschreibt die Fläche, die für einen nächsten Ortungspunkt auf Grundlage der bisherigen Bewegung in Frage kommt (von rot nach gelb nimmt die Wahrscheinlichkeit ab).

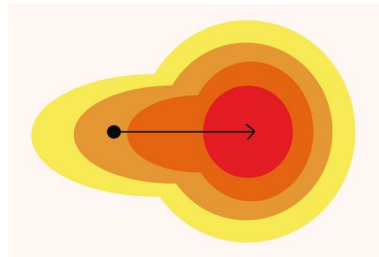


Abbildung 2.37.: MagicMap Aufenthaltswahrscheinlichkeit [Humboldt-Universität zu Berlin](#) (c)

Als Eingangsgrößen für die Erzeugung des Bewegungsmodells werden die Geschwindigkeit v , die Orientierung/Richtung r sowie die zeitliche Differenz zur letzten Ortung Δt benötigt, welche aus dem bisher zurückgelegten Pfad ableitbar sind. Die Birnenform ergibt sich aus dem natürlichen Bewegungsmuster von Menschen, die auf möglichst kurzem und somit geradlinigen Weg zum Ziel gelangen wollen. Die aktuelle Position wird im Birnenhals und die Orientierung r in Richtung Birnenbauch angenommen. Der Mittelpunkt entspricht der Entfernung aus $s(t) = v * \Delta t$. Die Gewichtung der erzeugten Partikel erfolgt nach physikalischer Erreichbarkeit (durch das Umgebungsmodell) sowie der relativen Position zur birnenförmigen Fläche [Humboldt-Universität zu Berlin](#) (e).

Messungen von [Bruning u. a. \(2007\)](#) haben eine Ortungsgenauigkeit von 5-10 m auf Grundlage von alleiniger WLAN-Signalstärkenauswertung ergeben. Durch die Unterstützung von Referenzpunkten konnte eine Präzisionssteigerung von 1,82 m mittlerer Positionierungsfehler erzielt werden. Eine zusätzliche Verbesserung wurde durch den hybriden Ortungsansatz erzielt (1,20 m), indem WLAN- und ZigBee-Signalstärken kombiniert verwendet wurden. Zu den oben beschriebenen Plugins und ihren jeweiligen Ortungsverbesserungen werden keine Aussagen gemacht.

2.3. Fazit

In diesem Abschnitt wird ein Fazit aus den vergleichbaren Arbeiten gezogen. Die Ortungsverbesserung auf Grundlage von rein messwertbasierten Algorithmen bietet für *relative Systeme* hauptsächlich kalibrierende Eigenschaften, um systematische Fehler zu beheben (vgl. Kapitel 2.1.2). Sie wirken dem Drift entgegen und sorgen auf diese Weise für eine verbesserte Ortung. Für *absolute Systeme* und deren zufällige Fehlerverteilung können rein messwertbasierten Algorithmen nur bedingt für eine Verbesserung sorgen. Hierbei wird eine Glättung über mehrere Messwerte durchgeführt, was wiederum zu Einbußen bezüglich des Echtzeitverhaltens führt sowie auch eine Änderung korrekter Messwerte nach sich zieht. Erst mit der Anreicherung zusätzlicher Informationen (z. B. Annahmen über das Bewegungsmodell von Personen) können verbesserte Korrekturen berechnet werden (vgl. Kapitel 2.1.1).

Eine logische Weiterentwicklung in diese Richtung bieten die Map-Matching Verfahren. Die Auswahl an vergleichbaren Arbeiten auf diesem Gebiet fokussiert sich hauptsächlich auf IMU-gestützte Ortungssysteme, welche bauartbedingt durch den akkumulierenden Fehler (Drift) ein korrigierendes Referenzsystem benötigen. Der hybride Ansatz durch den Kartenabgleich führt, je nach verwendetem System, zu erheblichen Präzisionssteigerungen.

Ein relativ einfaches Verfahren (vgl. Kap. 2.2.2) abstrahiert die Karteninformationen auf ein Knoten- und Kantenmodell und korrigiert die aktuelle Position hin zu einem vorher festgelegten Navigationspfad. Das Footpath-Verfahren aus Kapitel 2.2.3 versucht den durch die IMU aufgezeichneten Pfad in den geplanten Navigationspfad einzupassen. Beide Algorithmen sind in ihrer Bewegungsfreiheit sehr stark eingeschränkt, da sie einen Start- sowie Zielpunkt voraussetzen.

Eine Verbesserung demgegenüber stellt die autonome Einpassung von Trajektorien in ein Gebäudemodell aus Kapitel 2.2.3 dar, welches es ermöglicht, die aufgezeichnete Trajektorie automatisiert an einem Graphenmodell auszurichten. Das Gebäudemodell wird aber immer noch stark vereinfacht und auch die aktuelle Position wird nur als Aufenthaltswahrscheinlichkeit auf einer Kante angegeben.

Mit dem CADMS Framework (vgl. Kap. 2.2.3) ist es hingegen möglich, den aufgezeichneten Bewegungspfad in einen Grundriss eines Gebäudes einzupassen. Besonderes Augenmerk wird auf den topologisch korrekten Raumwechsel gelegt und die Trajektorie entsprechend der begrenzenden Faktoren korrigiert. Das sorgt für eine erhebliche Verbesserung der Ortungsgenauigkeit.

Mit Hilfe von Partikelfiltern (siehe Kapitel 2.2.3 ff.) wird eine Anzahl möglicher Aufenthaltspositionen generiert und deren weitere Entwicklung auf Grundlage neuer Bewegungsinformationen

und einem Kartenabgleich bewertet. Ungültige Partikel (welche eine Wand traversieren) werden entsprechend des Prinzips des *Survival of the fittest* verworfen, während die verbleibenden Partikel eine Fläche für die wahrscheinliche aktuelle Position darstellen.

Das Map-Matching an sich benötigt keine zusätzliche Hardware zur Korrektur und minimiert somit den Installationsaufwand und die damit verbundenen Kosten. Es werden jedoch präzise IMU Sensoren verwendet, welche eine Ortung nur mit spezieller Hardware möglich macht. Dies entspricht nicht der Vorstellung für die eigene Realisierung, welche sich auf ein bestehendes absolutes Ortungssystem bezieht. Die verwendeten Algorithmen zur Ortungsverbesserung relativer Systeme sind nicht ohne Weiteres übertragbar, da sie sich meistens auf deren spezifische Eigenschaften spezialisieren und systematische Fehler korrigieren. Daher können lediglich Ideen als Ansatz für einen eigenen Algorithmus aus diesen Realisierungen gezogen werden.

Die vergleichbaren Arbeiten bezüglich absoluter Ortungssysteme (vgl. Kapitel 2.2.4), wie beispielsweise WLAN, sind in diesem Themenbereich nur sehr schwach vertreten. Da diese Systeme alleine keine ausreichende Präzision für viele Indoor Location Based Services besitzen, existiert auch hier der Bedarf an Möglichkeiten zur Ortungsverbesserung.

Ein algorithmisches Verfahren ohne zusätzliche Hardwareanforderungen, welches nur auf logischer Auswertung der Signalstärken basiert, führt noch nicht zu ausreichenden Verbesserungen. Erst mit Hilfe eines Gebäudemodells und Map-Matching können Messfehler und Systemrauschen ausreichend gut erkannt und korrigiert werden. Vergleichbar zu den relativen Verfahren existieren auch für die absoluten Ortungssysteme Partikelfilter mit Map-Matching Funktionalität (vgl. Kap. 2.2.4 ff.), welche zu einer erheblichen Verbesserung der Positionsschätzung führen.

MagicMap fokussiert sich hauptsächlich auf die Multilateration anhand der RSSI-Werte (siehe Kapitel 2.2.4). Hierbei spielen die Informationen aus dem Gebäudemodell eine wichtige Rolle für die berechnete Signaldämpfung und der sich daraus ergebenden Entfernung zum Access Point. Ein Bewegungs- und Umgebungsmodell überprüft die physikalische Erreichbarkeit und sorgt für eine zusätzliche Ortungsverbesserung. Alle Ansätze verwenden auf die ein oder andere Weise einen Partikelfilter um das Map-Matching zu integrieren.

Der Bereich des rein logischen Map-Filterings von ortsdiskreten (absoluten) Positionen bietet die Möglichkeit für eine eigene Entwicklung, da zu diesem speziellen Themengebiet sehr wenig vergleichbare Arbeiten zu finden sind und noch Innovationen realisierbar sind.

3. Analyse

Dieses Kapitel befasst sich mit der Analyse der Anforderungen welche an einen Algorithmus zur Ortungsverbesserung gestellt werden. Zunächst werden in Kapitel 3.1 die Szenarien erläutert welche eine gesteigerte Präzision einer Indoorlokalisierung benötigen. Der Fokus wird auf Anwendungsfälle aus dem öffentlichen Umfeld gelegt, da hier der Bedarf an günstigen und weitflächig abdeckenden Ortungslösungen gegeben ist, welche auf bestehender Infrastruktur basieren (vgl. WLAN-Ortung aus Kap. 2.2.4 sowie Liu u. a. (2007)). Die Szenarien reichen von der einfachen Ortung, über die Indoornavigation, bis hin zur Aktivitätenerkennung anhand von Bewegungsmustern.

Anschließend wird in Kapitel 3.2 eine Anforderungsanalyse durchgeführt, um die in den Szenarien skizzierten Anwendungsfälle abzudecken. Hierfür werden zunächst die Rahmenbedingungen betrachtet, für welche der Algorithmus konzipiert werden soll.

Daraus werden in Kap. 3.2.2 die funktionalen und in Kap. 3.2.3 die nicht-funktionalen Anforderungen an das System formuliert. Dies umfasst das Korrekturverhalten und die geforderten Präzisionen, als auch allgemeine qualitative Anforderungen. In einem abschließenden Fazit (Kapitel 3.3) werden die Ergebnisse der Analyse zusammengefasst und bilden die Überleitung zum Design (siehe Kap. 4).

3.1. Szenarien

Die Szenarien für den Indoorbereich reichen von der einfachen Ortung einer Person, über die Indoornavigation, bis hin zur Aktivitätenerkennung anhand von Bewegungsmustern. Die Anwendungsfälle sind im öffentlichen Bereichen wie z. B. Bürogebäuden, Krankenhäusern oder Flughäfen angesiedelt und profitieren von einer algorithmischen Ortungsverbesserung der bereits bestehenden Installation.

In folgenden Unterkapiteln werden zunächst mögliche Ortungs- und Navigationsszenarien (Kap. 3.1.1) kombiniert vorgestellt. Im Anschluss folgt in Kapitel 3.1.2 ein beispielhaftes Anwendungsgebiet der Activity Pattern Recognition (Aktivitätenerkennung), welches durch eine gesteigerte Ortungsgenauigkeit eine verbesserte Mustererkennung erzielt.

3.1.1. Ortung und Navigation

Die Ortung innerhalb von Gebäuden bietet viele interessante Anwendungsszenarien, angefangen von der einfachen Positionsbestimmung von Gegenständen in einem Warenlager, über die Ortung von Personen in einem Bürokomplex (welche anhand eines Informationssystems den Aufenthaltsort des gesuchten Kollegen angezeigt bekommen), bis hin zur Navigation zu einem beliebigen Punkt innerhalb eines Gebäudes.

Je präziser die Position eines Gegenstandes in einem Warenlager bestimmt werden kann, um so geringer ist der Aufwand und folglich die Kosten, diesen zu finden und auszulagern. Ebenso erspart die Information über den aktuellen Aufenthaltsort eines Kollegen eine zeitintensive Suche in einem weitläufigen Bürogebäude. Eine Navigation unterstützt die Orientierung und Wegfindung innerhalb von Gebäuden und kann auf Grundlage präziser Ortungsinformationen eine Abweichung vom geplanten Navigationspfad schneller feststellen und eine Korrektur bzw. Alternativroute anbieten.

Hochpräzise Ortungssysteme können diese Anforderungen erfüllen, erfordern jedoch hohe Investitionen (vgl. Kap. 2). Daher kommen kostengünstige Varianten in Betracht, welche bestehende Infrastrukturen (z. B. WLAN-Access Points) zur Positionsbestimmung verwenden, aber eine geringere Auflösung besitzen.

3.1.2. Activity Pattern Recognition

Um die Problematik unzureichender Ortungsgenauigkeit und dessen Folgen zu verdeutlichen, wird der Service der *Activity Recognition* herangezogen. Dieser findet Anwendung in der Masterarbeit von Voskuhl (2012), welcher Szenenanalysen im *Living Place* der HAW Hamburg durchgeführt hat (vgl. Kapitel 2.1.1). Hierbei werden anhand von Bewegungsmustern, Schlussfolgerungen auf eine Ausführung einer Aktivität gezogen.

Eine Trajektorie aus dem Bett, über den Flur hin zur Küche, in Kombination mit der Uhrzeit lässt darauf schließen, dass der Benutzer aufgestanden ist und demnächst frühstücken wird. Alle für dieses Szenario festgelegten Aktionen können dann automatisiert ausgeführt werden; Sei es das Öffnen der Fenster, die Lichtsteuerung, oder die Kaffeezubereitung. Ausschlaggebend hierfür ist die möglichst eindeutige Erkennung eines Bewegungsmusters bezüglich einer hinterlegten Vorlage. Ein unpräzises Ortungssystem oder Messfehler erschweren diesen Vergleich erheblich. Entweder nimmt man hohe Toleranzen und somit Fehlinterpretationen in Kauf (ein morgendlicher Gang zur Toilette wird fälschlicherweise als Aufstehszenario interpretiert), oder der Abgleich ist zu restriktiv und die Messfehler führen dazu, dass das Muster erst gar nicht erkannt wird.

Dieses Beispiel und die Szenarien aus Kapitel 3.1.1 verdeutlichen die Notwendigkeit möglichst präziser Ortungsinformationen. Es gilt daher, Ortungsfehler im Vorfeld zu bereinigen bzw. zu korrigieren, bevor sie auf höherer logischer Ebene interpretiert werden.

3.2. Anforderungsanalyse

Der folgende Abschnitt beschäftigt sich mit der Anforderungsanalyse eines Algorithmus zur Ortungsverbesserung für die in Kapitel 3.1 dargestellten Szenarien. Die Ergebnisse des vorherigen Kapitels haben gezeigt, dass unterschiedlichste Anwendungsfälle existieren, welche eine hohe Ortungspräzision benötigen, bzw. durch diese eine gesteigerte Leistungsfähigkeit erbringen und somit angestrebt werden sollten. Die Szenarien sind hauptsächlich im öffentlichen Bereich angesiedelt, was es nötig macht, sowohl den Installationsaufwand durch zusätzliche Hardware, als auch die Kosten gering zu halten, um eine hohe Abdeckung und vielseitige Nutzbarkeit zu ermöglichen.

Es gilt daher zu klären, welche Bedingungen erfüllt sein müssen, um ein bestehendes Ortungssystem ohne die Verwendung zusätzlicher Hardware, sondern auf Grundlage von algorithmischen Auswertungen, in dessen Präzision zu verbessern. Der Fokus liegt hierbei auf einem hybriden Ansatz der algorithmischen Auswertung und Korrektur, wie in Kapitel 2.3 bereits erörtert. Das bedeutet, dass in Hinblick auf die Realisierung keine zusätzliche Hardware verwendet wird, sondern ein Map-Matching basiertes Korrekturverfahren angestrebt werden soll.

Der räumliche Kontext beschränkt sich auf die kontinuierliche Indoorortung von sich bewegendenden Personen. Anhand eines bestehenden absoluten Ortungssystems wird in regelmäßigen Abständen die aktuelle Position bestimmt, welche im Anschluss durch den Algorithmus und mit Hilfe des Map-Matching auf Plausibilität überprüft und ggf. korrigiert wird.

Zunächst werden in Kapitel 3.2.1 die Randbedingungen betrachtet, welche das System begrenzen und Einfluss auf die Konzeption des Algorithmus haben. Basierend darauf werden im Anschluss in Abschnitt 3.2.2 die funktionalen Anforderungen herausgearbeitet, die sich mit den Eigenschaften auseinandersetzen, welche der Algorithmus leisten soll. Dies beinhaltet insbesondere die technischen Anforderungen sowie das Korrekturverhalten spezieller Fehlerfälle. Die nicht-funktionalen Anforderungen in 3.2.3 beschreiben die qualitativen Ansprüche an das System und umfassen beispielsweise die zeitlichen Rahmenbedingungen sowie die transparente Nutzbarkeit gegenüber dem Anwender.

3.2.1. Randbedingungen

Die Randbedingungen, bzw. Rahmenbedingungen stellen die Umgebung des Systems dar, welche Einfluss auf das Korrekturverhalten des Algorithmus haben. Dies sind Einschränkungen als auch Annahmen über die Umgebung, welche als gegeben angenommen werden und erfüllt werden müssen, und eine Grundlage für die Designentscheidung (siehe Kapitel 4) darstellen. Aufgrund der Eigenschaften des Algorithmus müssen gewisse Mindestanforderungen an das Ortungssystem gestellt werden, um die gewünschte Funktionalität zu gewährleisten.

Sollte eine Positionsbestimmung mit einem zu großen Jitter erfolgen, sodass die mögliche Position innerhalb eines Umkreises liegt, der zu viele erreichbare Räume abdeckt, so kann der Algorithmus auf Grundlage der vorherigen (ebenfalls stark schwankenden) Positionsdaten unter Umständen keine eindeutige Lösung bestimmen.

Abbildung 3.1 zeigt ein mögliches Beispiel für den Fall, dass ein Ortungssystem mit einer hohen Toleranz vorliegt.

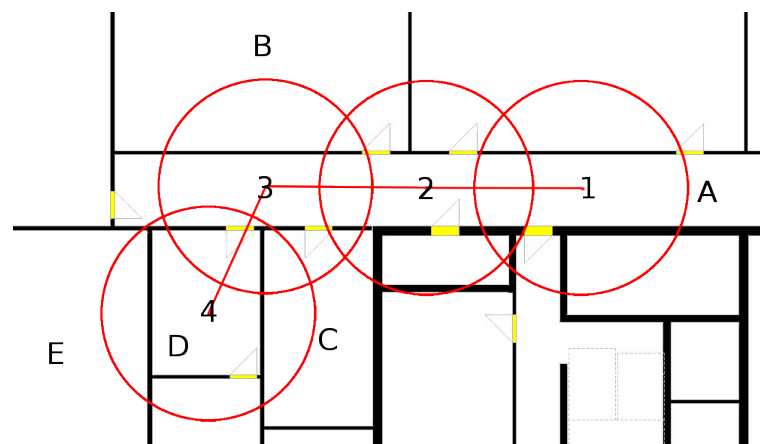


Abbildung 3.1.: Ortungssystem mit einer großen mittleren Abweichung

Unter *Toleranz*, bzw. *Präzision* wird hier die mittlere Abweichung verstanden. D. h. auf Grundlage von früheren Messreihen wurde ermittelt, wie weit die tatsächliche von der angegebenen Position im Schnitt entfernt liegt. Sie bildet somit einen Radius für die Aufenthaltswahrscheinlichkeit. Das Positionsupdate 3 überdeckt mehrere Räume, die tatsächliche Position kann in Raum A, B, C oder D liegen. Die Berücksichtigung der historischen Entwicklung liefert auch keine eindeutige Lösung, da die Raumentwicklung von Position 2 in alle - von Position 3 - abgedeckten Räume logisch möglich ist. Erst mit Position 4 kann rückwirkend Raum B ausgeschlossen werden, da dieser sonst nur für einen Ortungspunkt betreten und sofort wieder verlassen worden wäre. Angenommen die vorherige Position 3 wurde korrekterweise in Raum

A verortet, so kann aufgrund logischer Betrachtung lediglich Raum E als Nachfolger (3 auf 4) ausgeschlossen werden. Die restlichen Räume sind physikalisch über Türen erreichbar.

Es gilt daher im Verlauf der Entwicklung und der Evaluation zu überprüfen, welche Präzision ein Ortungssystem mindestens aufweisen muss, damit noch eine logische Interpretation der Ortungspositionen möglich ist, ohne erst über viele Messungen und Mittelwertbildungen eine Tendenz der wahren Position zu erhalten. Dies würde ansonsten einer Korrektur einer sich bewegenden Person zur Laufzeit widersprechen.

Ebenso spielt die Frequenz der Positionsupdates eine wichtige Rolle, ob ein Ortungssystem geeignet ist. Sollte die Frequenz zu niedrig sein, sodass es einem Menschen möglich ist, innerhalb dieses Zeitabschnittes mehrere Räume zu durchqueren, können keine ausreichend einschränkenden Annahmen aus dem Bewegungsmodell gezogen werden und die Aufenthaltswahrscheinlichkeit würde sich auf eine zu große Fläche verteilen.

3.2.2. Funktionale Anforderungen

Anhand der funktionalen Anforderungen wird das gewünschte Verhalten des Algorithmus sowie dessen Umgebung beschrieben. Im folgenden Abschnitt wird der Begriff des *Systems* verwendet um die Eigenschaften zu erläutern. In Kapitel 3.2.1 wurden die Randbedingungen erläutert, welche nun anhand funktionaler Anforderungen konkretisiert werden.

Die vergleichbaren Arbeiten haben gezeigt, dass ein Abgleich der Ortungsdaten mit einem Gebäudemodell zu einer erheblichen Verbesserung der Ortungsgenauigkeit führen kann. Daher soll das System so ausgelegt werden, dass eintreffende Positionsdaten anhand eines semantischen Geometriemodells räumlich analysiert werden können. Hierfür müssen die Koordinatensysteme und Skalierungsfaktoren in Einklang gebracht werden, um jederzeit ein genaues Abbild der tatsächlichen Position innerhalb des Gebäudes zu erhalten.

Die Herausforderung einer Ortungsverbesserung besteht aus zwei Hauptaspekten. Der *Feststellung einer fehlerhaften Ortung* und der *Korrektur der fehlerhaften Position*. Für ersteres stellt ein Test auf rein physikalische Erreichbarkeit von Raum A zu Raum B über beliebig viele dazwischenliegende Räume und Türen keine ausreichend beschränkende Plausibilitätsüberprüfung dar, da mit hoher Wahrscheinlichkeit kein Raum existiert, der nicht irgendwie erreichbar ist. Um eine logische Überprüfung der Positionsdaten zu ermöglichen wird daher ein Bewegungsmodell benötigt, das Annahmen über die menschliche Bewegung trifft, anhand derer die Plausibilität bestimmt wird. Zu beachten sind vor allem die maximal mögliche Strecke, die ein Mensch innerhalb eines Zeitabschnittes zurücklegen kann, welche abhängig von seiner Höchstgeschwindigkeit ist. Abhängig von dieser vorgegebenen Annahme wird der Aktionsradius bestimmt und der neue Ortungspunkt topologisch auf Erreichbarkeit überprüft.

Weiterführende Einschränkungen durch das Bewegungsmodell nutzen Voskuhl (2012), Khider u. a. (2008) sowie Kaiser u. a. (2011a), indem beispielsweise die bisherige Bewegungsrichtung in die Bewertung mit einbezogen wird.

Im Fall einer festgestellten fehlerhaften Ortung soll eine Korrektur zum nächstgelegenen logisch erreichbaren Punkt innerhalb des Gebäudemodells stattfinden. Dieser Punkt wird definiert aus dem vorherigen (korrigierten) Ortungspunkt P_{n-1} , dem davon ausgehenden maximalen Aktionsradius und dem Abstand zum aktuellen (fehlerhaften) Ortungspunkt P_n . Ziel ist es einen Punkt zu finden, der möglichst nah an P_n liegt und topologisch erreichbar von P_{n-1} entsprechend des Bewegungsmodells ist. Hierfür werden Informationen bezüglich Wänden und Türen zum Wechsel des Raumkontextes aus dem Gebäudeplan verwendet.

Die folgenden Grafiken beschreiben unterschiedliche Korrekturszenarien anhand beispielhafter Bewegungstrajektorien in einem Gebäude. Abbildung 3.2 stellt eine Übersicht auf die unterschiedlichen Pfadtypen dar.

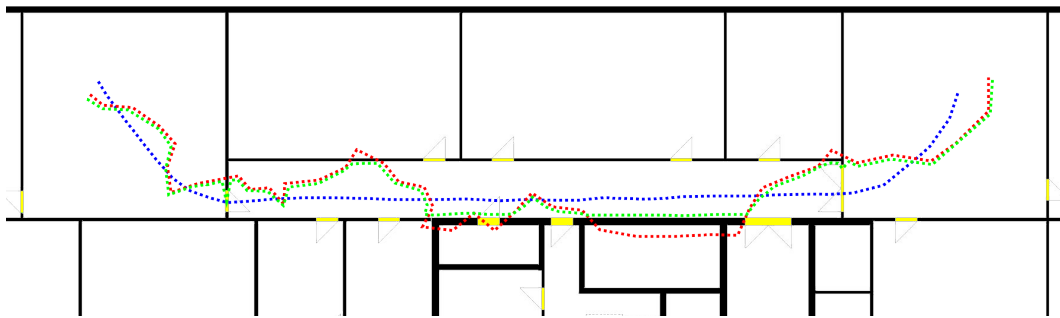


Abbildung 3.2.: Tatsächlicher Pfad (blau), unkorrigierter Pfad (rot), korrigiert (grün)

Die Wände des Gebäudemodells werden als schwarze Polygone dargestellt, Übergänge zwischen den Räumen (Türen) sind gelb eingefärbt. Der punktierte blaue Pfad entspricht der tatsächlich zurückgelegten Trajektorie der zu ortenden Person, die Position wird durch das Ortungssystem bestimmt und liefert abhängig von der Präzision eine abweichende Trajektorie (rot). Anhand der grünen Trajektorie wird das Korrekturverhalten des Algorithmus beschrieben, wie er auf den roten Pfad wirken würde. In den Beispielen werden die einzelnen Ortungspunkte zeitlich und räumlich weit voneinander entfernt dargestellt, um die möglichen Fehlerfälle visuell zu verdeutlichen. Die Punkte auf der Trajektorie entsprechen somit *nicht* einzelnen Ortungspunkten, sondern stellen lediglich eine verbindende Kante dar, welche die zeitliche Abhängigkeit zueinander visualisieren soll.

Ein häufig vorkommendes Szenario wird in Abbildung 3.3 skizziert. Der unkorrigierte Pfad (rot) verläuft durch den Raum und nähert sich der Wand. In diesem Abschnitt folgt

die korrigierte Trajektorie (grün) dem ursprünglichen Verlauf. Sobald die rote Trajektorie (aufgrund von Messfehlern) die Wand durchschreitet, verbleibt die korrigierte Trajektorie im ursprünglichen Raum, da der benachbarte Raum anhand des Bewegungsmodells nicht über naheliegende Türen erreichbar ist.

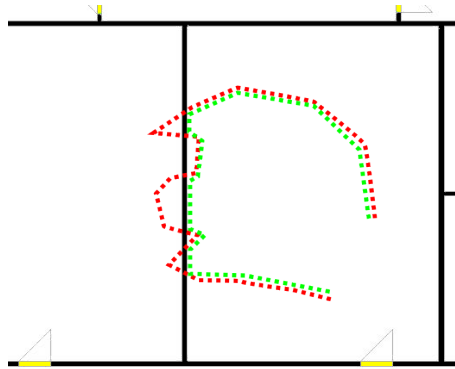


Abbildung 3.3.: Korrektur entlang einer Wand zu einem nicht erreichbaren Raum

Die blaue Trajektorie in Abbildung 3.4 beschreibt einen Weg vom Gang, über einen dazwischenliegenden Raum, hin zum Endpunkt (von rechts nach links). Durch die fehlerhafte Ortung (rot) verläuft die gemessene Trajektorie durch die angrenzenden Räume.

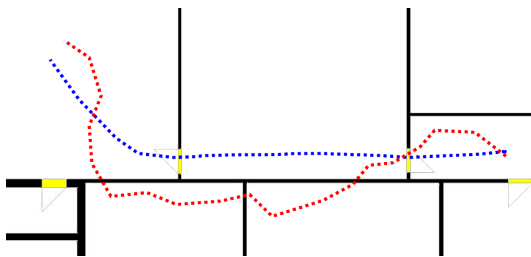


Abbildung 3.4.: Fehlerhafter Pfad (rot) durch unerreichbare Räume

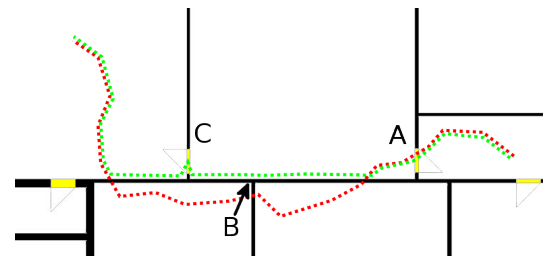


Abbildung 3.5.: Korrektur (grün) über Türen entlang von Wänden

Unter der Annahme, dass die Ortung zu Beginn korrekt ist, sind die unteren angrenzenden Räume topologisch über Türen durch das Bewegungsmodell nicht erreichbar. Der Fehlerzustand wurde somit erkannt und wird mit Hilfe des Algorithmus korrigiert (vgl. Abb. 3.5). Die grüne Trajektorie ist das Resultat zwischen einer möglichst kleinen Distanz zu dem fehlerhaften Pfad (rot) und den logisch erreichbaren Räumen ausgehend von der Startposition über die Tür A. Der Raumwechsel an Punkt B hat keinen Einfluss auf das Korrekturverhalten, da sich der grüne Pfad im darüber liegenden Raum ohne Probleme entlang der Wand bewegen

kann. An Punkt C hingegen ist der Sachverhalt umgekehrt. In diesem Fall muss vor dem Raumwechsel überprüft werden, ob der angrenzende Raum vom bisherigen Verlauf der grünen Trajektorie über eine Tür erreichbar ist und der Pfad entsprechend über diese korrigiert werden. Sobald der rote Pfad wieder in den topologisch erreichbaren Raum wechselt, werden die Ortungsinformationen unkorrigiert übernommen, da keine Einschränkungen durch das Gebäudemodell mehr vorliegen.

Sollte der Algorithmus in einen fehlerhaften Zustand gelangen, indem z. B. die korrigierte Position in einem Raum *hängen bleibt*, ist es nötig, dass dies automatisch erkannt wird und sich der Algorithmus von alleine erholt. In Grafik 3.6 verläuft die korrigierte Position (von rechts gesehen) bis Punkt A identisch zum roten Pfad. Der eigentliche korrekte Pfadverlauf wäre aber C entlang des Flures gewesen, dessen mögliche Pfadentwicklung in diesem Beispiel nicht betrachtet wurde.

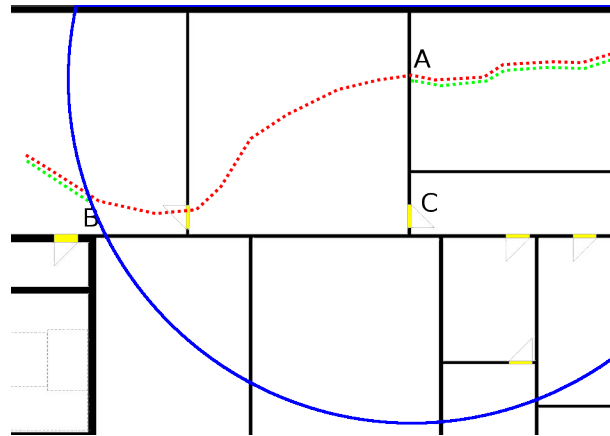


Abbildung 3.6.: Selbstständige Erholung aus einem Fehlerzustand

Entsprechend des Algorithmus wird die Position ab A an der Wand korrigiert, da der nebenliegende Raum nicht logisch über eine in der Nähe befindliche Tür erreichbar ist. Die korrigierte Position verbleibt nun bei A , bis sich der unkorrigierte Ortungspunkt weiter als ein zuvor festgelegter Schwellwert (blauer Kreis um A mit Radius \overline{AB}) entfernt hat. Ab diesem Punkt wird angenommen, dass sich der Algorithmus in einem Fehlerzustand befindet, und die aktuelle unkorrigierte Position (bei B) wird als neuer Ausgangspunkt für die korrigierte Trajektorie angenommen.

Der nächste Fehlerfall beschäftigt sich mit der Wahrscheinlichkeitsbetrachtung für mehrere mögliche folgende Räume. In Abbildung 3.7 wird der erste Schritt dargestellt, der zu einer mehrdeutigen Lösung führen kann.

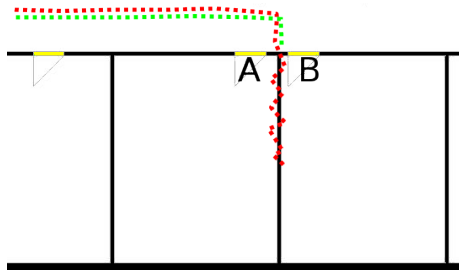


Abbildung 3.7.: Mehrdeutige Lösung beim Raumwechsel

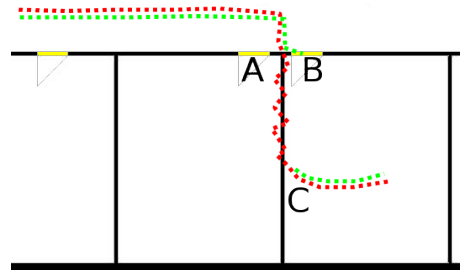


Abbildung 3.8.: Entscheidung für einen Raum erst mit erhöhter Wahrscheinlichkeit

Hierbei verläuft die Trajektorie über den Flur auf die nahe beieinander liegenden Türen *A* und *B* zu, welche in unterschiedliche Räume führen. Sollte sich der weitere Verlauf des Pfades entlang der trennenden Wand erstrecken, so kann nicht eindeutig der tatsächlich betretene Raum bestimmt werden. Für den Fall, dass man sich zu früh für einen Raum entscheidet (während die Ortungspunkte noch in der Nähe der Wand sind) und sich von den Türen weg nach unten bewegt, und die Ortungen anschließend in den benachbarten Raum verlaufen, so würde dieser nachträgliche Raumwechsel nicht mehr durch den Algorithmus erlaubt werden. Daher ist es nötig, bei einem Raumwechsel alle erreichbaren Räume zu betrachten und bei einer mehrdeutigen Lösung, die Alternativen nicht aus den Augen zu verlieren. In Grafik 3.8 verbleibt die korrigierte Position so lange vor den beiden möglichen Räumen, bis anhand der unkorrigierten Trajektorie eine eindeutige Tendenz für einen Raum (siehe *C*) erkennbar ist. Erst dann wird die korrigierte Position in den Raum übernommen, um ein Springen zwischen den Beiden zu verhindern.

Im Falle einer einzigen erreichbaren Tür als Raumübergang, wird diese Entscheidung (wie in Abbildung 3.9 zu sehen) ebenfalls etwas verzögert (Punkt *A*), um ein Springen in benachbarte Räume zu vermeiden, welche bei einer Bewegung entlang eines schmalen Ganges in erreichbarer und logischer Nähe sind.

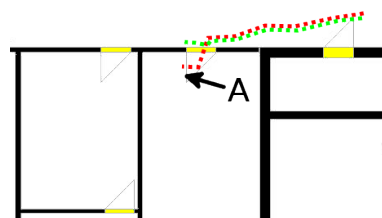


Abbildung 3.9.: Entscheidung für einen Raumwechsel wird verzögert

Die Entwicklung des weiteren Pfades, bzw. die Entscheidung für einen Raumwechsel kann nicht anhand eines einzelnen neuen Messpunkt getroffen werden, da aufgrund der ungenauen Präzision ein benachbarter Raum angegeben werden kann. So kann sich der nächste Ortungspunkt wieder im vorherigen Raum befinden (vgl. Abb. 3.10) und sich der Pfad von A in Richtung B weiter entwickeln. Jedoch kann sich die Trajektorie auch in Richtung C bewegen, wie in Abbildung 3.11 dargestellt.

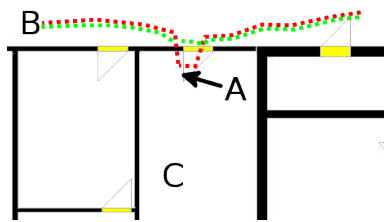


Abbildung 3.10.: Pfadweiterentwicklung von A in Richtung B

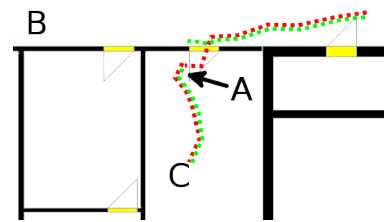


Abbildung 3.11.: Pfadweiterentwicklung von A in Richtung C

Im Falle einer mehrdeutigen Lösung, bzw. mehreren möglichen Positionen in unterschiedlichen erreichbaren Räumen, die eine ähnlich Entfernung zu P_n aufweisen, soll der Algorithmus eine Wahrscheinlichkeitsbetrachtung durchführen und erst anhand zusätzlicher Ortungsinformationen und wachsender Plausibilität eine Entscheidung für einen Raum treffen. Dies kann als eine Art des Mittels angesehen werden, sorgt für eine ruhigere Ortungsposition, führt aber in manchen Situationen zu einem verzögerten Raumwechsel und Sprüngen.

3.2.3. Nicht-funktionale Anforderungen

Anhand der nicht-funktionalen Anforderungen werden die qualitativen Eigenschaften des Systems als Ziel beschrieben und in der Evaluation (siehe Kapitel 5) überprüft. Die Anforderungen werden aus der Sicht des Korrektur-Algorithmus in Bezug zum Anwender formuliert und repräsentieren die Usability des Systems.

Benutzbarkeit

Das erste gewünschte Verhalten bezieht sich auf die Frequenz der Positionsupdates. Jede Aktualisierung der Position des Ortungssystems soll eine Überprüfung durch den Algorithmus nach sich ziehen. Die aktuelle Position wird auf Plausibilität getestet, gegebenenfalls korrigiert und an den Benutzer weitergeleitet. Es besteht somit eine 1:1 Beziehung zwischen ein- und ausgehenden Ortungspositionen. Dies liegt darin begründet, dass nachstehende Anwendungen ein unverändertes Ansprechverhalten bezüglich

des ursprünglichen Ortungssystems benötigen und sich u. U. darauf verlassen. Im Bezug auf die Benutzbarkeit soll daher ein Design angestrebt werden, welches dem Anwender erlaubt, auf ein Ortungssystem zuzugreifen, ohne dem Wissen oder der Sichtbarkeit, ob ein Korrektur-Algorithmus genutzt wird. Somit darf das Korrekturverfahren keine Auswirkungen auf das Ortungssystem haben (welches noch eigenständig lauffähig sein soll) und muss auch in die andere Richtung bezüglich des Anwenders, die gleichen Eigenschaften des Systems imitieren. Dies beinhaltet die unveränderte Frequenz, den strukturellen Aufbau der Nachrichten sowie dessen Übertragungsweg. Die Korrektur und Weiterleitung der Positionsdaten durch den Algorithmus soll hierbei vor dem Eintreffen neuer Ortungsdaten abgeschlossen sein, um ein Aufstauen und Verzögern der Nachrichten zu vermeiden, welches dem transparenten Design widersprechen würde.

Korrektheit und Zuverlässigkeit

Die Überprüfung der Ergebnisse des Algorithmus stellen eine interessante Herausforderung dar, da ein Referenzsystem zur Feststellung von Korrekturfehlern nicht vorhanden ist. Die Positionsangaben eines echten Ortungssystems sind bereits fehlerbehaftet und die tatsächlich zurückgelegte Trajektorie ist nicht verfügbar um die Korrektur des Algorithmus zu evaluieren. Es könnte nur die logische Korrektheit bezüglich der aufgestellten Hypothese des betretenen Raumes anhand der zurückgelegten Trajektorie festgestellt werden. Von einer *korrekten Position* spricht man in diesem Zusammenhang als einer möglichst geringen Anzahl falscher Hypothesen der Pfadentwicklung und den daraus folgenden fehlerhaften Ortungspunkten.

Um Aussagen über die Qualität des Algorithmus treffen zu können soll das System daher zunächst mittels eines simulierten Ortungssystems überprüft werden. Hierbei besteht die Möglichkeit, genaue Messungen bezüglich der zurückgelegten (simulierten) und der korrigierten Trajektorie durchzuführen, da der tatsächliche Pfad absolut vorgegeben ist. Es existieren somit drei Trajektorien (vgl. Abb. 3.2), der tatsächlich zurückgelegte Pfad (blau), der Pfad des simulierten Ortungssystem mit künstlichen Messfehlern (rot) sowie der korrigierte Pfad (grün) des Algorithmus auf Basis des roten Pfades. Dies hat den Vorteil, dass alle Parameter des Ortungssystems einstellbar sind (Updatefrequenz, Jitter, extreme Ausreißer, usw.) und nach der Korrektur direkt mit der tatsächlichen Trajektorie verglichen werden können; Sei es für Messungen der mittleren Abweichung oder um die Leistungsfähigkeit anhand ausgewählter Szenarien zu testen und Erfahrungswerte zu sammeln, um daraus Prognosen für das spätere echte Korrekturverhalten abzuleiten und eine Bewertungsgrundlage für die Evaluation zu erhalten. Durch die Simulation kann systematisch getestet werden an welchen Stellen der Algorithmus an seine Grenzen

stößt, um ihn iterativ zu verbessern und Fälle zu identifizieren, an denen eine Korrektur nicht mehr möglich ist. Hierfür soll das System die nötigen Informationen für die spätere Auswertung bereit halten und entsprechend konzipiert werden.

Skalierbarkeit

Neben der mittleren Abweichung des Korrektur-Algorithmus spielt die Skalierbarkeit ebenfalls eine wichtige Rolle. So wird es nicht ausreichend sein, eine Person zur Zeit in einem Gebäude zu verorten und dessen Trajektorie zu korrigieren, sondern es wird nötig sein eine variierende Menge an zeitgleichen Ortungspositionen zu bearbeiten und zu verwalten. Zusätzlich kann die Gebäudegröße und strukturelle Komplexität je nach Einsatzort variieren. Die Komplexität des Themenbereichs der Skalierbarkeit ist nicht zu unterschätzen und beschäftigt sich mit der Verteilung sowie der Koordinierung von Aufgaben. Durch das Clustern und Synchronisieren von Berechnungen können zeitgleich mehrere Aufgaben abgearbeitet werden und einer zunehmenden Nachfrage gerecht werden. Da man sich beliebig mit dem Thema der Skalierbarkeit und Parallelisierung beschäftigen kann und der Bottleneck wahrscheinlich bei dem (gegebenen und unveränderlichen) Ortungssystem liegt, ist dies nicht Fokus der Arbeit. Das System soll jedoch im Hinblick auf eine definierte Anzahl Nutzer die entsprechenden Ortungspfade zeitgleich verarbeiten und korrigieren können, und zusätzlich unterschiedlich skalierende Gebäudemodelle unterstützen.

Änderbarkeit und Austauschbarkeit

Der Algorithmus verwendet ein bestehendes Ortungssystem und nutzt intern ein Gebäudemodell zum Map-Matching. Es ist ein eigenständiges System zwischen dem Ortungssystem und dem Anwender und soll daher modular konzipiert werden, ohne starke Abhängigkeiten zu anderen Komponenten aufzubauen, um es optional zu halten. Eine gekapselte Modellierung bietet die Möglichkeit der flexiblen Integration in andere Systeme ohne aufwändige Anpassungen am System oder zugeschnittene Komponenten. Eine Portierung an einen neuen Standort kann es aufgrund veränderter Infrastruktur nötig machen, das Ortungssystem auszutauschen. Sofern die definierten Schnittstellen des Systems genutzt werden, ist ein Austausch des Ortungssystems vorstellbar. Denn der Algorithmus ist unabhängig davon lauffähig und betrachtet nur einzelne Ortungspunkte, ohne die Funktionsweise des Ortungssystems kennen zu müssen. Die Einschränkungen aus den Randbedingungen (siehe Kapitel 3.2.1) müssen jedoch beachtet werden.

3.3. Fazit

In diesem Kapitel wurde gezeigt, dass die Relevanz einer Ortungsverbesserung besonders für größere öffentliche Gebäude gegeben ist. Die wesentlichen funktionalen Anforderungen fokussieren sich auf die *Identifikation eines Fehlers* und die anschließende *Ortungsverbesserung für mehrdeutige Korrekturpfade*, wobei ein Gebäude- und Bewegungsmodell verwendet werden soll um physikalisch mögliche Pfade zu identifizieren. Auf dieser Grundlage soll ausgehend von einer bekannten Position eine logische Betrachtung der Ortungspunkte erfolgen. Die zurückgelegte Trajektorie soll durch das Gebäudemodell beschränkt werden und lässt nur physikalisch erreichbare und dem Bewegungsmodell entsprechende logische Pfade zu. Sollte ein neuer Ortungspunkt außerhalb der Wahrscheinlichkeitsflächen liegen, so wird dieser als fehlerhaft eingestuft und korrigiert.

Eine Korrektur soll in einem Fehlerfall angewendet werden und hat folgende Eigenschaften:

- Die korrigierte Position ist vom vorherigen Ortungspunkt logisch erreichbar
- Der Abstand zum ursprünglichen (fehlerhaften) Ortungspunkt soll möglichst gering sein
- Mehrdeutige Pfadentwicklungen werden separat verfolgt und ihre Eintrittswahrscheinlichkeit bewertet
- Die Entscheidung für einen Raumwechsel ist abhängig von der Wahrscheinlichkeit, dass die Position darin liegt.

Entsprechend der nicht-funktionalen Anforderungen der Änderbarkeit und Austauschbarkeit soll das System möglichst stark gekapselt werden und über definierte Schnittstellen ein optionales Modul zwischen dem Ortungssystem und dem Anwender darstellen. Besonders im Bezug auf das *Look and Feel* aus der Sicht des Betrachters zu den vorherigen Schnittstellen des Ortungssystems ist zu beachten, dass eine relativ einfache Umstellung auf das neue Korrekturmodul möglich ist. Die Skalierbarkeit spielt bei dieser Realisierung eine untergeordnete Rolle und fordert lediglich die zeitgleiche Unterstützung der Korrektur von mehr als einer Person. Anhand der Forderungen bezüglich der Benutzbarkeit soll der Anwender keinen Unterschied im Verhalten zum ursprünglichen Ortungssystem feststellen. Dies beinhaltet u.a. die Frequenz der Aktualisierung sowie die 1:1 Beziehung der Positionsupdates zwischen den Ausgängen des Ortungssystems und des Korrektur-Algorithmus.

Die erreichbare Genauigkeit hängt sehr stark von dem verwendeten Gebäudemodell und dessen logischen Beschränkungen für den Bewegungspfad ab und kann daher im Vorfeld nicht quantifiziert werden. Ein minimales Qualitätskriterium fordert, dass der Korrektur-Algorithmus

einen durchgängig logischen Pfad durch das Gebäude liefert. Dieser soll der tatsächlichen Abfolge an Raumwechseln entsprechen, oder der Wahrscheinlichkeitsentwicklung bestmöglich folgen.

Speziell konzipierte Fehlerfälle bezüglich der möglichen Trajektorien (vgl. Kapitel 3.2.2) sollen anhand von Testszenarios systematisch gegenüber dem Algorithmus überprüft werden. Um Aussagen über die Korrektheit und Zuverlässigkeit treffen zu können, wird das Ortungssystem simuliert und Messungen bezüglich der tatsächlichen, ursprünglichen und korrigierten Position durchgeführt. Die Simulation ermöglicht es, diese Testfälle kontrolliert durchzuführen und mit den Ergebnissen als Rahmen- und Randbedingungen in die Realität zu wechseln ohne diese erneut überprüfen zu müssen.

4. Design

In diesem Kapitel werden die Anforderungen und Komponenten aus der Analyse (siehe Kapitel 3) vorgestellt und zu einem System zusammengefasst.

Zunächst wird in Abschnitt 4.1 die generelle Idee des eigenen Algorithmus abstrakt erklärt und auf die unterschiedlichen Entwicklungsrichtungen eingegangen, welche sich in einer restriktiven bzw. fehlertoleranten Korrektur widerspiegeln.

In Kapitel 4.2 werden die selbst entwickelten Algorithmen zunächst im grundlegenden Prinzip dargestellt und deren Gemeinsamkeiten (Kap. 4.2.1) herausgearbeitet. Danach werden Spezialfälle (Kap. 4.2.2) und die Funktionsweise sukzessiv im Detail betrachtet (Kap. 4.2.3). Die beiden Algorithmen werden im Anschluss getrennt betrachtet und deren unterschiedliches Korrekturverhalten herausgearbeitet (Kap. 4.2.4 und 4.2.5). Anschließend erfolgt in Kapitel 4.3 ein *Top-Down*-Überblick auf das Gesamtsystem. Die jeweiligen Teilkomponenten, bestehend aus dem Ortungssystem, der bestehenden ISIS¹-Infrastruktur zur Verwaltung von Ortungsinformationen und Visualisierung sowie das optionale Korrektursystem werden aufgeschlüsselt und separat im Detail betrachtet. Dies erfordert unter anderem eine Gebäudemodellierung (siehe Kap. 4.3.4) sowie deren zusätzliche Anreicherung mit semantischem Wissen, um einen ausreichenden *level of detail* zu erzielen.

Für die Visualisierung 4.3.5 muss ein vorhandenes Framework um zusätzliche Funktionalitäten erweitert werden, um Korrekturinformationen und Pfadhistorien zu vermitteln.

4.1. Idee

Die in Kapitel 2 angesprochenen Herausforderungen der präzisen Indoorortung sind ein aktuelles Forschungsgebiet mit vielen unterschiedlichen Entwicklungsrichtungen und Lösungsansätzen (vgl. Mestre u. a. (2011)). In dem Fall des eigenen Algorithmus findet zunächst eine Spezialisierung auf ein Themengebiet statt: Die Ortungsverbesserung bestehender absoluter Lokalisierungsverfahren anhand eines logischen Abgleichs mit Gebäudemodellen. Hierfür ist es nötig, zunächst die Schwierigkeiten und Problemfälle der Ortung innerhalb von Gebäuden zu

¹Indoor Spatial Information Service

identifizieren. Ein Vergleich mit den Outdoor-Navigationssystemen und deren Map-Matching-Algorithmen ist hilfreich, da die Problemstellung ähnlich ist. Eine Ortungsposition wird dort mit einer Karte abgeglichen und unter Berücksichtigung der Bewegungsrichtung zur nächstgelegenen Straße verschoben.

Im Gegensatz dazu ist das Indoor-Map-Matching um einiges komplexer, da die zugrunde gelegte Karte keinem Knoten- und Kantenmodell entspricht, sondern Flächen mit Übergangsbeschränkungen aufweist und einem einzelnen, unabhängig betrachteten Ortungspunkt wenige Einschränkungen für eine Korrekturlogik bietet. Zusätzlich sind die Freiheitsgrade der Bewegung² vielfältiger als bei der Autonavigation, welche linear entlang der Straße verläuft. Die Präzision des Ortungssystems zählt ebenfalls zu den Risikofaktoren und muss im Indoorbereich um einiges genauer sein als z. B. außerhalb durch das GPS, da die Strukturen in denen navigiert wird um ein vielfaches enger zusammen liegen. Sollte die Positionsbestimmung zu ungenau sein und willkürlich durch das Gebäude wandern, kann auch mit Hilfe logischer Auswertungen keine Verbesserung zu erwarten sein.

Nachdem die Schwierigkeiten der Indoorortung angerissen wurden, besteht nun die Aufgabe darin, ein Konzept zu entwickeln, welches eine Präzisionssteigerung der Ortungsinformationen bieten kann und auf die Herausforderungen im Indoorbereich zugeschnitten ist. Die Ortungsverbesserung lässt sich zunächst in zwei Aufgabenbereiche unterteilen, in die *Identifizierung einer fehlerhaften Ortungsposition* sowie deren anschließende *Korrektur*.

Personen bewegen sich innerhalb eines Gebäudes, deren Position durch ein fehlerbehaftetes Ortungssystem bestimmt wird. Da kein alternatives Ortungssystem als Referenz verwendet werden kann, um einen Fehler gegenüber der tatsächlichen Position festzustellen, muss der Fehlerzustand neu definiert werden. Es können nur die Messwerte auf logische Korrektheit gegenüber dem Gebäudemodell überprüft werden und aus einem auftretenden Widerspruch abgeleitet werden, ob ein Fehler vorliegt und einer Korrektur bedarf. Ein Map-Matching eines einzelnen Ortungspunktes in einen Raum erscheint nicht zielführend, da nahezu der gesamte Grundriss aus Räumen besteht und eine solche Korrektur lediglich verhindern würde, dass die Position nicht innerhalb einer Wand verortet wird.

Demnach müssen die Ortungspunkte in Relation zueinander gesehen und eine Pfadhistorie gebildet werden, um daraus zusätzliches semantisches Wissen ableiten zu können. Sobald der aktuelle Punkt mit seinen Vorgängern verglichen wird, kann mit Hilfe des Map-Matching eine logische Betrachtung des zurückgelegten Weges angestellt werden.

Vereinfacht dargestellt, soll überprüft werden, ob seit der letzten Ortung (P_{n-1}) und der aktuellen Ortung (P_n) ein Raumwechsel stattgefunden hat. Sollte dies der Fall sein, werden

²Auch bekannt als: Liberty of movement

mögliche Übergänge (Türen) gesucht, welche in erreichbarer Nähe sind und einen logischen Pfad zwischen P_{n-1} und P_n ermöglichen. Schlägt diese Überprüfung fehl und es liegt ein logischer Widerspruch vor, kann P_n als fehlerhafter Ortungspunkt angesehen werden³, welcher im nächsten Schritt korrigiert wird. Die *Korrektur* verschiebt den aktuellen Punkt an eine von P_{n-1} erreichbare Position. Da hierfür meistens eine Vielzahl an Möglichkeiten besteht, findet eine Bewertung der alternativen Positionen statt. In die Bewertung fließen unter anderem der Abstand von $P_n^{korrigiert}$ zu P_n , als auch zu P_{n-1} ein, welche mit dem *Bewegungsmodell* eines Menschen in Gebäuden sowie der *physikalischen Erreichbarkeit* verglichen werden. Zusätzlich erfolgt eine Wahrscheinlichkeitsbetrachtung, ob ein Raumwechsel tatsächlich stattgefunden hat und welcher Raum bei einer mehrdeutigen Lösung zu bevorzugen ist.

Für den Korrektur-Algorithmus bestehen zwei grundlegende Entwicklungsrichtungen, welche auf dem unterschiedlichen Verständnis bezüglich einer *Korrektur* beruhen. Das geplante System soll eine Echtzeitkorrektur anbieten, d. h. für jeden neuen Ortungspunkt wird der (zu dieser Zeit) wahrscheinlichste Korrekturpunkt berechnet und ausgegeben. Eine nachträgliche Korrektur des bereits korrigierten Punktes ist somit nicht möglich.

Entweder verfolgt man den Ansatz, dass die zurückgelegte Trajektorie *dauerhaft zusammenhängend und plausibel* ist, geht damit aber auch das Risiko ein, dass eine fehlerhafte Entscheidung für einen Raumwechsel aufgrund des restriktiven Verhaltens zu einem Deadlock führt. Oder man verfolgt den Ansatz, dass man mehrere Hypothesen der Raumentwicklung parallel betrachtet und sich flexibel für die größte Wahrscheinlichkeit entscheidet. Dies wirkt dem oben beschriebenen Deadlock entgegen, führt jedoch dazu, dass die zurückgelegte Trajektorie Sprünge aufweisen kann, was eine anschließende Auswertung (z. B. Activity Pattern Recognition) erschwert. Beide Ideen haben ihre Berechtigung und führen aufgrund ihrer Vor- und Nachteile zu unterschiedlichen Korrekturverhalten. Es gilt daher, beide Ansätze separat umzusetzen und deren Leistungsfähigkeit und Schwächen anhand von Testszenarien herauszuarbeiten.

4.2. Algorithmen

In diesem Kapitel wird der Algorithmus der eigenen Entwicklung erläutert. Das Funktionsprinzip wurde im vorherigen Kapitel 4.1 bereits angesprochen und wird im folgenden Abschnitt konkretisiert. Hierbei erfolgt zunächst ein grober Gesamtüberblick und im Anschluss deren jeweilige spezifischen Funktionsweisen (siehe Abschnitt 4.2.1). Das Kapitel 4.2.2 geht gesondert auf die Spezialfälle der Initialisierungsphase und Deadlocklösung ein, welche eine abgeänderte algorithmische Auswertung verwenden. Für die logische Interpretation der Pfad-

³Unter der Annahme, dass P_{n-1} korrekt ist, bzw. bereits korrigiert wurde

historie wird eine zusätzliche Wahrscheinlichkeitsbetrachtung zur Bewertung hinzugezogen (vgl. Kap. 4.2.3). Diese wird als Grundlage für die folgende Korrekturentscheidung benötigt. Aufbauend auf diesem Wissen werden in den beiden Unterkapiteln (4.2.4 und 4.2.5) alternative Entwicklungsrichtungen des Algorithmus thematisiert, welche einem unterschiedlichen Korrekturverständnis entsprechen.

Der grundlegende Ansatz geht von einem zeitdiskreten Ortungssystem aus, welches kartesische Koordinaten bezüglich eines definierten Ursprungs liefert. Die Positions-Rohdaten werden in einer Pfadhistorie gespeichert und der aktuelle Ortungspunkt dagegen abgeglichen. Der Algorithmus zur Ortungsverbesserung empfängt die Positionsangaben des Ortungssystems, korrigiert diese und leitet sie an den Nutzer weiter, wobei eine Korrektur der Koordinaten nur nach Erkennung eines Fehlers erfolgt und ansonsten unverändert bleibt.

Einige folgend verwendete Begriffe bedürfen einer Erklärung, damit deren Bedeutung in diesem Kontext eindeutig und nachvollziehbar ist. Für das Map-Matching wird ein Gebäudemodell verwendet, dessen Detailgrad reduziert wurde. Wände definieren *Räume* und Türen bilden *Übergänge* zwischen den Räumen. Ein Raumwechsel kann somit nur anhand von Übergängen erfolgen, eine Traversierung von Wänden wird als logisch inkorrekt bewertet.

Die Begriffe der *Ortung*, *Position* oder des *Punktes* werden als synonym angesehen und beschreiben jeweils einen Koordinatenpunkt, welcher ggf. mit zusätzlichen Informationen angereichert sein kann. Die historische Aneinanderreihung von Punkten ergibt einen *Pfad*, bzw. eine *Trajektorie*. Ein *ursprünglicher* Punkt bzw. Pfad wird durch das Ortungssystem geliefert und ist im Rahmen seiner Präzision - entgegen dem *tatsächlichen* Punkt - fehlerbehaftet. Der Algorithmus sendet nach seiner logischen Analyse *korrigierte* Positionen.

Die Auswertung überprüft den aktuellen Ortungspunkt auf *logische* Kontinuität bezüglich der Pfadhistorie und entscheidet, ob ein Raum auf *legale Weise* betreten wurde, oder ob ein Widerspruch vorliegt und der Ortungspunkt als *falsch* eingestuft wird. Anhand dieser Definitionen sollte ein leichteres Verständnis der folgenden Funktionsweise des Algorithmus möglich sein.

4.2.1. Gemeinsamkeiten

Der Algorithmus wird (vereinfacht) in einem Ablaufdiagramm (siehe Abb. 4.1) dargestellt. Das Grundprinzip ist für beide Entwicklungsrichtungen gleich und unterscheidet sich erst im Detail durch die Korrektur der Position (vgl. Schritt 6). Daher kann im Vorfeld die Funktionsweise kombiniert erläutert werden.

Der Algorithmus empfängt jeweils den aktuellen Ortungspunkt P und bildet diesen im Gebäude ab, um den umgebenden Raum zu identifizieren (Schritt 1). Für den Fall, dass das System bereits initialisiert ist und anhand vorhergegangener Ortungen der zuletzt legal betretene

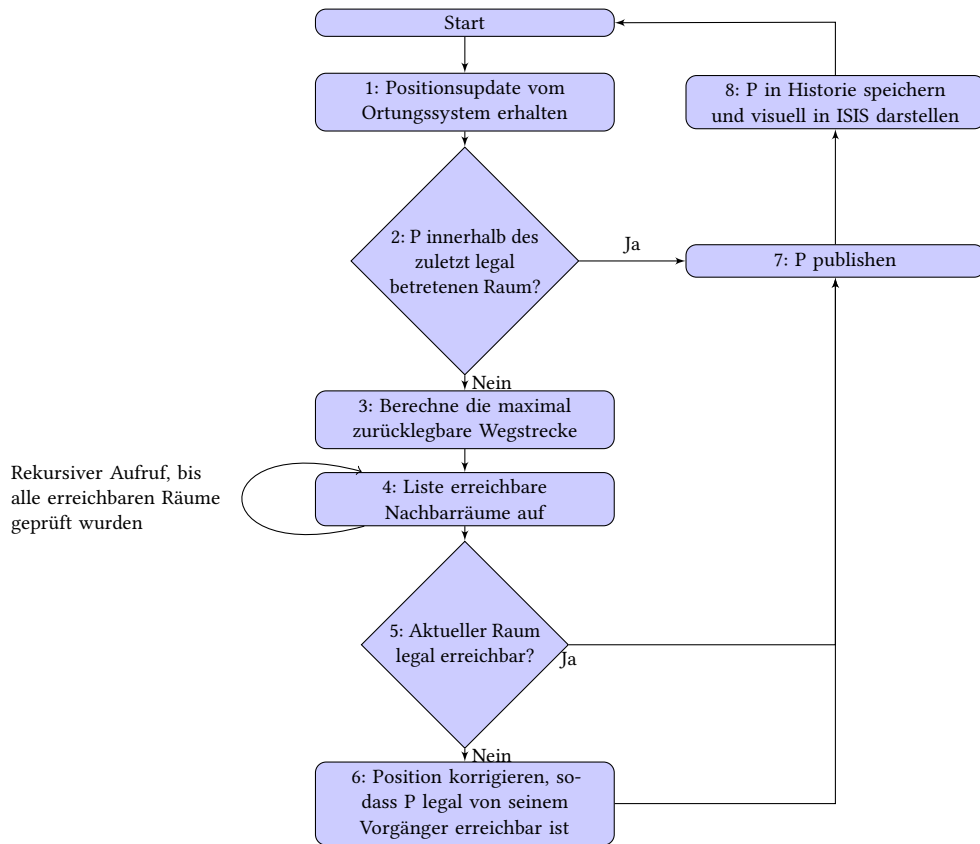


Abbildung 4.1.: Ablaufdiagramm des selbst entwickelten Algorithmus

Raum bekannt ist, wird der aktuelle Raum von P damit verglichen (vgl. Schritt 2). Sollten die Räume identisch sein, existieren keine baulichen Einschränkungen für eine logische Korrektur und die Position wird unverändert weitergeleitet.

Bei einem erkannten Raumwechsel wird in Schritt 3 zunächst die maximal mögliche zurücklegbare Distanz berechnet. Dieser Aktionsradius ergibt sich aus der Zeitdifferenz zur vorhergehenden Ortung und einer im Bewegungsmodell festgelegten Höchstgeschwindigkeit von Personen innerhalb von Gebäuden.

In Schritt 4 werden ausgehend von der letzten legalen Position rekursiv alle erreichbaren Türen untersucht, um eine Verbindung zwischen dem letzten legalen Raum und dem aktuellen Raum zu finden. Sollte die aktuelle Position auf diese Weise erreichbar sein, ist ein Raumwechsel entsprechend des Bewegungsmodells logisch und somit erlaubt.

Der Ansatz des logischen Widerspruchs setzt voraus, dass Positionsupdates nicht als einzelne Messwerte betrachtet werden, sondern in einen zeitlichen Zusammenhang gebracht werden müssen, um Aussagen über die Historie des zurückgelegten Weges treffen zu können und die Plausibilität zu überprüfen. Die Trajektorie der Ortungspunkte wird im Gebäudemodell abgebildet und für jeden neuen Ortungspunkt P_n wird in Schritt 5 eine logische Überprüfung auf Plausibilität gegenüber dem jeweiligen Vorgänger P_{n-1} vorgenommen. Dies geschieht anhand verschiedener Bewertungsfaktoren, welche das *Bewegungsmodell* eines Menschen in Gebäuden betrachten sowie die *physikalische Erreichbarkeit* überprüfen.

Für den Fall, dass ein Raumwechsel als illegal eingestuft wird, muss eine Korrektur durchgeführt werden (vgl. Schritt 6). Da sich die Algorithmen in diesem Punkt unterscheiden, werden sie in den Unterkapiteln *Restriktiver Ansatz* und *Fehlertoleranter Ansatz* separat betrachtet. Der Rückgabewert der Korrektur entspricht einem veränderten Ortungspunkt, welcher in Schritt 7 an die Umgebung mitgeteilt wird und in Schritt 8 in die Pfadhistorie aufgenommen und visualisiert wird.

Nachdem eine Übersicht auf die Funktionsweise des Algorithmus gegeben wurde, folgt nun eine vertiefende Betrachtung der einzelnen Schritte.

Schritt 1

Die vom Ortungssystem übermittelten Positionsupdates enthalten eine eindeutige Identifizierungsnummer, um mehrere Personen und deren Trajektorien parallel betrachten zu können. Die Position wird in Form von Koordinaten angegeben und muss in dem verwendeten Gebäudemodell abgebildet werden. Ein gemeinsamer Koordinatenursprung und die entsprechende Skalierung sorgen für einen optimalen Abgleich um den umgebenden Raumkontext aus dem Modell zu extrahieren. Dies erfolgt anhand bereits verfügbarer Funktionen von Bibliotheken, welche in Kapitel 4.3.4 in Verbindung mit dem Gebäudemodell näher erläutert werden.

Schritt 2

Der umgebende Raum R_{aktuell} des aktuellen Ortungspunktes wird mit dem zuletzt legal betretenen Raum R_{legal} verglichen. R_{legal} ist mit dem umgebenden Raum von P_{n-1} gleichzusetzen, da davon ausgegangen wird, dass die vorherige Position korrigiert wurde. Sollten die beiden Räume identisch sein, kann keine Überprüfung auf die logische Erreichbarkeit zwischen den Räumen vorgenommen werden, da keine baulichen Hindernisse die Bewegung einschränken könnten. Eine Überprüfung innerhalb von Räumen könnte höchstens auf Ebene des Radius-Filters erfolgen (vgl. Kap. 2.1.1) um Ausreißer zu identifizieren. Doch für die folgende Korrektur würden keine ausreichenden Informationen für

eine verbesserte Alternativposition vorliegen. Demnach wird die Ortungsposition P_n für den Fall $R_{aktuell} == R_{legal}$ als korrekt angenommen und der Algorithmus springt zu Schritt 7. Sollte ein Raumwechsel stattgefunden haben, kann eine logische Überprüfung durchgeführt werden und der Algorithmus geht zum nächsten Schritt (3) über.

Schritt 3

Da nahezu jeder Raum über einen beliebig langen Weg durch Türen im Gebäude erreichbar ist, benötigt man einen einschränkenden Kontrollmechanismus. Dieser besteht aus einem Entfernungsfiler und schränkt die Suche nach erreichbaren Räumen ein. Die maximal zurücklegbare Distanz berechnet sich nach der Formel $\Delta s = \bar{v} * \Delta t$. Anhand von Zeitstempeln der Positionsupdates lässt sich die Zeitdifferenz $\Delta t = t_{P_n} - t_{P_{n-1}}$ bestimmen. Durch Multiplikation mit einer vorgegebenen Höchstgeschwindigkeit v_{max} erhält man die maximal mögliche Distanz, die innerhalb dieses Zeitabschnittes zurückgelegt werden kann. Der Geschwindigkeitswert beruht auf einem Bewegungsmodell, welches festlegt wie schnell sich eine Person durch ein Gebäude bewegen kann. Im Rahmen der Simulation und Evaluation wird die Distanz fest vorgegeben um ein gleichbleibendes und visuell nachvollziehbares Korrekturverhalten für geplante Testszenarien zu erhalten.

Schritt 4

In Abbildung 4.2 wird die rekursive Raumsuche dargestellt. Ausgehend von der letzten

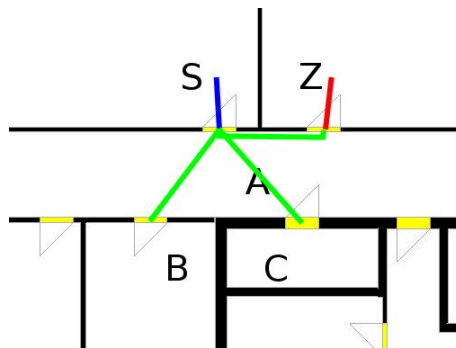


Abbildung 4.2.: Rekursive Raumsuche

legalen Position (Start = S), werden alle erreichbaren Türen aufgelistet (blaue Strecke). Von jedem gegenüberliegenden Raum (in diesem Fall ist nur eine Tür von S erreichbar) wird diese Suche rekursiv fortgesetzt (grüne Strecken zu den Räumen B, C sowie dem Zielraum), bis entweder die maximale Distanz überschritten ist, oder der Zielpunkt (rote Linie nach Z) erreicht wird. In der eigenen Realisierung erfolgt dies durch einen Funktionsaufruf mit den dazugehörigen Parametern wie z. B. dem Start- und Zielraum

sowie dem zur Verfügung stehenden Suchradius. Um Schleifen in der Rekursion zu unterbinden, wird eine temporäre Historie an bereits untersuchten Räumen angelegt und die Rekursion bei einem erneuten Besuch abgebrochen. Für die Berechnung der zurückgelegten Distanz wird der Abstand zwischen dem Startpunkt und dem Türschwerpunkt sowie der jeweilige Abstand zwischen den folgenden Türschwerpunkten und dem Zielpunkt addiert. Die Rekursion bricht sofort bei einer Überschreitung der maximal erlaubten Distanz ab oder liefert bei einer erfolgreich gefundenen Verbindung den logisch zurückgelegten Pfad.

Schritt 5

Sollte die aktuelle Position auf diese Weise erreichbar sein, ist ein Raumwechsel entsprechend des Gebäude- und Bewegungsmodells logisch und somit erlaubt. Der Algorithmus akzeptiert den Raumwechsel und übernimmt P_n als korrekten Ortungspunkt und setzt $R_{\text{aktuell}} = R_{\text{legal}}$, sofern die Wahrscheinlichkeit für R_{aktuell} hoch genug ist. Die Berechnung der Wahrscheinlichkeit für die Räume wird in einem eigenen Kapitel [4.2.3](#) gesondert thematisiert. Im Falle eines erlaubten Raumwechsels springt der Algorithmus zu Schritt 7, ansonsten ist eine Korrektur (Schritt 6) nötig.

Schritt 6

Sollte eine fehlerhafte Ortung identifiziert worden sein, so wird sie in diesem Schritt korrigiert. Neben der Wahrscheinlichkeitsberechnung verhält sich der Algorithmus an dieser Stelle je nach gewähltem Korrekturverfahren unterschiedlich (siehe Kapitel [4.2.4](#) und [4.2.5](#)). Der als *korrekt* angenommene Raum $R_{\text{korrigiert}}$ kann sich demnach unterscheiden. Die darauf folgende Berechnung verhält sich dann wieder gleich. Nachdem $R_{\text{korrigiert}}$ festgelegt wurde, muss noch die Position $P_n^{\text{korrigiert}}$ bestimmt werden. Hierfür wird entlang des Raumpolygons ein Punkt gesucht, dessen Entfernung zum ursprünglichen Ortungspunkt P_n minimal ist. Das resultierende Verhalten äußert sich in einem korrigierten Punkt, der sich in einem legalen Raum befindet und zugleich versucht die ursprüngliche Ortung mit einzubeziehen, indem er sich entlang der Wand bewegt.

Schritt 7

Sobald die korrigierte Position $P_n^{\text{korrigiert}}$ berechnet wurde, muss diese an die Umgebung mitgeteilt werden. Dies erfolgt über die gleichen Kanäle wie der Empfang des Positionsupdates seitens des Ortungssystems. Hierbei wird lediglich ein neues Topic (im Fall einer Blackboard-Kommunikation) verwendet, um zwischen unkorrigierten und korrigierten Messwerten von außerhalb unterscheiden zu können. Der Aufbau und die

Struktur der Nachricht entsprechen dem Original und sind höchstens um zusätzliche Datenfelder erweitert.

Schritt 8

Im letzten Schritt wird $R_{korrigiert}$ als zuletzt legal betretener Raum R_{legal} definiert und $P_n^{korrigiert}$ dient für die nächste Ortung als Referenzwert (P_{n-1}). $P_n^{korrigiert}$ wird in der Pfadhistorie gespeichert, welcher als Ringpuffer wählbarer Größe realisiert ist. Dies hat den Vorteil, dass die Trajektorie eine maximale Länge hat und zurückliegende Punkte automatisch entfernt werden um einen Speicherüberlauf und Performanceeinbußen zu verhindern. Zusätzlich bleibt die Visualisierung durch eine definierte Pfadlänge übersichtlicher.

4.2.2. Spezialfälle

Der zuvor beschriebene Ablauf des Algorithmus ist für den normalen Betrieb ausgelegt. Spezialfälle wie z. B. die Initialisierung oder die Auflösung eines Deadlocks sind unterschiedlich zu behandeln.

Initialisierung

Die Initialisierungsphase zeichnet sich dadurch aus, dass für jeden neu startenden Ortungspfad zu Beginn noch keine vorhergehenden Messwerte existieren, um eine logische Bewertung durchzuführen. Ziel der Initialisierung soll es sein, einen legalen Startraum und Startpunkt zu definieren, auf deren Grundlage der in Kapitel 4.2.1 beschriebene Algorithmus regulär weiter arbeiten kann. Die Festlegung des Startpunktes kann auf unterschiedliche Weise erfolgen, wobei jeder Ansatz seine Vor- und Nachteile hat.

Man kann nach dem Beginn der Ortung zunächst über mehrere Positionsupdates mitteln und über deren Schwerpunkt den Startwert festlegen. Somit gleicht man den Messfehler des Ortungssystems aus und erhöht die Wahrscheinlichkeit, den tatsächlichen Startraum auszuwählen. Dies ist jedoch nur für eine statische Position sinnvoll und setzt voraus, dass sich die Person zu Beginn der Ortung nicht bewegt und sich das System kalibrieren kann.

Eine weitere Möglichkeit besteht darin, den ersten Ortungspunkt manuell durch den Nutzer eingeben zu lassen. Dies erfordert jedoch einen unverhältnismäßig hohen Aufwand und beinhaltet die Möglichkeit einer fehlerhaften Positionierung z. B. aufgrund von mangelnder Ortskenntnis.

Der naheliegendste Ansatz übernimmt die erste Position des Ortungssystem und legt diese als gegeben und korrekt aus. Dies birgt natürlich die Gefahr, dass ein Raum ausgewählt wird

welcher innerhalb des Streukreises der Ortungspräzision liegt, aber nicht dem tatsächlichen Raum entspricht. Bietet aber den Vorteil, dass er zu einer schnellen und automatisierten Initialisierung führt.

Ein verbesserter Ansatz bewertet den ersten Ortungspunkt entsprechend seiner wahrscheinlichsten tatsächlichen Position. Man kann sich vorstellen, dass die meisten Ortungspfade in der Nähe des Eingangs beginnen (für neu in ein Gebäude eintreffende Personen). Innerhalb eines Gebäudes können dies die Übergänge zu anderen Stockwerken wie beispielsweise Treppenhäuser oder Fahrstühle sein. Sollte die erste Ortung in der Nähe (innerhalb der Toleranzen des Ortungssystems) eines solchen Übergangsbereiches liegen, wird die Position dorthin korrigiert. Diese korrigierende Initialisierung kann jedoch nur erfolgen, wenn diese Übergangsbereiche definiert werden, der erste Ortungspunkt in der Nähe liegt und die Streuung des Ortungssystems bekannt ist.

Für die Simulation und Evaluation sollen vorgefertigte Testszenarien analysiert werden. Diese beginnen nicht zwangsläufig im Eingangsbereich oder an Stockwerksübergängen und müssten extra dorthin verlagert werden um eine automatisierte Initialkorrektur zu erfahren. Daher wird auf diese Funktionalität verzichtet und der erste Ortungspunkt einer neuen Trajektorie als korrekt und gegeben angenommen.

Deadlock

Der in Kapitel 3.2.2 angesprochenen Fall des Deadlocks erfordert ebenfalls eine gesonderte Betrachtung für den Ablauf des Algorithmus. Die Entstehung eines Deadlocks lässt sich dadurch erklären, dass P_{n-1} im Raum R_{legal} liegt, welcher keinen logisch erreichbaren Übergang zu R_{P_n} bietet. Entsprechend des Algorithmus (unabhängig vom gewählten Korrekturmodell) wird $P_n^{korrigiert}$ nach R_{legal} verschoben und dort die kürzeste Distanz zu P_n bestimmt. Die Raumgeometrie von R_{legal} ist hierbei der begrenzende Faktor, wodurch sich die Punkte P_n und $P_n^{korrigiert}$ nach einer fehlerhaften Raumwahl im ungünstigen Fall beliebig weit voneinander entfernen würden.

Eine selbstständige Erholung aus einem Fehlerzustand (siehe Abb. 3.6) soll automatisch erfolgen, wenn der Abstand zwischen dem korrigierten Ortungspunkt $P_n^{korrigiert}$ und dem aktuellen unkorrigierten Punkt P_n einen vorgegebenen maximalen Abstand überschreitet. Dieser Abstand sollte so gewählt werden, dass er um ein Vielfaches größer ist als die Toleranz des Ortungssystems um einen Deadlock eindeutig feststellen zu können, ohne eine noch laufende Korrektur vorschnell zu unterbrechen.

4.2.3. Berechnung der Wahrscheinlichkeiten

Für die Ausführungen zur Berechnung der Wahrscheinlichkeiten werden einige Begriffe verwendet, die einer kurzen Erklärung bedürfen.

Wahrscheinlichkeit

Für jeden Raum wird die Wahrscheinlichkeit (WLK) in einer Liste gespeichert. Diese stellt die Annahme dar, mit welcher der Raum als *betreten* gilt. Zusätzlich zur physikalischen Erreichbarkeit des Raumes wird die Wahrscheinlichkeit als weiterer Bewertungsfaktor bei einem Raumwechsel herangezogen.

Pendeln

Als Pendeln versteht man, dass sich die tatsächliche Position in Raum A in der Nähe einer Tür befindet und ein Teil der Ortungspunkte aufgrund der Toleranzen in Raum B auf der anderen Seite der Tür liegen. Physikalisch ist dieser Raum entsprechend des Bewegungsmodells erreichbar und würde den neuen Punkt in Raum B umgehend akzeptieren. Sollte der nächste Ortungspunkt wieder in Raum A usw. sein, so kann ein ständiger Raumwechsel die Folge sein.

Parent-Raum

Als Parent-Raum bezeichnet man den hierarchischen Ausgangsraum einer rekursiven Erreichbarkeitssuche.

aktiv

Dieses zusätzliche Raumattribut signalisiert, dass dieser Raum bei einer zurückliegenden mehrdeutigen Pfadentwicklung ein erreichbarer Kandidat war und für eine Wahrscheinlichkeitsbetrachtung in Frage kommt.

Startboost

Um einen logisch korrekten Raumwechsel auch anhand dessen räumlicher Distanz zu bewerten, erhalten die Räume initial bei der rekursiven Erreichbarkeitssuche eine anteilige Wahrscheinlichkeit unabhängig davon, ob sie betreten wurden.

In der Analyse wurde unter anderem das Thema der mehrdeutigen Pfadentwicklung (siehe Abb. 3.7 und 3.8) angesprochen. Dies hängt sehr stark mit der Entscheidung für einen Raumwechsel (vgl. Abb. 3.9) zusammen. Um ein ungewolltes *Pendeln* zwischen erreichbaren Nachbarräumen zu minimieren, wird zusätzlich eine Wahrscheinlichkeitsbetrachtung vorgenommen anhand derer sporadische Ausreißer gemittelt werden sollen.

4. Design

Sollte eine Ortung in einem angrenzenden Raum liegen und dieser logisch erreichbar sein, so findet ein Raumwechsel erst dann statt, wenn die Wahrscheinlichkeit für diesen Raum einen vorgegebenen Wert überschritten hat und verbleibt ansonsten in dem zuletzt legal betretenen Raum (vgl. die Weiterentwicklung des Pfades von A in Richtung B in Abbildung 3.10). Die Berechnung der Wahrscheinlichkeiten für die unterschiedlichen Räume wird von mehreren Parametern beeinflusst.

Da aufgrund der Toleranzen des Ortungssystems mehrere benachbarte Räume für einen Ortungspunkt in Frage kommen können, werden bei einem festgestellten Raumwechsel die erreichbaren Räume ausgehend vom Parent-Raum des Punktes P_{n-1} aufgelistet und als *aktiv* markiert. Für den Fall, dass diese Räume bisher eine Bewertung von 0 hatten, erhalten sie einen Startboost von maximal der Hälfte der aktuellen Bewertung des Parent-Raumes sowie einen anteiligen Abzug, je nach Entfernung von Tür zu Tür. Dies soll näher liegende Räume entsprechend höher bewerten (siehe Abb. 4.3).

Hier erhält Raum 1102 eine WLK von 48, R1101c und R111101 jeweils 38 und noch weiter entfernte Räume eine stetig abnehmende Bewertung.

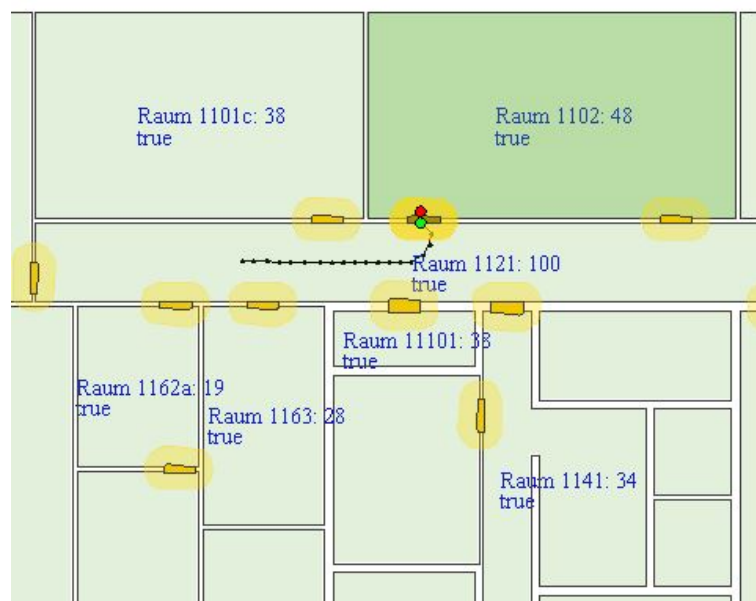


Abbildung 4.3.: Verteilung der WLK bei einem erkannten Raumwechsel

Alle erreichbaren Räume von P_{n-1} werden nach Entfernung aufsteigend sortiert aufgelistet, wobei der erste Eintrag der am nächsten gelegenen Tür entspricht und als Referenzwert

(*reference*) dient, welcher keinen Abzug für die Bewertung bekommen soll. Erst alle weiter entfernten Türen erhalten einen anteiligen Abzug ($factor = 1/reference$).

Für jeden erreichbaren Raum (entsprechend des Bewegungsmodells) wird der Startboost nach der Formel $x = parent/2 - factor * distance$ berechnet. Die WLK x setzt sich zusammen aus der Hälfte der WLK des parent-Startraumes, abzüglich der Entfernung zur jeweiligen Tür (*distance*), multipliziert mit dem zuvor berechneten Faktor, um ein relatives Verhältnis zu erhalten. Die höchste zu erreichende Bewertung beträgt 49 ($100/2 - 1$) und soll anderen Räumen die Möglichkeit geben sich zu entwickeln, bevor eine Entscheidung für einen Raumwechsel getroffen wird. Die Bewertung wird dem Raum jedoch nur gutgeschrieben, wenn dieser zuvor eine WLK von 0 hatte und einen Startboost benötigt.

Räume die nicht *aktiv* gesetzt sind erhalten automatisch eine 0-Bewertung, da sie logisch nicht erreichbar sind. Die Mechanismen, damit ein Raum vom aktiven in den inaktiven Zustand wechselt, hängen von dem verwendeten Algorithmus ab und werden in Kapitel [4.2.4](#) und [4.2.5](#) gesondert betrachtet.

Zusätzlich zur Wahrscheinlichkeitsberechnung bei einem Raumwechsel werden bei jedem Positionsupdate des Ortungssystems die Wahrscheinlichkeiten ebenfalls neu berechnet. Beeinflussende Faktoren sind hauptsächlich der zuletzt legal betretene Raum sowie die aktuelle Position. Über die aktuelle Position lässt sich der aktuell betretene Raum bestimmen und sich die anderen (nicht betretenen) Räumen herleiten. Die Wahrscheinlichkeit für jeden *aktiven* und *nicht betretenen* Raum wird verringert (der Raum R_{legal} behält immer eine minimale Restpunktzahl, bis ein neuer plausibler Raum gefunden wurde). Und im Gegenzug wird die WLK für den *betretenen* Raum erhöht, sofern dieser aktiv ist. Mit Hilfe dieser Funktionalität sollen wiederholte Ortungen innerhalb eines Raumes dessen Wahrscheinlichkeit erhöhen und die Hypothese bestärken, dass sich die tatsächliche Position dort befindet. Besonderheiten für das Zurückhalten von verändernden Werten werden in den folgenden beiden Kapiteln unterschiedlich gehandhabt, da sie eng mit der korrigierenden Algorithmus-Phase zusammenhängen.

4.2.4. Restriktiver Ansatz

Der restriktive Ansatz für den eigenen Algorithmus versucht einen möglichst durchgängigen und logisch nachvollziehbaren Pfad zu erstellen. Dies hat zur Folge, dass eine einmal getroffene Entscheidungen für eine mögliche Raumentwicklung im Nachhinein nicht mehr geändert werden kann und im ungünstigen Fall zu einem Deadlock führen kann. Der Vorteil besteht jedoch darin, dass der resultierende Pfad einer dem Menschen möglichen Trajektorie folgt (ohne Wände zu durchkreuzen oder Sprünge aufzuweisen) und diese mit Bewegungsmustern abgeglichen werden kann.

Die hauptsächlichsten Unterschiede im Verhalten des Algorithmus treten im Bereich der Korrektur (siehe Schritt 6 in Abb. 4.1) und der Berechnung der Wahrscheinlichkeiten auf. Die Entscheidung für einen Raumwechsel basiert auf der logischen Erreichbarkeit anhand des Bewegungsmodells und der aktuellen Wahrscheinlichkeit des zu betretenden Raumes. Sollten alle Faktoren für einen Wechsel sprechen, wird $P_n^{korrigiert}$ in dem neuen Raum verortet. Eine nachträgliche Umentscheidung für einen nahe gelegenen Raum ist nicht mehr möglich, da dieser logisch von $P_n^{korrigiert}$ nicht mehr zu erreichen ist, und die Trajektorie ansonsten durch die Wand verlaufen würde.

Das Korrekturverhalten lässt sich anhand der Grafik 3.7 verdeutlichen. Der zuletzt legal betretene Raum R_{legal} ist der langgestreckte Flur, in dem die letzten Ortungspunkte in Richtung der Türen A und B verlaufen. Da die letzte Ortung im Flur in erreichbarer Nähe zu den Räumen A und B liegt, werden beide *aktiv* gesetzt, da sie logisch erreichbar sind. Die Berechnung der Wahrscheinlichkeiten für den Startboost ergeben ähnliche Werte, da sie nahezu gleich weit entfernt liegen. Sollte der nächste Ortungspunkt in Raum B verortet werden, würde anhand der bisherigen Bewertungsfaktoren sofort ein Raumwechsel stattfinden. Wie in dem Beispiel zu sehen, kann die folgende Position jedoch wieder in Raum A liegen, da sich die tatsächliche Position in der Nähe der Wand befindet und aufgrund der Toleranzen des Ortungssystems beide Räume abdecken kann. Es wäre daher verfrüht, gleich beim ersten Ortungspunkt in einen der beiden Räume zu wechseln. Sondern erst bei einer eindeutigen Raumentwicklung den Wechsel zuzulassen und ansonsten die Position im Flur zwischen den beiden Türen zurück zu halten.

Um das gewünschte Verhalten der verzögerten Entscheidung zu erzielen, werden zunächst alle Räume im Umkreis von *WALLRADIUS*-Metern aufgelistet. *WALLRADIUS* definiert den zu untersuchenden Abstand zu Wänden und sollte entsprechend der Toleranzen des Ortungssystems gewählt werden, damit benachbarte Räume bei einer unpräzisen Ortung mit beachtet werden.

Im Ablaufdiagramm 4.4 wird anschließend die Wahrscheinlichkeitsverteilung für jeden neu eintreffenden Ortungspunkt berechnet. Sollten sich mehrere Räume in der Liste *roomDistances* befinden, so bleiben die Bewertungen unverändert (Knoten 1 - 3). Wenn sich nur ein Raum im besagten Umkreis befindet (Knoten 1 + 4), wird für den Fall des aktuell betretenen Raumes die Wahrscheinlichkeit um den Wert *ADDVALUEINSIDEROOM* erhöht (Knoten 9) und auf einen Maximalwert von 100 gedeckelt (Knoten 10). Für alle nicht betretenen Räume wird pro Positionsupdate ein konstanter Wert *SUBVALUEOUTSIDEROOM* abgezogen (Knoten 5) und auf minimal 0 begrenzt (Knoten 7). Der zuletzt legal betretene Raum behält jedoch immer

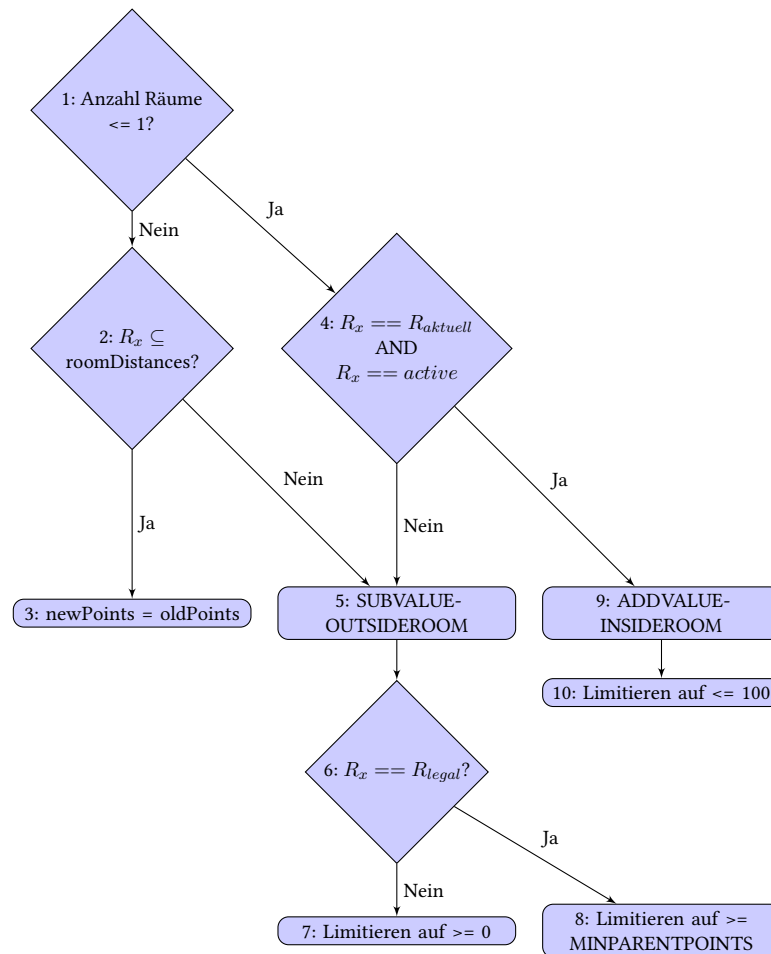


Abbildung 4.4.: Berechnung der Wahrscheinlichkeiten für den restriktiven Algorithmus

eine Mindestbewertung von *MINPARENTPOINTS* (Knoten 6 + 8), bis dieser von einem neuen (wahrscheinlicheren) Raum abgelöst wird.

Diese Funktionalität sorgt dafür, dass ein betretener und aktiver Raum mit der Zeit immer wahrscheinlicher wird und andere Räume über die Zeit als unwahrscheinlich eingestuft werden und beim Erreichen einer WLK von 0 automatisch auf *inaktiv* gesetzt werden. Um den zuletzt legal betretenen Raum vor der Deaktivierung zu schützen, wird dieser auf einem Minimalwert gehalten, bis ein Raumwechsel auf Grundlage der physikalischen Erreichbarkeit und der Wahrscheinlichkeitsbetrachtung erfolgt.

4.2.5. Fehlertoleranter Ansatz

Der fehlertolerante Ansatz für den Algorithmus ist darauf ausgelegt, mehrere mögliche Raumentwicklungen parallel zu betrachten und entsprechend der aktuell höchsten Wahrscheinlichkeit den momentanen Aufenthaltsort zu definieren. Dies führt unter Umständen zu Sprüngen zwischen den Räumen, obwohl kein legaler Raumübergang zu dieser Zeit erreichbar ist (jedoch in der Historie zuvor einmal möglich war). Da es sich um einen Online-Korrektur-Algorithmus handelt, welcher Ortungsinformationen zur Laufzeit korrigiert ohne die zukünftige Pfadentwicklung zu kennen, stellt dieser Ansatz einen Kompromiss zwischen Korrektheit und Fehlertoleranz dar. Die Details werden anhand des Ablaufdiagramms 4.5 erläutert.

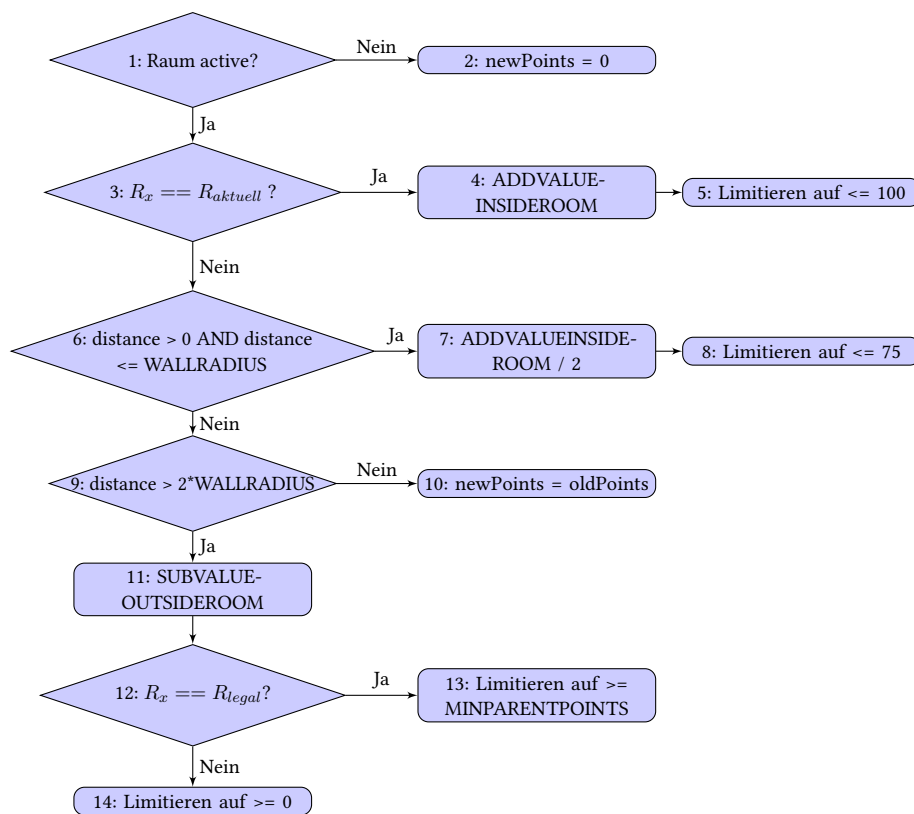


Abbildung 4.5.: Berechnung der Wahrscheinlichkeiten für den fehlertoleranten Algorithmus

Beim Eintreffen eines Ortungspunktes wird für jeden Raum die Wahrscheinlichkeit neu berechnet. Ähnlich wie bei dem *restriktiven Verfahren* erfolgt nur eine Bewertung für *aktive* Räume und wird ansonsten auf 0 gesetzt (Knoten 1 + 2). Im nächsten Durchlauf werden Räume mit einer WLK von 0 automatisch deaktiviert und entfallen als mögliche Alternativräume. Die

Berechnung für den aktuell betretenen Raum verläuft ebenfalls auf gleiche Weise und wird mit *ADDVALUEINSIDEROOM* bestärkt und begrenzt (Knoten 3 - 5).

Der Unterschied betrifft die Bewertung *nicht betretener Räume*. Für jeden Raum wird eine Position berechnet, welche am nächsten zu P_n liegt und deren Distanz zueinander raumspezifisch gespeichert wird.

Wenn der Abstand weniger als *WALLRADIUS* beträgt, wird die neue additive Wahrscheinlichkeit um den Faktor 2 reduziert hochgezählt, bis zu einem Maximalwert von 75 (Knoten 6 - 8). Für eine Distanz von mehr als beispielsweise 1 m (doppelter *WALLRADIUS*) wird die WLK heruntergezählt (Knoten 9 + 11), wobei wiederum der zuletzt legal betretene Raum nur bis zu einem Minimalwert reduziert wird (Knoten 12 + 13). Für den dazwischen liegenden Abstand bleibt die Bewertung unverändert (Knoten 10).

Die Wahrscheinlichkeit für den aktuell betretenen Raum zählt kontinuierlich hoch, während in der Nähe liegende (aktive) Räume ebenfalls einen abgeschwächten Bonus erhalten. Damit wird erreicht, dass eine mögliche parallel verlaufende Trajektorie im Nachbarraum als zusätzliche Alternative beachtet wird und die weitere Pfadentwicklung fortsetzen kann, wenn sie im aktuell legal betretenen Raum auf begrenzende Wände trifft. Je weiter die alternativen Räume entfernt liegen, desto unwahrscheinlicher werden sie für eine plausible Pfadentwicklung, was sich auf deren Wahrscheinlichkeit auswirkt. Anhand der aktuellen Wahrscheinlichkeitsverteilung erfolgt eine Korrektur zu dem am höchsten bewerteten Raum, ohne nochmals die physikalische Erreichbarkeit in diesem Moment zu überprüfen, da sie auf Grundlage des Algorithmus zuvor schon durch die Aktivierung als mögliche Raumentwicklung identifiziert wurde.

Das Beispiel in Abbildung 4.6 verdeutlicht das Korrekturverhalten. Die ursprüngliche Trajektorie (rot) verläuft vom Flur *A* in den Raum *B* (in der Nähe zur Wand *C*) und kreuzt die Wand zu Raum *D* um im Flur *F* zu enden. Dieser Pfadverlauf ist offensichtlich nicht korrekt, da kein physikalischer Übergang von *B* nach *D* existiert. Der Flurwechsel muss demnach über die Räume *C* und *E* stattgefunden haben. Zu Beginn der Korrektur sind alle Räume *inaktiv* bis auf den Flur *A*, der somit auch die höchste Wahrscheinlichkeit besitzt. Beim ersten Ortungspunkt in Raum *B* werden die erreichbaren Räume von der letzten legalen Position (P_{n-1} in Raum *A*) aufgelistet und auf *aktiv* gesetzt (Räume *B* und *C*), aufgrund des zuvor beschriebenen Startboostes wechselt $P_n^{korrigiert}$ zeitnah in Raum *B*. Zeitgleich nimmt die Wahrscheinlichkeit für *C* ebenfalls (aber nicht so schnell wie für *B*) zu, da P_n in der Nähe der Trennwand liegt. Sobald die Ortungspunkte in den Raum *D* wandern, nimmt die Wahrscheinlichkeit für *B* ab, die korrigierte Position verbleibt aber noch so lange dort (Position 1), bis sie von der höheren Wahrscheinlichkeit des Raumes *C* abgelöst wird und ein Sprung von Position 1 nach 2

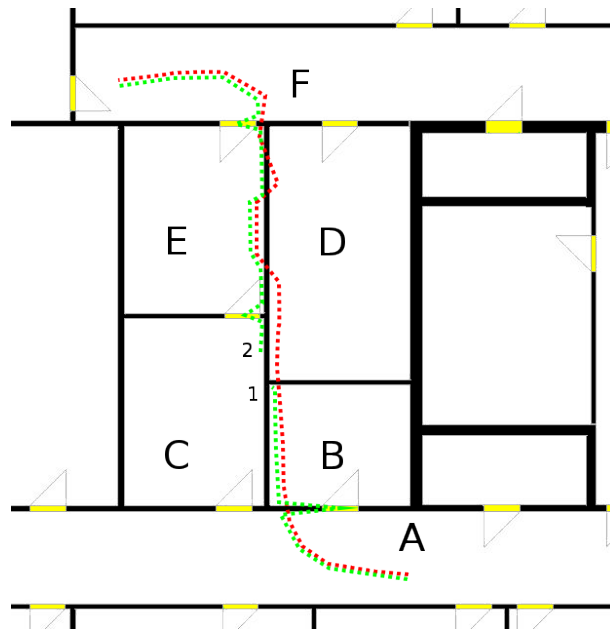


Abbildung 4.6.: Fehlertoleranter Ansatz mit Raumsprüngen

stattfindet. C erhält keinen Abzug, da die am nächsten gelegene Position des Raumes höher wandern kann als in Raum B und noch innerhalb des WALLRADIUS an P_n liegt. Sobald C die höchste Wahrscheinlichkeit besitzt, wird der Raum C als R_{legal} angenommen und $P_n^{korrigiert}$ entsprechend nahe zu P_n verschoben. Der weitere Verlauf über E nach F entspricht wieder dem generellen Regelverhalten, da Raum D nicht aktiv ist und somit für eine alternative Pfadentwicklung nicht in Frage kommt.

Anhand dieses Ansatzes kann zwar nicht verhindert werden, dass sich die korrigierte Position zunächst für den falschen Raum entscheidet (dies ist auf Grundlage der bisher zurückgelegten Trajektorie auch gar nicht möglich). Es wird aber verhindert, dass $P_n^{korrigiert}$ in dem einmal fälschlicherweise angenommenen Raum hängen bleibt (Deadlock), indem der Fehlerzustand durch einen Sprung in eine parallele Raumentwicklung aufgelöst wird.

4.3. Architektur und Integration

In diesem Kapitel wird die Architektur sowie die Integration in bestehende Systeme erläutert. Da der Algorithmus zunächst Ortungsinformationen benötigt, die in einem Gebäude abgebildet werden müssen und relative Informationen bezüglich Wänden und Türen liefern soll, wird ein unterstützendes Framework verwendet, in welches der Algorithmus integriert werden

kann. Als Grundlage hierfür bietet sich eine bereits erprobte Blackboard-Architektur des *Living Place* der HAW an, welche in Kapitel 4.3.1 näher erläutert wird. Darauf aufbauend wird der *Indoor Spatial Information Service* (ISIS) zur Bereitstellung der zur Berechnung benötigten Informationen verwendet (siehe Kap. 4.3.2). Die angepassten Komponenten und Schnittstellen werden in Kapitel 4.3.3 im Speziellen aufgeschlüsselt und das Gebäudemodell mit seinen strukturellen Anfragen in Kapitel 4.3.4 behandelt. Abschließend folgt die Erläuterung zur Visualisierung (vgl. Kap. 4.3.5), mit deren Hilfe die Korrekturen im Modell veranschaulicht werden.

4.3.1. Living Place

Das *Living Place* der HAW Hamburg ist ein Smart Home zur Erprobung unterschiedlichster Technologien und Neuentwicklungen. Als Grundlage zur einheitlichen Kommunikation der einzelnen Teilsysteme dient eine Blackboard-Architektur, welche Nachrichten über ein Publish-Subscriber-Modell austauscht. Im Speziellen wird ActiveMQ ([Apache.org \(2013\)](#)) in Verbindung mit MongoDB ([10gen, Inc. \(2013\)](#)) zur Übertragung und persistenten Speicherung der Nachrichten verwendet. Alle Sensordaten (z. B. Positionsupdates des Ortungssystems) stehen über ihre jeweiligen Topics zur Verfügung und können von Anwendungen zur Weiterverarbeitung abgerufen werden.

Diese erprobte Anwendungsarchitektur und das Knowhow sollen als Grundlage für die Testumgebung der eigenen Realisierung dienen. Im Falle der Indoor-Ortung wird ein simuliertes Ortungssystem sowie eine lokal ausgeführte Blackboard-Kommunikation verwendet, um eine kontrollierbare Umgebung zu schaffen.

4.3.2. ISIS

Der *Indoor Spatial Information Service* (ISIS) wurde im Rahmen einer Masterarbeit von [Karstaedt \(2012\)](#) entwickelt und bietet eine „Kontextinterpretation in Smart Homes auf Basis semantischer 3D Gebäudemodelle“. Die Ortungsdaten werden vom ISIS empfangen, im internen Gebäudemodell abgebildet, relationale Anfragen bearbeitet und an die Umwelt mitgeteilt (siehe Abb. 4.7). Zusätzlich besteht die Möglichkeit, automatisiert Nachrichten zu verschicken, sobald zuvor festgelegte Events auftreten, wie z. B. das Betreten eines Raumes oder Türbereiches.

Die Softwarearchitektur lässt sich als *Service-orientierte-Schichten-Architektur* charakterisieren, deren Komponenten in Abbildung 4.8 aufgegliedert sind und von der Visualisierung, über die Kommunikation bis hin zur Persistenz reichen.

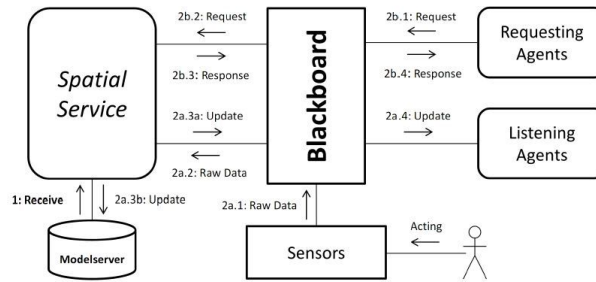


Abbildung 4.7.: ISIS-Kommunikationsdiagramm Karstaedt (2012)

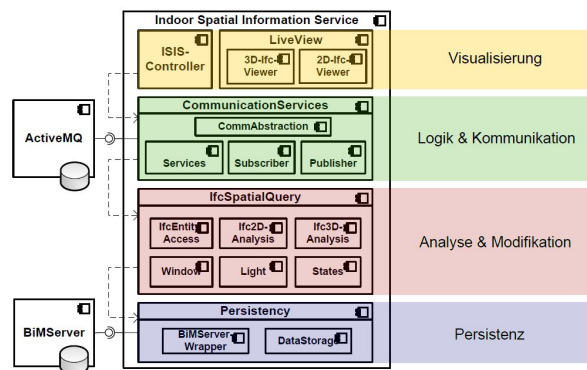


Abbildung 4.8.: ISIS-Architekturschaubild Karstaedt (2012)

Für die eigene Realisierung bietet es sich an, dieses bestehende System zu nutzen und um zusätzliche Funktionalitäten im *ISIS-Controller* zu erweitern. Die Integration findet an mehreren Stellen statt und betrifft einerseits die Kommunikation (um die korrigierten Ortungsdaten über ein separates Topic an die Umwelt mitzuteilen). Andererseits muss auch die Analyse-schicht angepasst werden, um neue Überprüfungsmethoden bezüglich des Gebäudemodells zu bieten. So muss z. B. ein Punkt innerhalb eines Raumes berechnet werden, welcher am nächsten zu einem äußeren Referenzpunkt liegt, um eine korrigierte Position zu erhalten. Die Algorithmen werden als modular auswählbares Element in die Software integriert und sind über Konfigurationsdateien in ihrem Regelverhalten anpassbar.

Das Zusammenspiel der im ISIS verwendeten Tools und Bibliotheken wird in Abbildung 4.9 dargestellt. Der *BIMServer* liefert das Gebäudemodell im IFC⁴-Format, welches im nächsten Schritt durch die *Open IFC Tools* der **Bauhaus-Universität Weimar** (2013) in eine zugreifbare Struktur umgewandelt wird. In dem hier verwendeten 2D-Fall wird es an die *Java Topology*

⁴Industry Foundation Classes (Objektbasiertes Dateiformat)

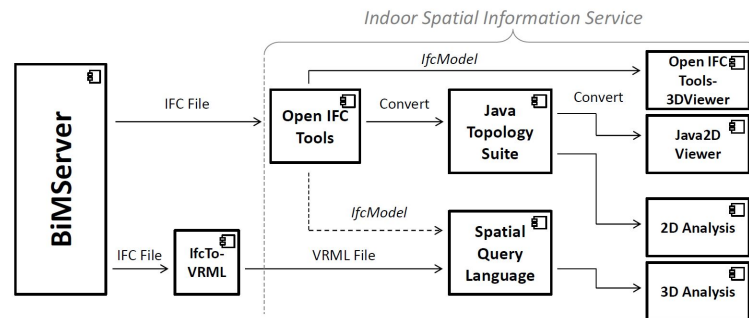


Abbildung 4.9.: Zusammenspiel der Tools und Bibliotheken Karstaedt (2012)

Suite (JTS) von Vividsolutions (2013) weitergeleitet, welches topologische Berechnungen auf Grundlage eines Gebäudemodells zulässt.

4.3.3. Komponenten und Schnittstellen

In diesem Kapitel werden die Komponenten und Schnittstellen der eigenen Realisierung vorgestellt. Der Korrekturalgorithmus wird als optional auswählbares Modul in das bestehende ISIS-System integriert. Das Modul ist hauptsächlich in der Analyseschicht der ISIS-Architektur verankert (siehe Abb. 4.8, Ifc2D-Analysis). Jedes Positionsupdate wird in einem Update-Handler überprüft und an die Analyseschicht weitergeleitet. Je nachdem ob eine Korrektur gewünscht ist, erfolgt ein geschlossener Funktionsaufruf, der als Übergabeparameter die ursprüngliche Position enthält und ein Antwortobjekt mit der korrigierten Position zurück liefert. Die Parametrisierung der einzelnen Werte zur Einstellung des Regelverhaltens erfolgt mittels einer Konfigurationsdatei und erlaubt schnelle Anpassungen an zentraler Stelle (z. B. die Werte für ADDVALUEINSIDEROOM oder WALLRADIUS).

Die Kommunikation erfolgt über einen Subscriber auf dem Topic für die Ortungsdaten und veröffentlicht die korrigierten Positionen über einen Publisher auf einem separaten Topic (vgl. Abb. 4.8, Architekturebene *Logik und Kommunikation*). Externen Anwendungen steht es somit frei, ob sie eine Korrektur wünschen und sich am entsprechenden Topic anmelden um ihre Ortungsdaten zu erhalten. Aufgrund der *Nicht-funktionalen Anforderungen* (siehe Kap. 3.2.3) ist es jedoch auch möglich, dass Anwendungen das Korrektur-Topic abonnieren und die Ortungsdaten empfangen, als ob es sich um das ursprüngliche Ortungssystem handeln würde. Somit sind keine Anpassungen seitens der Anwendung nötig, um von der Korrektur zu profitieren. Es besteht aber die Möglichkeit, detaillierte Informationen aus der Nachricht zu extrahieren. Der Aufbau der über das Topic gesendeten Nachrichten orientiert sich am Format

4. Design

der ursprünglichen Positionsupdates und erweitert diese um zusätzliche Felder (siehe Tabelle 4.1).

Ursprüngliches Positionsupdate	Korrigiertes Positionsupdate
ID	ID
Maßeinheit	Maßeinheit
Neue Position (x-,y-,z-Koordinate)	Neue <i>korrigierte</i> Position (x-,y-,z-Koordinate)
	Indikator ob die Position korrigiert wurde
	Wahrscheinlichkeit der neuen Position
	Alte <i>unkorrigierte</i> Position

Tabelle 4.1.: Vergleich der Nachrichten eines Positionsupdates

Anstatt eines echten Ortungssystems wird ein Mockup verwendet, welches es ermöglicht die Position mittels eines virtuellen Grundrisses zu bestimmen. Das Mockup wurde für *Living Place*-interne Tests von **Karstaedt** entwickelt und muss für die eigene Entwicklung angepasst werden. Dies betrifft die Umstellung auf den Grundriss des 11. Stockwerks mit entsprechender Skalierung, Kalibrierung des Koordinatenursprungs und Darstellung sowie funktionalen Erweiterungen, um die Nutzerfreundlichkeit zu verbessern. In Abbildung 4.10 ist die GUI des Mockups zu sehen, während ein neuer *Tag* erstellt wird.

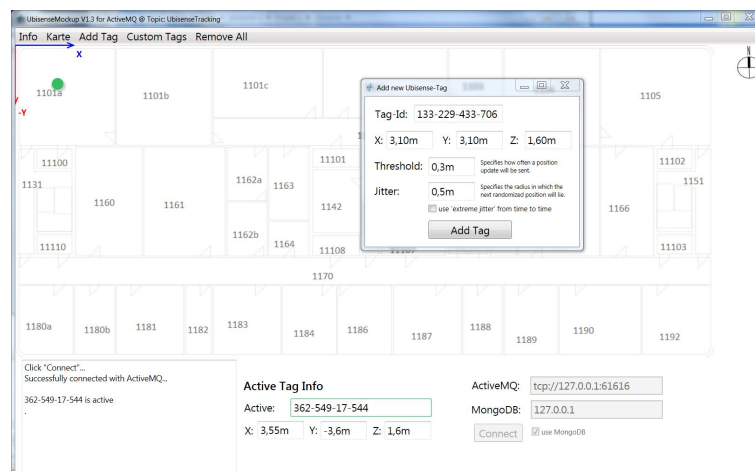


Abbildung 4.10.: Publisher Mockup des Ortungssystems für den 11. Stock

Jeder *Tag* enthält eine eindeutige ID und lässt sich mittels der Maus frei auf dem Grundriss bewegen, um unterschiedliche Testszenarios vorzugeben. Je nach Wunsch kann ein beliebiger Jitter eingestellt werden, welcher dafür sorgt, dass die übertragene Position gegenüber der tatsächlichen Position um einen zufälligen Wert innerhalb des Streubereichs schwankt. Der

Jitter erzeugt eine zufällige Abweichung in x- und y-Richtung im Bereich von 0 bis x Metern, wobei x für den maximalen Streuradius steht. Dies soll die Messtoleranzen und Störungen eines echten Ortungssystems emulieren. Zusätzlich besteht die Möglichkeit *extreme Ausreißer* von mehreren Metern sporadisch einfließen zu lassen, um das Regelverhalten für solche Sonderfälle zu überprüfen (vgl. Kap. 2.1.1).

4.3.4. Gebäudemodellierung

Das im ISIS verwendete Gebäudemodell ermöglicht topologische Berechnungen und muss von externer Seite geliefert werden. Bisher existierte nur ein 3D-Gebäudemodell des *Living Place* für ISIS. Dieses bietet aufgrund seines kleinen Grundrisses keine parallel verlaufenden Alternativrouten und zu wenig restriktive Elemente (Wände), um ausreichend Testszenarien für die Korrekturalgorithmen zu liefern (vgl. Abb. 4.11).

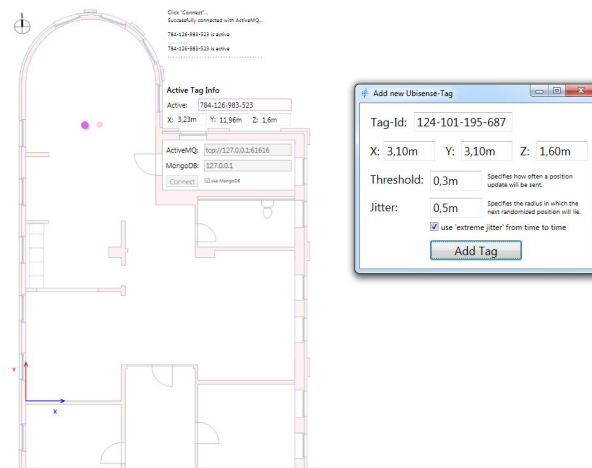


Abbildung 4.11.: Unzureichende restriktive Elemente im *Living Place Karstaedt* (2012)

Aus diesem Grund wird der 11. Stock des TI-Gebäudes der HAW (BT7) ausgewählt, da dieser dem Aufbau eines Bürogebäudes eher entspricht. Verhältnismäßig schmale Gänge, viele (teilweise untereinander verbundene) angrenzende Räume und eng definierte Übergänge zwischen den Räumen bieten eine gute Grundlage für die Plausibilitätsüberprüfung. Da bisher noch kein 3D-Modell im passenden Format verfügbar ist, wird aus CAD-Grundrissdaten mit Hilfe von *Revit Architecture* (Autodesk (2012)) ein Gebäudemodell konstruiert (vgl. Abb. 4.12). Ein besonderes Augenmerk liegt auf der Modellierung von Wänden, Türen und eindeutigen Rauminformationen.

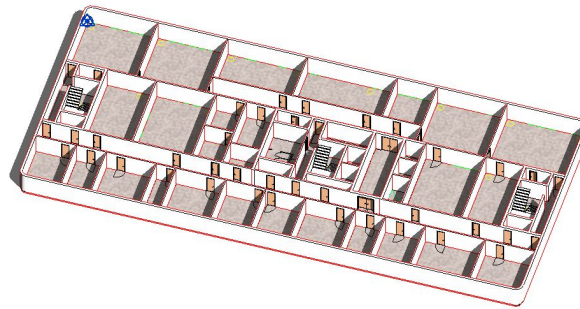


Abbildung 4.12.: 3D-Modellierung des 11. Stocks

Nach einem Export in das IFC 2x3 Format kann das Modell auf dem BIM-Server gespeichert werden und steht dem ISIS zur Auswertung zur Verfügung. Über die integrierten Bibliotheken *Open IFC Tools* und *JTS* können topologische Anfragen zu einer Koordinate beantwortet werden. Wichtig hierbei ist die Modellierung der sogenannten eindeutigen *Spaces*, welche Räume als eine Entität definieren und einen betretenen Raum leicht identifizierbar machen. Die angrenzenden Wände und Türen können als Polygone oder auch als hierarchische Liste zurückgeliefert und für Abstandsberechnungen genutzt werden.

4.3.5. Visualisierung

Die Visualisierung erfolgt über einen angepassten 2D-Viewer im ISIS und verwendet den *Open IFC Tools Viewer*. Beim Eintreffen eines Ortungspunktes über den Subscriber wird die aktualisierte Position des *Tags* mittels eines Observer-Patterns an die Visualisierungsebene mitgeteilt. Die Update-Funktion des Viewers überprüft, ob es sich um ein Antwortobjekt des Korrektur-Algorithmus handelt, verschiebt den *Tag* und fügt die aktuelle Position zu einer Historie hinzu. Die Historie wird in einem Ringpuffer einstellbarer Größe gespeichert, um die maximale Länge der zurückgelegten Trajektorie automatisch zu begrenzen und die Übersichtlichkeit zu erhalten. Der GUI-Manager stellt im Anschluss den zurückgelegten Pfad dar, indem die einzelnen Positionen aus der Historie untereinander verbunden werden. Sollte auf Seiten des Korrektur-Algorithmus der logische Pfad unterbrochen werden⁵, so wird dies im Antwortobjekt kenntlich gemacht und auch die visuelle Repräsentation des Pfades unterbrochen. Bei einem legalen Raumwechsel wird zusätzlich zu der korrigierten Position auch der logische dazwischen liegende Pfad übertragen. Dieser setzt sich aus den Türschwerpunkten der traversierten Räume zusammen und verhindert das Durchkreuzen von Wänden um den tatsächlich

⁵Erholung aus einem Fehlerzustand, oder Sprung in einen wahrscheinlicheren Raum (siehe Kap. 4.2.5)

zurückgelegten Pfad zu verdeutlichen (siehe Abb. 4.13, von Raum 1101c, über die Tür zu Raum 1121 und dort über die Tür zu Raum 1102).

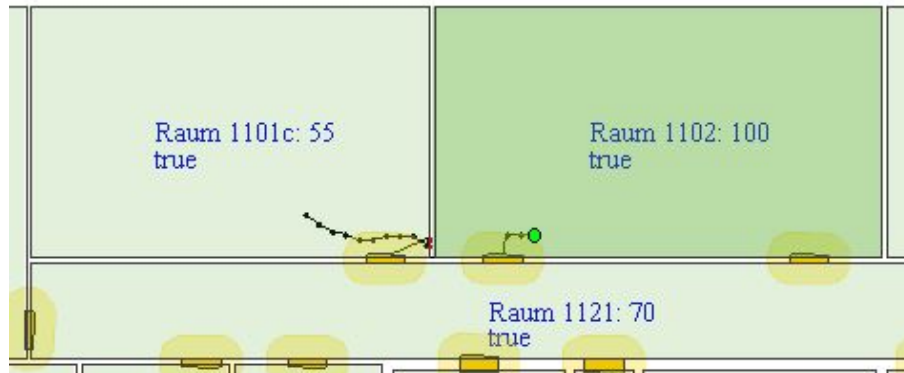


Abbildung 4.13.: Visualisierung eines logischen Raumwechsels über die Türschwerpunkte

Das Antwortobjekt enthält zusätzlich eine Liste mit allen bereits überprüften Räumen und deren Status, ob sie *aktiv* sind sowie deren aktuelle Wahrscheinlichkeit. Diese Liste wird zusätzlich pro Update in der GUI aktualisiert und erleichtert neben den Konsolenlogs die visuelle Überprüfung der Korrekturentscheidungen des Algorithmus.

4.4. Fazit

Abschließend wird für das Design-Kapitel ein Fazit gezogen.

Die Realisierung erfordert neben der Entwicklung der Korrekturlogik auch mehrere unterstützende Strukturen.

Für die Überprüfung der Algorithmen ist es nötig, Testfälle bestehend aus mehreren Ortungspositionen zu simulieren. Dies erfolgt durch ein angepasstes Publisher-Mockup mit einer eigens dazu überarbeiteten GUI und entsprechender Kalibrierung.

Neben den Ortungsdaten ist ein Gebäudemodell für die logische Überprüfung auf Korrektheit nötig. Hierfür wird ein nutzbarer Grundriss mit einer repräsentativen Raumaufteilung gewählt und daraus ein 3D-Gebäudemodell konstruiert.

Zur Verwaltung der Ortungsdaten, der Visualisierung und des Zugriffs auf die Elemente des Gebäudemodells für topologische Anfragen, wird ein bestehendes Framework (ISIS) verwendet. Dafür wird es um zusätzliche Funktionalitäten erweitert und den eigenen Ansprüchen angepasst.

Auf dieser Grundlage können die Algorithmen aufgebaut werden. Aufgrund der Tatsache, dass

es sich um ein Online-Korrekturverfahren handelt, sind zur Laufzeit nicht alle Ortungspositionen bekannt. Die Korrektur wird aber schon von Beginn an gefordert und die Entscheidung für einen Korrekturverlauf kann nachträglich nicht mehr revidiert werden. Der *restriktive* Algorithmus verfolgt den Ansatz, dass die korrigierten Ortungspunkte in ihrer Gesamtheit eine logisch nachvollziehbare und durchgängige Trajektorie durch das Gebäude darstellen und somit einer höher stehenden Auswertungslogik entsprechend, aufbereitetes Material zur Verfügung stellen. Bei einer mehrdeutigen Raumentwicklung wird die Entscheidung bis zur Erkennung einer deutlichen Tendenz verzögert.

Der *fehlertolerante* Algorithmus basiert zwar auch auf der logischen Erreichbarkeit von Räumen, favorisiert jedoch den wahrscheinlichsten Raum als aktuell betretenen Raum. Somit können sich mehrere parallele Bewegungspfade durch Räume entwickeln und die korrigierte Position springt zum jeweils am höchsten bewerteten Raum.

Die korrekte Funktionsweise der Algorithmen gilt es im folgenden Kapitel zu evaluieren.

5. Evaluation

Im nachfolgenden Kapitel werden die Ergebnisse der eigenen Entwicklung evaluiert. Hierfür erfolgt in Abschnitt 5.1 zunächst eine Darlegung des Testaufbaus und der Vorgehensweise. Im darauf folgenden Kapitel 5.2 wird eine Auswertung entsprechend der zuvor festgelegten Szenarien des *restriktiven* und *fehlertoleranten Algorithmus* vorgenommen. Im Anschluss werden in Kapitel 5.3 die Algorithmen zusammenfassend verglichen und ein Fazit gezogen.

Die Bewertung erfolgt anhand eines Vergleichs zwischen der tatsächlichen (P_t), der unkorrigierten (P_u) und der korrigierten Position (P_k). Durch die Simulation des Ortungssystems ist die tatsächliche Position verfügbar und dient als absoluter Referenzwert (siehe Kapitel 4.3.3). Der Pfad wird mit dem Mauszeiger im Mockup vorgegeben. P_u wird durch das simulierte Ortungssystem anhand von P_t und einem zufälligen Jitter¹ generiert und soll die Toleranz eines echten Ortungssystems imitieren. P_k ist die durch den Algorithmus korrigierte Position von P_u . Der Abstand zwischen P_t und P_u sowie der Abstand zwischen P_t und P_k sind ein Maß für den Fehler bezüglich der Referenzposition P_t . Die angebotene Korrektur wird im Vergleich zur unkorrigierten Position als *besser* angesehen, wenn der Abstand $\overline{P_t P_k}$ kleiner ist als $\overline{P_t P_u}$.

5.1. Testaufbau und Szenarien

Um die beiden Algorithmen vergleichen zu können, müssen Testszzenarien entwickelt werden, anhand derer die unterschiedlichen Korrekturverhalten sichtbar werden. In Abbildung 5.1 werden die Szenarien dargestellt (blau = Szenario 1, grün = Szenario 2, rot = Szenario 3), wobei A den Start und B das Ende der Trajektorie markiert. Es wurden drei repräsentative Szenarien ausgewählt, die folgenden Anwendungsfällen entsprechen:

Szenario 1 (blau) - Bewegung entlang eines Flures

Die tatsächlich zurückgelegte Trajektorie bewegt sich entlang des Flures, ohne einen Raumwechsel zu vollziehen. Die unkorrigierte Position wird aufgrund der Toleranzen

¹Entsprechend der Random-Funktion wird ein Offset zwischen 0 m und einem vorgegebenen maximalen Radius in x- und y-Richtung berechnet

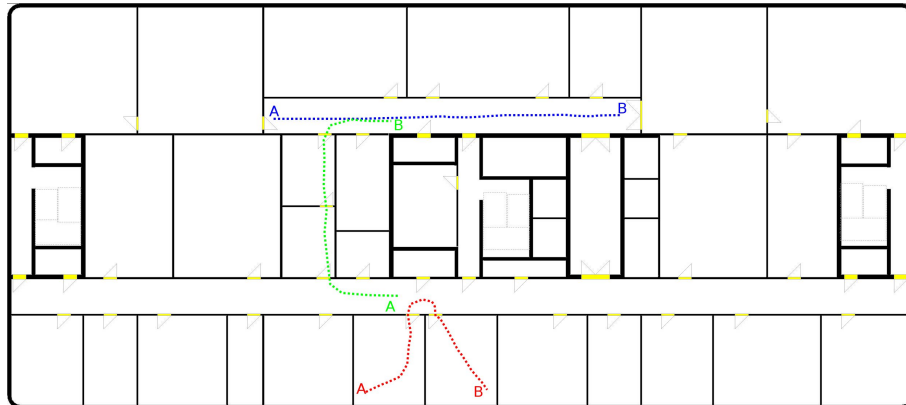


Abbildung 5.1.: Szenarien für die Evaluation der Algorithmen

zufällig in den angrenzenden Räumen liegen. Dies gilt es mit dem Algorithmus zu erkennen und zu korrigieren.

Szenario 2 (grün) - Mehrere Raumwechsel von Flur zu Flur

Die zweite Trajektorie vollzieht einen Flurwechsel über mehrere aufeinander folgende Räume. Weitere Räume sind von dem unteren Flur aus in erreichbarer Nähe und kommen für die logische Überprüfung des Raumwechsels in Frage. Dieses Szenario soll die Korrektur bei einer mehrdeutigen Pfadentwicklung überprüfen.

Szenario 3 (rot) - Wechsel in einen Nachbarraum

Hier wird auf kurzer Distanz ein Raumwechsel über den Flur vorgenommen, was einem Besuch eines Nachbarbüros entsprechen soll. Bei der Korrektur ist zu überprüfen, ob die Erkennung eines Raumwechsels stattfindet und das Pendeln zwischen den Räumen verhindert wird.

Jedes Szenario wird mit drei unterschiedlichen (simulierten) Ortungsgenauigkeiten von 1, 5 und 10 Metern durchlaufen. Für jede Toleranz wird eine Messreihe von 5 unabhängigen Trajektorien einschließlich des zufälligen Jitters erstellt, um einen Trend der Korrektur feststellen zu können. Hierfür werden die von dem simulierten Ortungssystem gelieferten Positionsdaten einschließlich des Jitters aufgezeichnet und für den 1. und 2. Algorithmus (*restriktiver* und *fehlertoleranter* Ansatz) identisch wieder abgespielt. So wird z. B. das 1. Szenario mit einer Toleranz von 1 m im 3. Durchlauf aufgezeichnet und für die beiden Algorithmen exakt gleich vorgegeben, um das unterschiedliche Korrekturverhalten unverfälscht zu erhalten und vergleichbar zu machen. Es ergibt sich das Testmuster aus [Abbildung 5.2.](#)

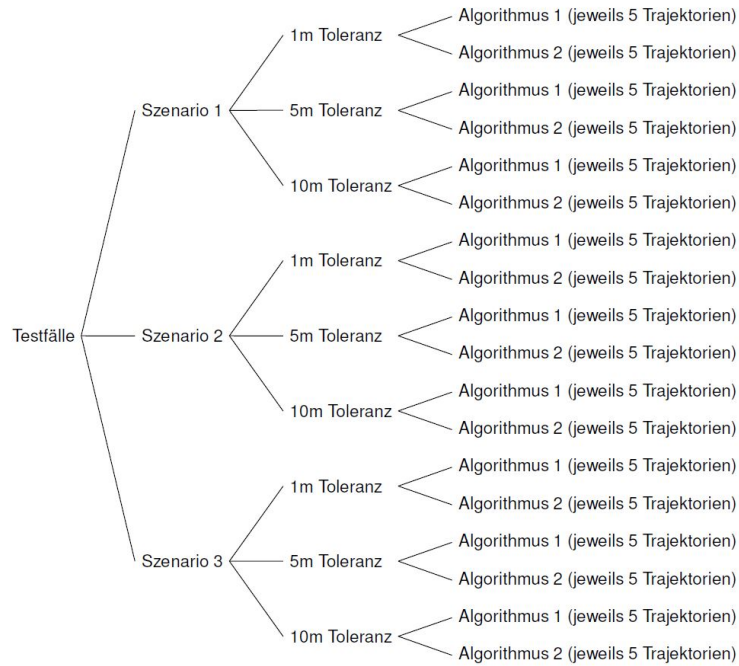


Abbildung 5.2.: Durchgeführte Testmuster

Die Messung liefert ein Tupel bestehend aus dem tatsächlichen Ortungspunkt, dem entsprechenden unkorrigierten und korrigierten Punkt. Die Distanzen werden in Relation zueinander gesetzt ($\frac{\overline{P_t P_k}}{\overline{P_t P_u}}$) und liefern eine Bewertungsmöglichkeit für die Korrekturergebnisse. Werte zwischen 0 und < 1 entsprechen einer verbesserten (positiven) Korrektur im Vergleich zu der unkorrigierten Position. Werte > 1 lassen darauf schließen, dass die korrigierte Position weiter von P_t entfernt liegt als die unkorrigierte Position (negative Korrektur). Bei einem Wert von exakt 1 liegt weder eine Verbesserung noch Verschlechterung vor und es ist sehr wahrscheinlich, dass P_k und P_u identisch sind und im selben Raum liegen.

Die Messreihen werden gemittelt (siehe Formel 5.1), die Varianz bestimmt (Formel 5.2) und die Standardabweichung betrachtet (Formel 5.3).

$$\text{Arithmetisches Mittel: } \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (5.1)$$

$$\text{Varianz: } \sigma^2 = \frac{1}{n} \sum_{i=1}^n |(x_i - \bar{x})^2| \quad (5.2)$$

$$\text{Standardabweichung: } \sigma = \sqrt{\sigma^2} \quad (5.3)$$

Anhand des arithmetischen Mittels werden die jeweiligen Messergebnisse zusammengefasst und gleichen einzelne Ausreißer aus. Je umfangreicher die Messreihen werden, desto mehr nähert man sich dem tatsächlichen Korrekturverhalten. Die Varianz ist der Durchschnitt der quadrierten Abweichungen der Messwerte vom Mittelwert. Je näher der Wert in Richtung 0 tendiert, desto näher liegen die Messungen um einen zentralen Messwert verteilt. Die Standardabweichung entspricht in etwa der Varianz, der Wertebereich ist jedoch identisch mit den Messreihen und beschreibt die Streuung um den Mittelwert.

5.2. Messergebnisse

Die Szenarien werden nacheinander abgearbeitet. Für jedes Szenario werden die Korrekturen für 1 m, 5 m und 10 m der beiden Algorithmen in einer Tabelle vergleichend gegenüber gestellt. In der GUI werden die korrigierten Trajektorien visualisiert. Die Anzeige ist darauf ausgelegt die Korrektur zur Laufzeit zu veranschaulichen. Sie zeigt den aktuellen unkorrigierten Punkt P_u (roter Punkt) und den entsprechenden korrigierten Punkt P_k (grüner Punkt) an. P_k wird in die Historie integriert und die logische Reihenfolge durch ein Verbinden mit dem jeweiligen Vorgänger veranschaulicht. Ein schwarzer Punkt bedeutet, dass P_k und P_u identisch sind und ein (kleiner) roter Punkt signalisiert eine Korrektur (siehe z. B. Abb. 5.12).

5.2.1. Szenario 1

In Szenario 1 wird die Korrektur entlang eines Flures überprüft. Es findet kein planmäßiger Raumwechsel statt, wobei die tatsächliche Trajektorie mittig im Flur verläuft (vgl. Abb. 5.3).

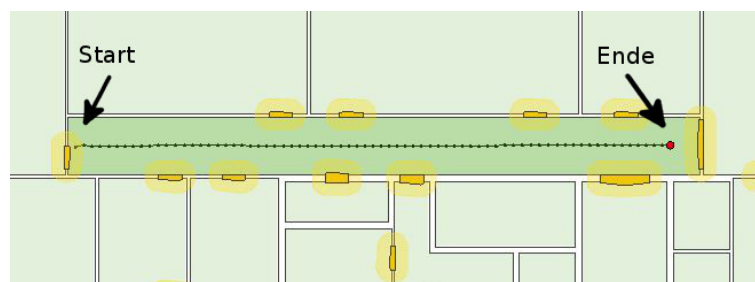


Abbildung 5.3.: Szenario 1 - beispielhafte tatsächliche Trajektorie

1m Toleranz

Die Messreihen für Szenario 1 mit einer Toleranz von 1 m haben für die beiden Algorithmen folgende Ergebnisse geliefert (siehe Tabelle 5.1).

	Algorithmus 1			Algorithmus 2		
	A. Mittel	Varianz	Standardabw.	A. Mittel	Varianz	Standardabw.
1. Durchlauf	1	0	0	1	0	0
2. Durchlauf	1	0	0	1	0	0
3. Durchlauf	1,0362	0,098	0,313	1,0362	0,098	0,313
4. Durchlauf	1	0	0	1	0	0
5. Durchlauf	1	0	0	1	0	0
Durchschnitt	1,0077	0,0209	0,1447	1,0077	0,0209	0,1447

Tabelle 5.1.: Szenario 1 - 1 m Toleranz - 5 Durchläufe

Bei einer Toleranz von 1 m springt die zufällig generierte Position selten in benachbarte Räume, demzufolge stimmen P_k und P_u nahezu immer überein. Dies wird anhand der visuellen Korrektur (siehe Abb. 5.5 und 5.6) deutlich, welche mit der unkorrigierten Trajektorie (vgl. Abb. 5.4) identisch ist.



Abbildung 5.4.: Unkorrigiert - Szenario 1 - 1 m Toleranz - 1. Durchlauf

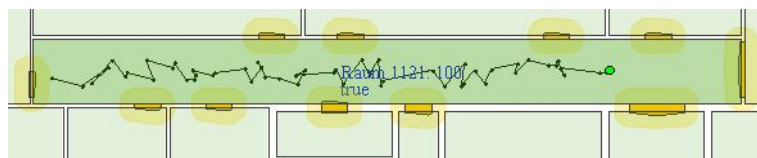


Abbildung 5.5.: Algorithmus 1 - Szenario 1 - 1 m Toleranz - 1. Durchlauf

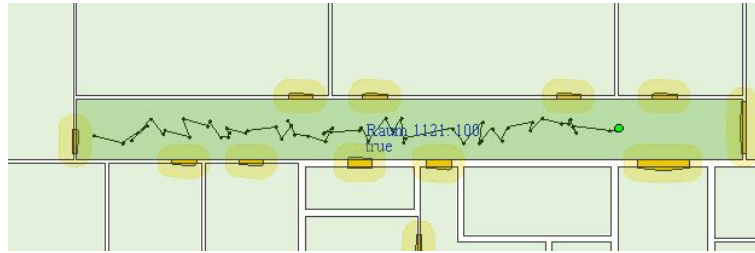


Abbildung 5.6.: Algorithmus 2 - Szenario 1 - 1 m Toleranz - 1. Durchlauf

Für eine Toleranz von einem Meter verhält sich der 2. Algorithmus nahezu genau so wie der *restriktive Ansatz*. Dies lässt sich dadurch erklären, dass die beiden Ansätze auf der selben Grundidee basieren und sich nur in der Bewertung bei einem Raumwechsel unterscheiden. Da bis auf den 3. Durchlauf kein Raumwechsel aufgrund der geringen Toleranz stattfindet, sind die Messergebnisse identisch.

5m Toleranz

Interessanter wird es, wenn die Toleranz auf 5 m erhöht wird, wobei eine positive Korrektur stattfindet. In Tabelle 5.2 sind die Messreihen der beiden Algorithmen bereits zusammengefasst.

	Arithmetisches Mittel	Varianz	Standardabweichung
Algorithmus 1	0,8858	0,0556	0,2357
Algorithmus 2	0,871	0,0337	0,1836

Tabelle 5.2.: Szenario 1 - 5 m Toleranz - Gemittelt über 5 Durchläufe

Die unkorrigierte Trajektorie (siehe Abb. 5.7) reicht in die angrenzenden Räume. Bis auf ein paar kleine Ausreißer (die für die unterschiedlichen Messergebnisse verantwortlich sind) wird in allen Durchläufen die gesamte Trajektorie in den tatsächlichen Raum (Flur) korrigiert (vgl. Abb. 5.8 und 5.9).

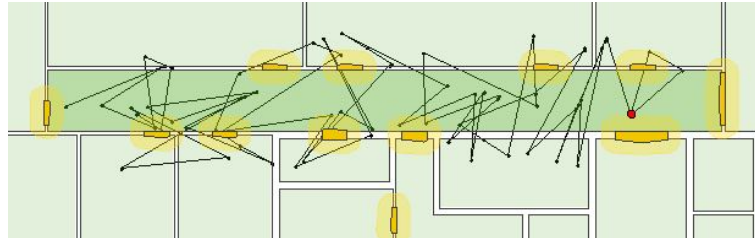


Abbildung 5.7.: Unkorrigiert - Szenario 1 - 5 m Toleranz - 2. Durchlauf

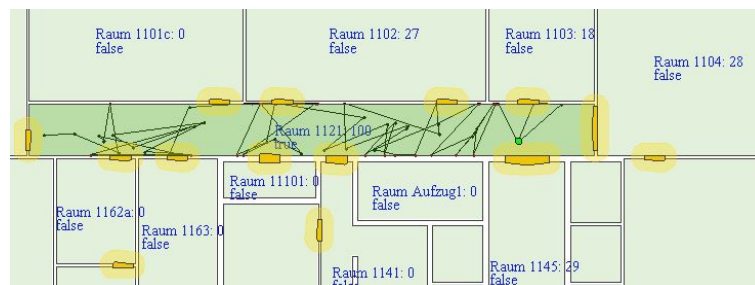


Abbildung 5.8.: Algorithmus 1 - Szenario 1 - 5 m Toleranz - 2. Durchlauf

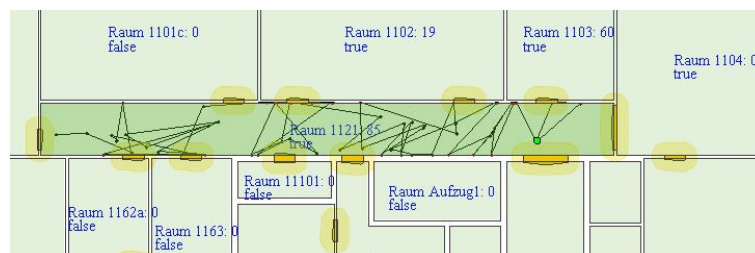


Abbildung 5.9.: Algorithmus 2 - Szenario 1 - 5 m Toleranz - 2. Durchlauf

10m Toleranz

Bei einer Toleranz von 10 m werden folgende Ergebnisse erzielt (Tabelle 5.3). Die Zahlen sprechen für den 1. Ansatz, der eine rechnerisch bessere Korrektur bietet.

	Arithmetisches Mittel	Varianz	Standardabweichung
Algorithmus 1	0,8146	0,1145	0,3384
Algorithmus 2	0,8413	0,1285	0,3584

Tabelle 5.3.: Szenario 1 - 10 m Toleranz - Gemittelt über 5 Durchläufe

Die unkorrigierte Trajektorie (vgl. Abb. 5.10) reicht weit in die angrenzenden Räume hinein. Im Falle des 2. Durchlaufs korrigiert der 1. Algorithmus auf den ersten Blick visuell besser als der fehlertolerante Ansatz (vgl. Abb. 5.11 und 5.12). Beim 2. Algorithmus sind mehr Trajektorien in den Nachbarräumen. Doch am Ende liegt der korrigierte Punkt wieder im Flur, während beim 1. Ansatz die Position noch im angrenzenden Raum verbleibt.

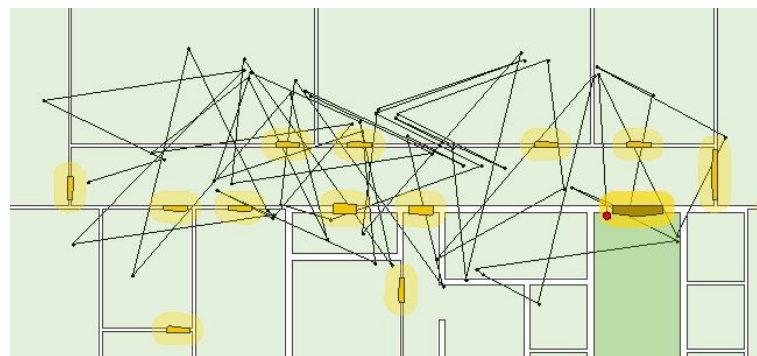


Abbildung 5.10.: Unkorrigiert - Szenario 1 - 10 m Toleranz - 2. Durchlauf

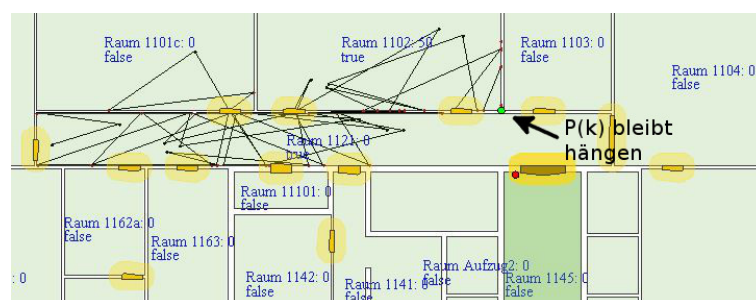


Abbildung 5.11.: Algorithmus 1 - Szenario 1 - 10 m Toleranz - 2. Durchlauf

5. Evaluation

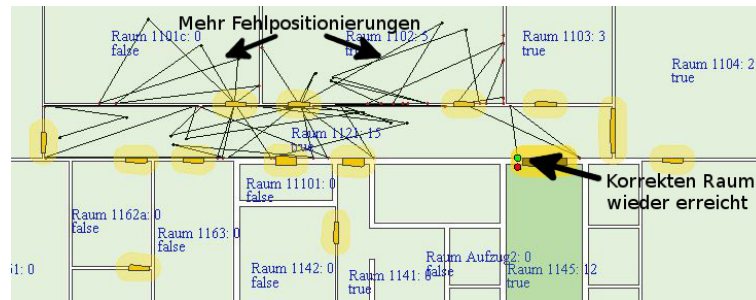


Abbildung 5.12.: Algorithmus 2 - Szenario 1 - 10 m Toleranz - 2. Durchlauf

Bei dem 4. Durchlauf (siehe Abb. 5.13) korrigiert der 2. Algorithmus viel besser als der restriktive Ansatz (vgl. Abb. 5.14 und 5.15).

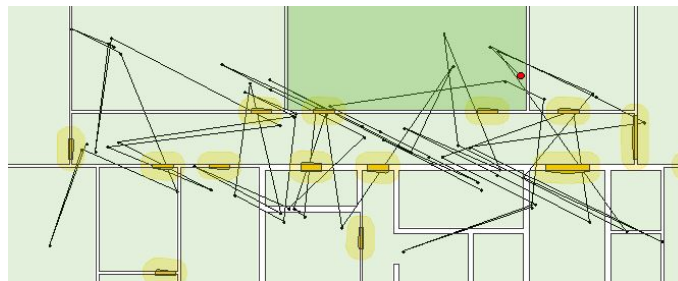


Abbildung 5.13.: Unkorrigiert - Szenario 1 - 10 m Toleranz - 4. Durchlauf

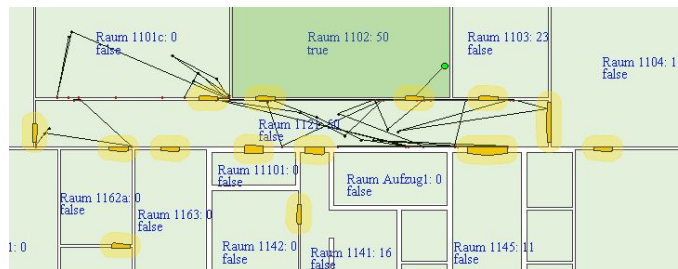


Abbildung 5.14.: Algorithmus 1 - Szenario 1 - 10 m Toleranz - 4. Durchlauf

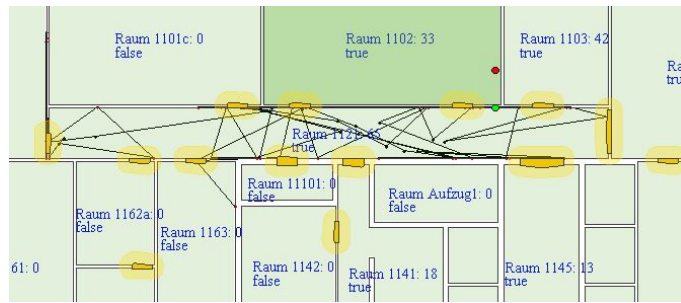


Abbildung 5.15.: Algorithmus 2 - Szenario 1 - 10 m Toleranz - 4. Durchlauf

Während rechnerisch noch immer eine positive Korrektur vorgenommen wird, weichen die visuellen Ergebnisse erheblich voneinander ab. Manche Fehlerfälle können ganz gut korrigiert werden, doch andere ungünstige Durchläufe können dazu führen, dass ein Nachbarraum als wahrscheinlich angenommen wird und die Position nicht mehr in den Flur zurück kehrt.

Für den restriktiven Ansatz werden nur noch zwei von fünf Testfällen einigermaßen korrekt in das Gebäudemodell eingepasst. Das gleiche gilt für den fehlertoleranten Ansatz. Es handelt sich jedoch nicht um die selben Testfälle. In den anderen Fällen sind die zufälligen Positionen zu selten im korrekten Raum (Flur), wodurch dessen WLK sinkt und die angrenzenden Räume wahrscheinlicher werden.

5.2.2. Szenario 2

In Szenario 2 startet die Trajektorie im unteren Flur und bewegt sich durch zwei Räume in den oberen Flur. Auf diesem Weg befinden sich mehrere logisch erreichbare Räume, jedoch führen nicht alle zum Ziel. Die tatsächliche Trajektorie für die folgenden Testfälle bewegt sich nach dem beispielhaften Muster von [Abbildung 5.16](#).

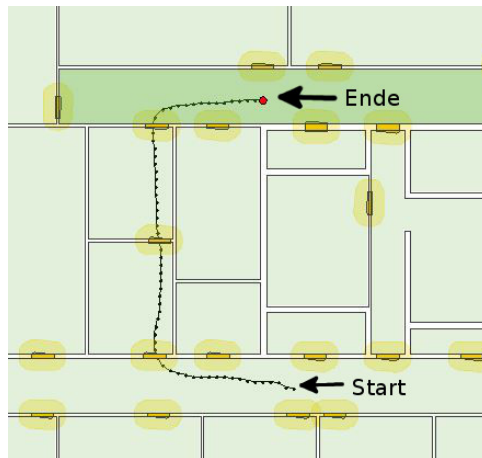


Abbildung 5.16.: Szenario 2 - beispielhafte tatsächliche Trajektorie

1m Toleranz

Aus Tabelle 5.4 wird ersichtlich, dass im Mittel immer eine *schlechtere* Korrektur vorgenommen wird, als das simulierte Ortungssystem liefert. Gerade bei Algorithmus 2 ist bei der Varianz eine starke Streuung auszumachen.

	Arithmetisches Mittel	Varianz	Standardabweichung
Algorithmus 1	1,1223	0,1983	0,4453
Algorithmus 2	1,3822	1,9556	1,3984

Tabelle 5.4.: Szenario 2 - 1 m Toleranz Toleranz - Gemittelt über 5 Durchläufe

Die Korrekturfehler für die 1 m Toleranz lassen sich dadurch erklären, dass ein Raumwechsel erst mit steigender Wahrscheinlichkeit für den neuen Raum stattfindet. Hierfür müssen mehrere Ortungen in dem neuen Raum liegen, währenddessen verbleibt die korrigierte Position im vorherigen Raum R_{legal} . Somit steigt der Abstand zur tatsächlichen Position und ist größer als $\overline{P_t P_u}$ (siehe Abb. 5.18 und 5.19 im Vergleich zu 5.17).

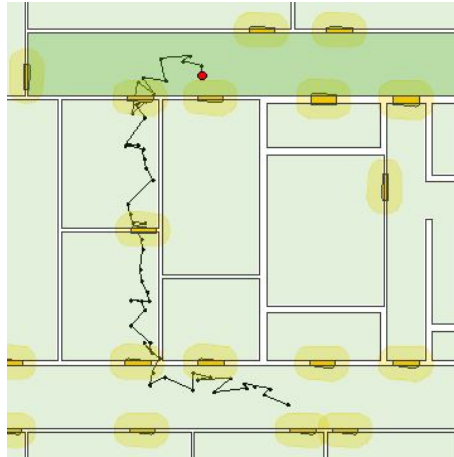


Abbildung 5.17.: Unkorrigiert - Szenario 2 - 1 m Toleranz - 1. Durchlauf

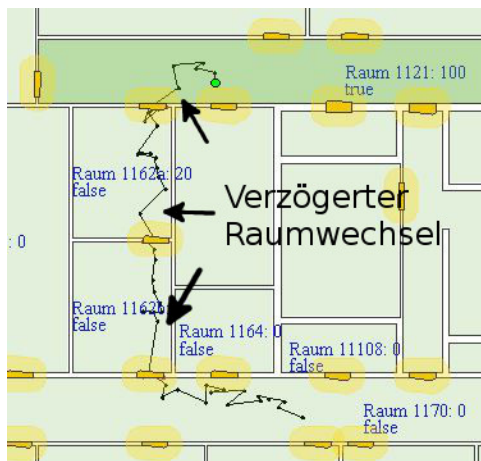


Abbildung 5.18.: Algorithmus 1 - Szenario 2 - 1 m Toleranz - 1. Durchlauf

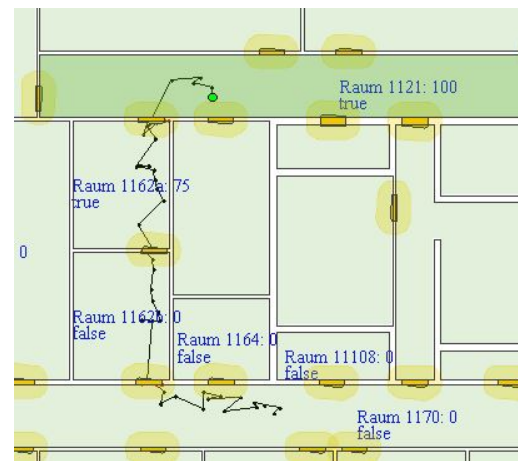


Abbildung 5.19.: Algorithmus 2 - Szenario 2 - 1 m Toleranz - 1. Durchlauf

Da das simulierte Ortungssystem eine zu hohe Präzision hat und die restriktiven Elemente zu weit entfernt liegen um einen positiven Effekt durch den Algorithmus zu haben, kommen lediglich die negativen Auswirkungen des verzögerten Raumwechsels zum Tragen. Die hohe Varianz ist einigen Extremwerten geschuldet, in welchen P_u zufällig sehr nahe an P_t liegt und P_k noch in dem vorherigen Raum R_{legal} zurückgehalten wird.

5m Toleranz

In der rechnerischen Auswertung (siehe Tabelle 5.5) wird deutlich, dass der 1. Algorithmus bei einer Toleranz von 5 m Schwierigkeiten hat die Trajektorie zu korrigieren. Dies äußert sich in einer sehr hohen Varianz und einer im Mittel schlechteren Korrektur. Die Varianz des 2. Algorithmus ist sehr niedrig, was auf eine konsistente Korrektur schließen lässt.

	Arithmetisches Mittel	Varianz	Standardabweichung
Algorithmus 1	1,407	3,1952	1,7875
Algorithmus 2	0,8755	0,0734	0,2709

Tabelle 5.5.: Szenario 2 - 5 m Toleranz - Gemittelt über 5 Durchläufe

Wird die Toleranz auf 5 m erhöht, so wird der tatsächliche Raumwechsel für Algorithmus 1 in drei von fünf Fällen und für Algorithmus 2 in allen Fällen korrekt nachvollzogen (exemplarisch wird Durchlauf 5 gewählt, vgl. Abb. 5.21 und 5.22 im Vergleich zu 5.20).

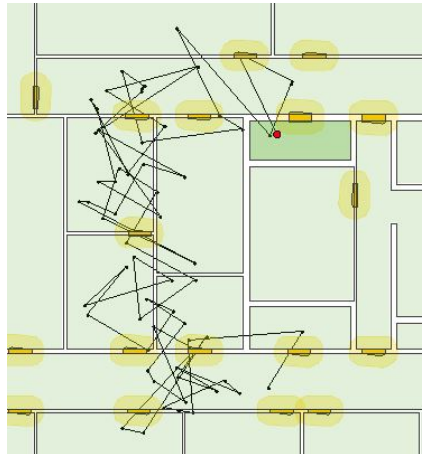


Abbildung 5.20.: Unkorrigiert - Szenario 2 - 5 m Toleranz - 5. Durchlauf

5. Evaluation

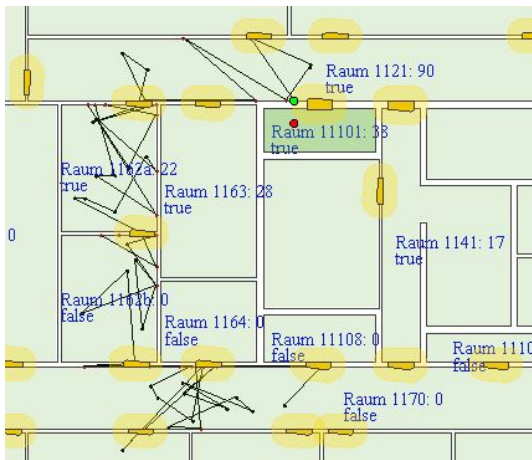


Abbildung 5.21.: Algorithmus 1 - Szenario 2 - 5 m Toleranz - 5. Durchlauf

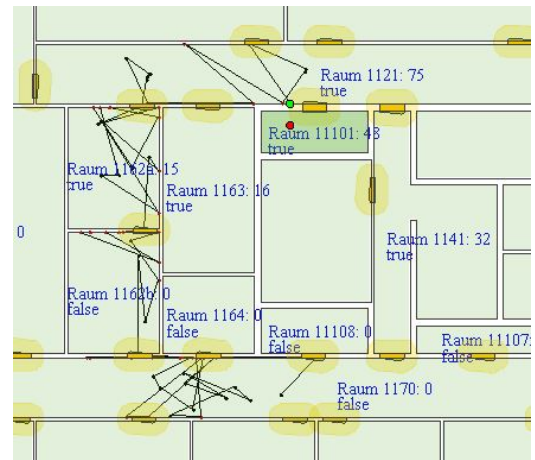


Abbildung 5.22.: Algorithmus 2 - Szenario 2 - 5 m Toleranz - 5. Durchlauf

In den beiden anderen Fällen liegen die zufällig generierten Ortungspunkte für den restriktiven Algorithmus so ungünstig (Abb. 5.23), dass die duale Raumentwicklung nicht erkannt wird und die Position in einem falschen Raum hängen bleibt (vgl. Abb. 5.24 im Vergleich zum 2. Algorithmus 5.25).

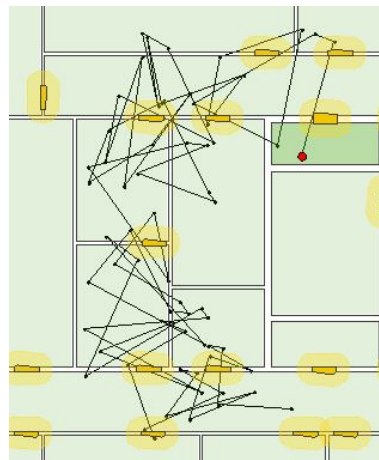


Abbildung 5.23.: Unkorrigiert - Szenario 2 - 5 m Toleranz - 2. Durchlauf

5. Evaluation

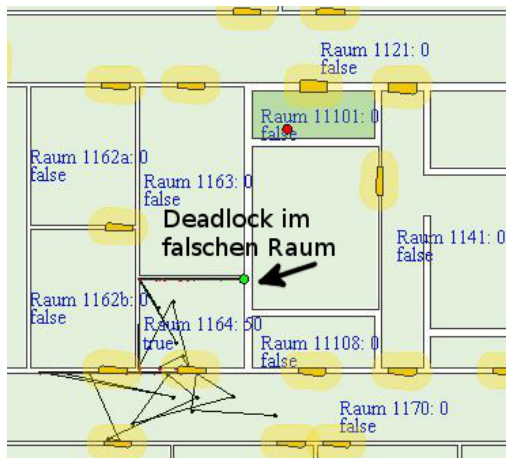


Abbildung 5.24.: Algorithmus 1 - Szenario 2 - 5 m Toleranz - 2. Durchlauf

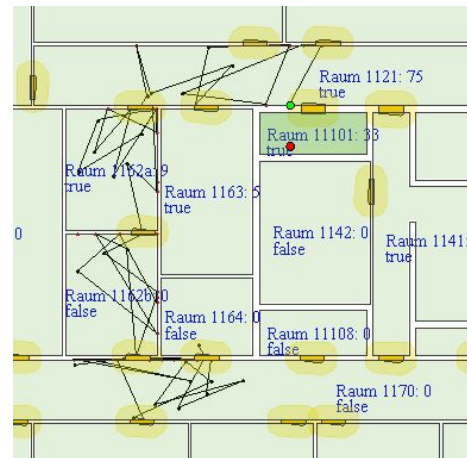


Abbildung 5.25.: Algorithmus 2 - Szenario 2 - 5 m Toleranz - 2. Durchlauf

10m Toleranz

Bei einer Toleranz von 10 m ist die durchschnittliche Korrektur des 1. Algorithmus schlechter als die unkorrigierte Trajektorie, wohingegen der 2. Algorithmus noch eine leichte Verbesserung bietet (siehe Tabelle 5.6).

	Arithmetisches Mittel	Varianz	Standardabweichung
Algorithmus 1	1,0605	0,6267	0,7916
Algorithmus 2	0,9347	0,5705	0,7553

Tabelle 5.6.: Szenario 2 - 10 m Toleranz - Gemittelt über 5 Durchläufe

Der restriktive Ansatz erkennt nur noch in einem von fünf Testfällen die entsprechenden korrekten Raumwechsel (siehe Abb. 5.27). Bei dem fehlertoleranten Ansatz folgt die Trajektorie in drei von fünf Fällen der tatsächlichen Raumabfolge (Abbildung 5.26 und 5.28).

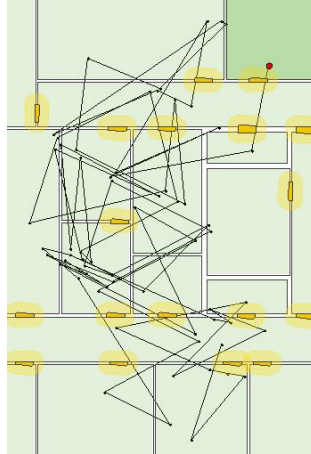


Abbildung 5.26.: Unkorrigiert - Szenario 2 - 10 m Toleranz - 5. Durchlauf

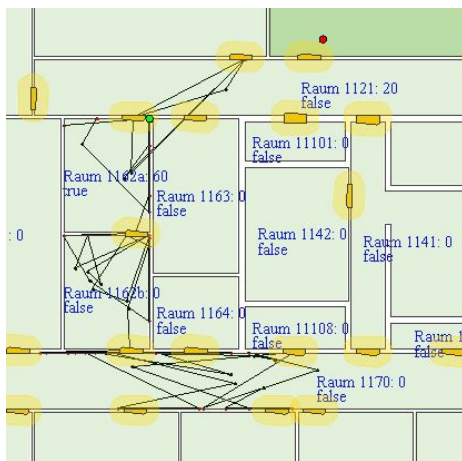


Abbildung 5.27.: Algorithmus 1 - Szenario 2 - 10 m Toleranz - 5. Durchlauf

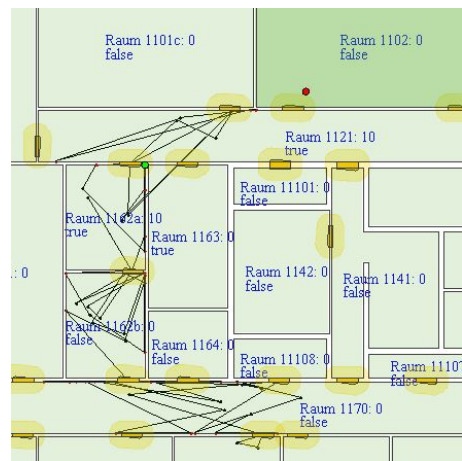


Abbildung 5.28.: Algorithmus 2 - Szenario 2 - 10 m Toleranz - 5. Durchlauf

In zwei Fällen wird die parallele Raumentwicklung vom 1. Algorithmus nicht erkannt und bleibt in einem falschen Raum hängen (siehe Abb. 5.30 bei dem Testlauf 5.29). Der 2. Algorithmus kann diesen Durchlauf richtig korrigieren (Abb. 5.31).

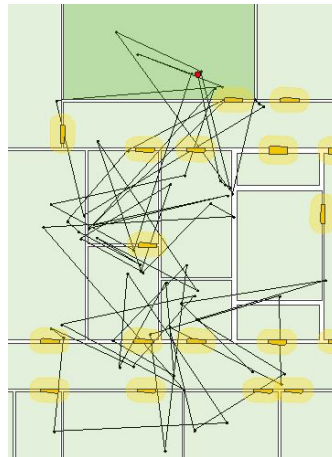


Abbildung 5.29.: Unkorrigiert - Szenario 2 - 10 m Toleranz - 3. Durchlauf

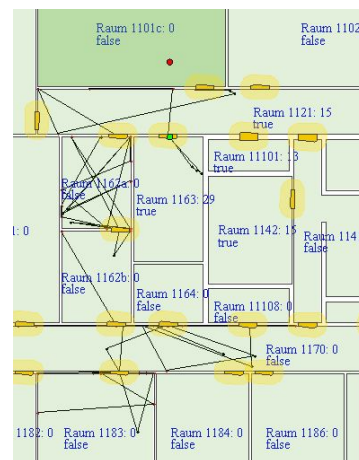
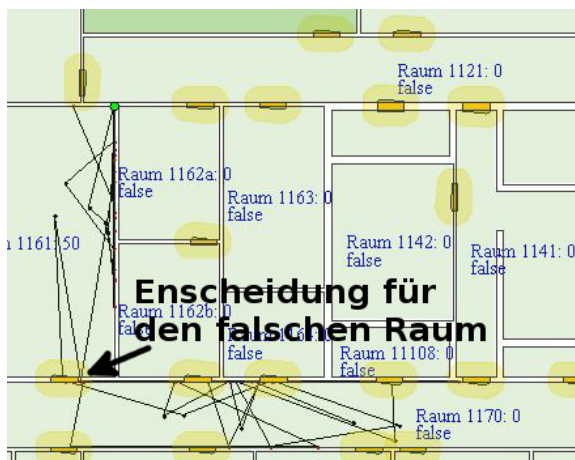


Abbildung 5.30.: Algorithmus 1 - Szenario 2 - 10 m Toleranz - 3. Durchlauf

Abbildung 5.31.: Algorithmus 2 - Szenario 2 - 10 m Toleranz - 3. Durchlauf

In einigen Fällen verbleibt die Position im unteren Flur und wird erst bei einer zu großen Abweichung anhand der Fehlererkennung in den oberen Raum korrigiert, dies betrifft jedoch auch teilweise den 2. Algorithmus (s. Abbildung 5.33 und 5.34). Die Korrektur erscheint auch sehr herausfordernd, da aufgrund der hohen Streuung selten mehrere

aufeinander folgende Ortungspunkte in ein und dem selben Raum liegen Dadurch wird die WLK zu schnell verringert (siehe Abb. 5.32).

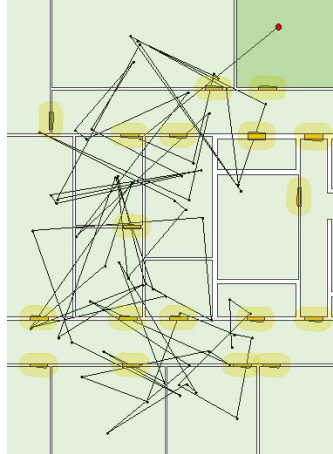


Abbildung 5.32.: Unkorrigiert - Szenario 2 - 10 m Toleranz - 1. Durchlauf

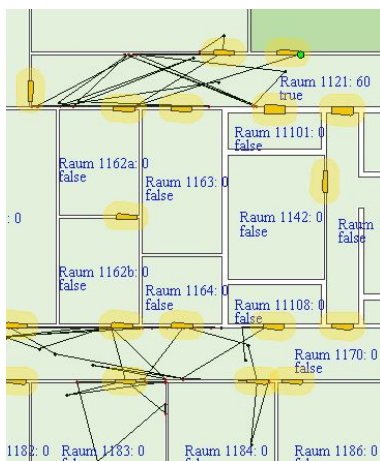


Abbildung 5.33.: Algorithmus 1 - Szenario 2 - 10 m Toleranz - 1. Durchlauf

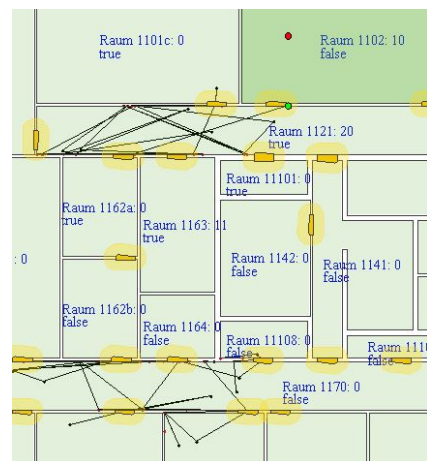


Abbildung 5.34.: Algorithmus 2 - Szenario 2 - 10 m Toleranz - 1. Durchlauf

5.2.3. Szenario 3

In Szenario 3 wird ein benachbarter Raum besucht. Die Besonderheit liegt bei der verhältnismäßig kurzen Trajektorie (siehe Abb. 5.35), wobei zusätzlich zwei Raumwechsel kurz hintereinander stattfinden (vom 1. Raum in den Flur und von dort in den 2. Raum).

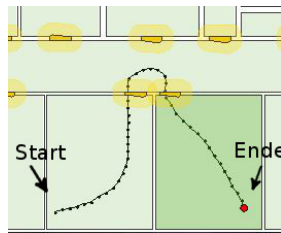


Abbildung 5.35.: Szenario 3 - beispielhafte tatsächliche Trajektorie

1m Toleranz

Die Messergebnisse von Szenario 3 für eine Toleranz von 1 m sind in der Tabelle 5.7 dargestellt.

	Arithmetisches Mittel	Varianz	Standardabweichung
Algorithmus 1	1,174	0,4564	0,6755
Algorithmus 2	1,4494	1,156	1,0752

Tabelle 5.7.: Szenario 3 - 1 m Toleranz - Gemittelt über 5 Durchläufe

Beide Algorithmen liefern wie zu erwarten eine verschlechterte Korrektur (siehe vorheriges Szenario), da die Raumübergänge verzögert werden und keine positive Korrektur an den Raumgrenzen stattfindet (vgl. Abb. 5.37 und 5.38 für Testszenario 5.36).

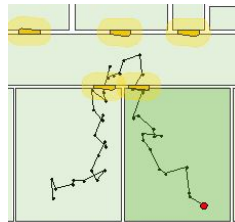


Abbildung 5.36.: Unkorrigiert - Szenario 3 - 1 m Toleranz - 1. Durchlauf

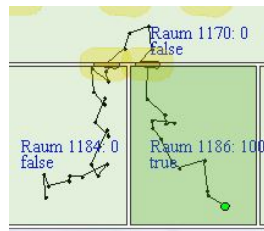


Abbildung 5.37.: Algorithmus 1 - Szenario 3 - 1 m Toleranz - 1. Durchlauf

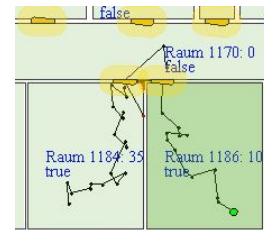


Abbildung 5.38.: Algorithmus 2 - Szenario 3 - 1 m Toleranz - 1. Durchlauf

In einem Testlauf für Algorithmus 2 wird die Trajektorie nicht korrekt erkannt und verbleibt im ersten Raum. Dies lässt sich dadurch erklären, dass nicht ausreichend viele Ortungen im Flur vorliegen um die WLK ausreichend hochzuzählen bevor der zweite Raum betreten wird. Dementsprechen hoch fällt auch die Varianz aus.

5m Toleranz

Bei einer Toleranz von 5 m führt der 1. Algorithmus im Mittel zu einer besseren Korrektur als der fehlertolerante Ansatz (vgl. Tabelle 5.8). Beispielfhaft für den zweiten Algorithmus wird im Anhang A.1 die gesamte Messung angegeben.

	Arithmetisches Mittel	Varianz	Standardabweichung
Algorithmus 1	0,9724	0,2106	0,459
Algorithmus 2	1,0094	0,3181	0,564

Tabelle 5.8.: Szenario 3 - 5 m Toleranz - Gemittelt über 5 Durchläufe

In allen Testfällen der beiden Algorithmen wird der Raumwechsel korrekt erkannt, was eine geringe Varianz zur Folge hat (vgl. Abbildung 5.40 und 5.41 für den Testfall 5.39). Das dennoch relativ knappe Verhältnis zwischen unkorrigierter und korrigierter Trajektorie lässt sich auf die kurze Teststrecke und deren verhältnismäßig großen Räumen zurückführen, welche keine ausreichende positive Einschränkung zur Korrektur bieten. Besonders hervorzuheben ist die Korrektur hinsichtlich der springenden Positionen zwischen den benachbarten Räumen (1184 und 1186), welche keine Wände durchkreuzen.

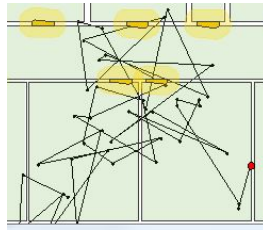


Abbildung 5.39.: Unkorrigiert - Szenario 3 - 5 m Toleranz - 4. Durchlauf

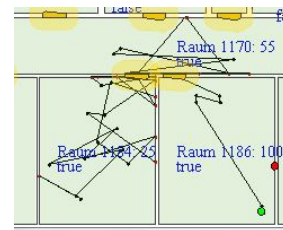
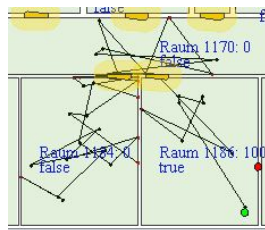


Abbildung 5.40.: Algorithmus 1 - Szenario 3 - 5 m Toleranz - 4. Durchlauf

Abbildung 5.41.: Algorithmus 2 - Szenario 3 - 5 m Toleranz - 4. Durchlauf

10m Toleranz

Die Messergebnisse (siehe Tabelle 5.9) für eine Toleranz in Höhe von 10 m führen für beide Algorithmen zu einer positiven Korrektur. Bei dem restriktiven Ansatz lässt die hohe Varianz auf ein sehr ungleichmäßiges Korrekturverhalten schließen.

	Arithmetisches Mittel	Varianz	Standardabweichung
Algorithmus 1	0,9535	1,3602	1,1663
Algorithmus 2	0,9139	0,4931	0,7022

Tabelle 5.9.: Szenario 3 - 10 m Toleranz - Gemittelt über 5 Durchläufe

Obwohl im Mittel eine positive Korrektur für den 1. Algorithmus angegeben wird, erfolgt in keinem der fünf Testfälle eine korrekte Erkennung der zurückgelegten Trajektorie, wohingegen drei Testfälle vom 2. Algorithmus korrekt erkannt werden (vgl. Abb. 5.43 und 5.44 für den Testfall 5.42). Im Vergleich zur unkorrigierten Trajektorie bietet der fehlertolerante Algorithmus noch eine kleine Verbesserung bei einer relativ kleinen Varianz, obwohl die Korrektur in machen Fällen im falschen Raum liegt.

Da die Korrekturen aber im Schnitt näher (aber nicht zwangsläufig im richtigen Raum)

an der tatsächlichen Position liegen als P_u , wird das Ergebnis rechnerisch noch als Verbesserung angesehen.

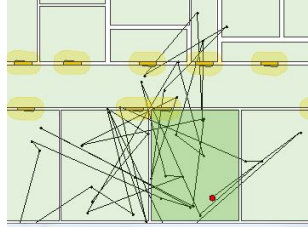


Abbildung 5.42.: Unkorrigiert - Szenario 3 - 10 m Toleranz - 3. Durchlauf

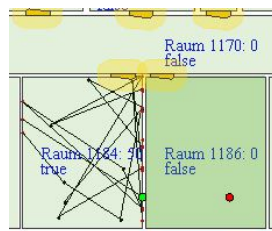


Abbildung 5.43.: Algorithmus 1 - Szenario 3 - 10 m Toleranz - 3. Durchlauf

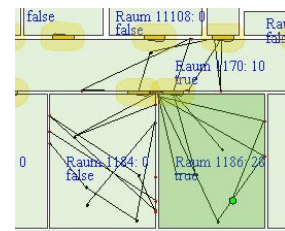


Abbildung 5.44.: Algorithmus 2 - Szenario 3 - 10 m Toleranz - 3. Durchlauf

5.3. Vergleichende Betrachtung der Korrekturergebnisse

In diesem Kapitel werden die Korrekturergebnisse des *restriktiven* und *fehlertoleranten* Algorithmus im Detail verglichen und bewertet. Ein gesonderter Vergleich der Korrekturergebnisse für jedes einzelne Szenario, jede Toleranz und jeden Testfall erscheint nicht zielführend für eine Gesamtbewertung. In [Abbildung 5.45](#) wird beispielhaft der Abstand zur tatsächlichen Position über den Messverlauf für die beiden Algorithmen aufgezeigt. Es ist zu erkennen, dass sich P_k für den 1. Algorithmus immer weiter von P_t entfernt und somit schlechter ist als der 2. Algorithmus. Um dieses Ergebnis in Zahlen fassbar zu machen erscheint es jedoch besser, die Messergebnisse zu kombinieren und in ein relatives Verhältnis zu setzen.

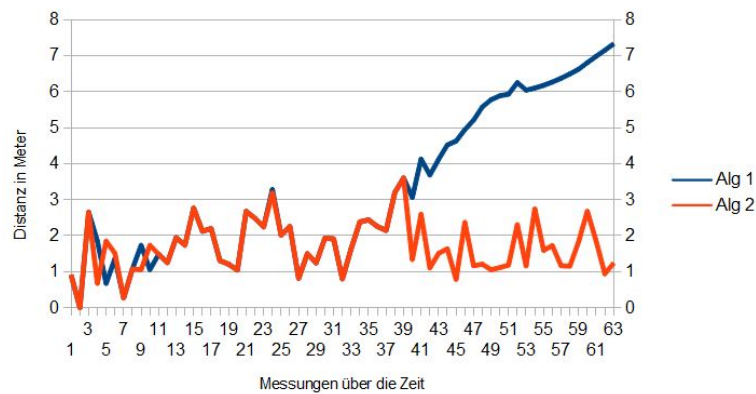


Abbildung 5.45.: Vergleich der Korrekturfelder für Szenario 2 mit einer Toleranz von 5m

Die Tabelle 5.10 fasst die gesamten erhobenen Messerergebnisse zusammen. Die beiden Algorithmen werden pro Toleranz (1, 5 und 10 Meter) gegenüber gestellt und vereinen jeweils die drei Szenarien um ein Gesamtbild des Korrekturverhaltens zu liefern. Alg 1 steht hierbei für den restriktiven Ansatz (siehe Kap. 4.2.4) und Alg 2 für den fehlertoleranten Ansatz (siehe Kap. 4.2.5).

	Alg 1 - 1m	Alg 2 - 1m	Alg 1 - 5m	Alg 2 - 5m	Alg 1 - 10m	Alg 2 - 10m
Mittelwert	1,0920	1,2546	1,0968	0,9074	0,9401	0,8935
Median	1	1	1	1	0,9076	0,9071
Minimum	0,2682	0,2682	0,0448	0,0448	0,0018	0,0018
Maximum	5,6493	14,7521	25,6154	6,3676	14,1986	12,0408
Varianz	0,2032	1,0373	1,2900	0,1230	0,6107	0,3822
Standardabw.	0,4507	1,0185	1,1358	0,3508	0,7815	0,6183

Tabelle 5.10.: Vergleich der Messergebnisse der beiden Algorithmen

Ein Mittelwert < 1 stellt eine positive Korrektur in Relation zur unkorrigierten Position dar ($\frac{P_t P_k}{P_t P_u}$). Für eine Toleranz von einem Meter bietet keiner der beiden Algorithmen eine Verbesserung der Position. Der 1. Algorithmus ist hierbei besser, was darauf zurückzuführen ist, dass dort der Raumwechsel aufgrund der Einstellungen schneller stattfindet.

Erst ab 5 Metern liefert der 2. Algorithmus eine bessere und positive Korrektur, was auf die parallele Raumentwicklung zurückzuführen ist. Der restriktive Ansatz bleibt in ein paar Testfällen häufiger in einem Deadlock hängen, was sich auch anhand des Maximums und der hohen Varianz zeigt.

Bei einer Toleranz von 10 Metern erfolgt bei beiden Algorithmen eine positive Korrektur, wobei der fehlertolerante Ansatz vorne liegt.

Das rechnerische Ergebnis alleine ist jedoch nicht aussagekräftig genug, da es keine Informationen über die Korrektur zum tatsächlich betretenen Raum liefert. Denn die korrigierte Position kann zwar im Verhältnis zur unkorrigierten Position näher an der tatsächlichen Position liegen und somit für ein positives Korrekturergebnis sorgen. Dies sagt jedoch nicht aus, dass P_k auch in dem tatsächlich betretenen Raum liegt. Hierfür ist die visuelle Auswertung der Messreihen notwendig wie oft sie sich für einen falschen Raum entscheiden, um eine Tendenz der Korrektur-Algorithmen zu erhalten. Für das erste Szenario liegt der restriktive Ansatz beispielsweise ungefähr um den Faktor drei häufiger in einem falschen Raum. Die beiden anderen Szenarien kommen auf ähnliche Resultate. Eine absolute Überlegenheit eines Ansatzes ist nicht gegeben. Der fehlertolerante Algorithmus liefert aber im Schnitt eine zuverlässigere Erkennung der tatsächlich zurückgelegten Trajektorie und verhält sich seinem Namen entsprechend fehlertoleranter um Deadlocks zu vermeiden.

5.4. Fazit

In diesem Kapitel wird ein Fazit der durchgeführten Evaluation gezogen.

Wie sich anhand der unterschiedlichen Testszenarien gezeigt hat, liefern die beiden Algorithmen eine gleichmäßige Korrektur innerhalb eines Bürogebäudes, das ausreichend restriktive Elemente für eine Plausibilitätsprüfung bietet. Das Springen zwischen unerreichbaren Räumen kann unterbunden werden und auch ein Pendeln zwischen erreichbaren benachbarten Räumen wird minimiert. Eine Korrektur innerhalb der Räume wird nicht vorgenommen und entspricht der ursprünglich gelieferten Position. Dies wurde in der Analyse vorhergesehen und lässt sich durch die grundlegende Idee auch nicht lösen.

Regelmäßigkeiten beim Korrekturverhalten sind aufgrund der unübersichtlichen und chaotischen Art der Testdaten nicht direkt nachvollziehbar. Lediglich das Resultat, das eine Raumabfolge korrekt erkannt wurde lässt sich feststellen. Diese Erfolgsaussicht hängt sehr stark mit der gewählten Toleranz des simulierten Ortungssystems zusammen. Ein Ortungssystem mit einer zu hohen Präzision erfährt keine Verbesserung, da die begrenzenden Wände (relativ gesehen) zu weit entfernt liegen. In die Bewertung wird lediglich der negative Einfluss des verzögerten Raumwechsels einbezogen.

Bei einer Toleranz von fünf Metern werden die besten Korrekturergebnisse erzielt. Hier ist die mittlere Wandentfernung und Streuung des Ortungssystems in einem guten Verhältnis und somit die tatsächlich zurückgelegte Trajektorie algorithmisch nachvollziehbar. Ab einer hohen

Toleranz werden die Aussagen über die Raumabfolge zunehmend unsicherer. Dies hat zur Folge, dass die korrigierte Position entweder in einem Raum verbleibt oder in falsche Räume verortet wird. Die Ursache liegt an der zu großen und komplett zufälligen Streuung der simulierten Ortung, wobei die Wahrscheinlichkeit, dass die Position im tatsächlich betretenen Raum liegt, rapide abnimmt. Und um so weniger (unkorrigierte) Ortungen in einem Raum liegen, desto unwahrscheinlicher wird er durch den Algorithmus eingestuft, dass dieser aktuell betreten ist. Eine maximale Toleranz für die Korrektur-Algorithmen lässt sich nicht ohne Weiteres festlegen, da die Erkennungsrate fließend ist und sehr stark von dem jeweilig zufällig generierten Testfall abhängt. Für das hier gewählte Ortungssystem liegt der optimale Korrekturbereich etwa um die fünf Meter und würde für eine präzisere Beschreibung der Grenzen noch umfangreiche Testreihen benötigen.

Die Korrekturergebnisse der jeweiligen Algorithmen hätten noch verbessert werden können, indem die spezifischen Parameter auf die entsprechend simulierte Toleranz abgestimmt worden wären (z. B. *WALLRADIUS*, *ADDVALUEINSIDEROOM* und *SUBVALUEOUTSIDEROOM*). Somit wäre bei einem präziseren Ortungssystem ein Raumwechsel schneller vorgenommen worden, bzw. bei einer höheren Toleranz mehr angrenzende Räume für die Bewertung in Betracht gezogen worden. Um jedoch das Korrekturverhalten der beiden Algorithmen untereinander für verschiedene Toleranzen vergleichen zu können, ohne andere Parameter zu verändern und Fehlerquellen auszuschließen, wurde auf die individuelle Anpassung der Algorithmen verzichtet. Dies führt unter Umständen zu einer verschlechterten Korrektur und holt nicht das Optimum aus dem Algorithmus heraus. Somit ist das Korrekturverhalten nicht zwangsläufig repräsentativ für einen kalibrierten Einsatz in einem Ortungssystem.

6. Schluss

In diesem Kapitel erfolgt eine abschließende Betrachtung der Masterarbeit. Zunächst werden in Kapitel 6.1 die erreichten Ziele zusammengefasst und in Kapitel 6.2 einer Bewertung unterzogen. In einem Ausblick (Kap. 6.3) werden mögliche Weiterentwicklungen und Ideen thematisiert.

6.1. Zusammenfassung

Die vergleichbaren Arbeiten (siehe Kapitel 2) haben gezeigt, dass es eine grundsätzliche Unterscheidung in *relative* und *absolute* Ortungssysteme gibt, deren Korrektur voneinander abweicht und nicht kompatibel ist. Reicht man rein messwertbasierte, absolute Ortungssysteme mit zusätzlichen Informationen an, so erzielen andere Realisierungen eine verbesserte Korrektur. Sowohl bei der Plausibilitätsüberprüfung anhand eines menschlichen Bewegungsmodells, als auch dem Abgleich mit einem zugrunde liegenden Gebäudemodell können unwahrscheinliche und offenbar fehlerhafte Ortungen ausgeschlossen, bzw. korrigiert werden. Für das eigene Korrekturverfahren ergibt sich daher aus den Erfahrungen der vergleichbaren Arbeiten, dass für eine Präzisionssteigerung ein zugrunde gelegtes Gebäude- und Bewegungsmodell benötigt wird. Auf dem Gebiet des Map-Matching für absolute Ortungssysteme existieren erst wenige Realisierungen. Diese basieren meistens auf Partikelfiltern. Daher besteht noch Forschungspotential für alternative Lösungsmöglichkeiten, wie sie bei der eigenen Realisierung Anwendung findet.

Im Verlauf der Analyse (vgl. Kapitel 3) wurde gezeigt, dass die Relevanz einer Ortungsverbesserung besonders für größere öffentliche Gebäude gegeben ist. Die funktionalen Anforderungen liegen bei der *Identifikation eines Fehlers* und der anschließenden *Ortungsverbesserung für mehrdeutige Korrekturpfade*. Hierfür werden sowohl Informationen aus dem Gebäude-, als auch dem Bewegungsmodell verwendet, um physikalisch mögliche Pfade zu identifizieren. Die nicht-funktionalen Anforderungen fokussieren sich hauptsächlich auf die starke Kapselung des Korrekturmoduls um dem Anwender eine leichte An- oder Abwahl der Funktionalität zu bieten. In der Analyse wurden zudem mögliche auftretende Problemstellungen erörtert, als auch deren optimalen Korrekturmöglichkeiten vorgeschlagen.

Die Realisierung der eigenen Entwicklung erfolgte in Kapitel 4 und erforderte mehrere unterstützende Strukturen. Einerseits wurde eine Simulationssoftware für die Ortungspositionen den eigenen Anforderungen entsprechend angepasst und erweitert. Andererseits erforderten die selbst entwickelten Algorithmen ein Framework um die benötigten Informationen (z. B. Ortungspunkt, Berechnungen im Gebäudemodell und Kommunikation) an zentraler Stelle zu vereinen und einen Zugriff zu ermöglichen. Hierfür wurde das ISIS-Framework im Bereich der Visualisierung erweitert, um den Pfadverlauf, die korrigierte und unkorrigierte Position, als auch die Raumwahrscheinlichkeiten darstellen zu können. Im Bereich der Kommunikation wurde ein zusätzlicher Publisher für die korrigierten Positionsdaten eingerichtet und der Subscriber auf das ursprüngliche Ortungssystem angepasst, um erweiterte Datenfelder für die spätere Evaluation zu unterstützen. Die meisten Anpassungen erfolgten im Bereich der räumlichen Analyse, um beispielsweise Punkt-Abstands-Berechnungen oder rekursive Raumsuchen zur Verfügung zu stellen. Die Umsetzung der theoretisch entwickelten Korrektur-Algorithmen beanspruchte hierbei die meiste Zeit und wurde iterativ in einem prototypischen Verfahren verbessert. Das Resultat besteht aus einem *restriktiven* Algorithmus, der eine logisch nachvollziehbare und vollständig durchgängige Trajektorie liefert und aus einem *fehlertoleranten* Algorithmus, welcher sich bei einer mehrdeutigen Pfadentwicklung für den Wahrscheinlichsten entscheidet und eine geringere Deadlockgefahr bietet.

Die Leistungsfähigkeit der beiden Algorithmen wurde anhand der Evaluation (siehe Kap. 5) überprüft. Hierfür mussten repräsentative Szenarien entwickelt und mit unterschiedlichen Toleranzen durchlaufen werden. Um die Messergebnisse vergleichen zu können, wurden die einzelnen Szenarien mit dem simulierten Ortungssystem vorgegeben und aufgezeichnet. Jeder Algorithmus erhielt somit das identische Testszenario und lieferte korrigierte Positionen, deren Abstände zur tatsächlichen Trajektorie für den Vergleich herangezogen wurden. Die Gegenüberstellung der jeweiligen Messreihen liefert eine erste Tendenz, für welche Toleranzen das entsprechende Korrekturverfahren brauchbare Resultate liefert.

6.2. Bewertung

Das Ziel der Masterarbeit war es, eine Präzisionssteigerung eines bestehenden Ortungssystems zu realisieren (siehe Kapitel 1.2). Auf Grundlage eines Gebäudemodells wird eine Abfolge von Ortungspunkten (Trajektorie) zur Laufzeit auf logische Korrektheit überprüft und ggf. korrigiert.

Die Ortungsdaten werden ohne Wissen über das Ortungssystem in Form eines Koordinatentupels empfangen und verarbeitet. Dies erleichtert ein Auswechseln des verwendeten Ortungssystems und entspricht den formulierten Zielen.

Die zu erreichende Präzision, nämlich der Erkennung der aktuellen Position auf räumlich und semantisch korrekten Kontext, konnte für die Mehrheit der Testfälle erfüllt werden. Sie hängt jedoch mit der gewählten Toleranz des Ortungssystems zusammen und ist nicht beliebig vergrößerbar. Für Korrekturen innerhalb von Räumen fehlen dem selbst entwickelten Algorithmus die restriktiven Elemente. Die korrigierte Position wird jedoch nicht als Fläche (der aktuell betretene Raum) angegeben, sondern in dem legal betretenen Raum so nahe wie möglich an die unkorrigierte Position verschoben, in der Annahme, dass die tatsächliche Position ebenfalls in der Nähe von P_u liegen muss. Dies ist natürlich nur eine grobe Näherung, führte in der Evaluation bei einer Toleranz von 5 m jedoch zu einer positiven Korrektur.

Neben dem räumlichen Abgleich mit einem Gebäudemodell erfolgt zusätzlich eine Plausibilitätsüberprüfung auf Basis eines einfachen Bewegungsmodells. Dies führt dazu, dass Messungen nur in erreichbarer Entfernung zugelassen werden und alternative Korrekturpositionen logisch nachvollziehbar sind.

Der entwickelte Korrektur-Algorithmus liefert keine direkte Präzisionssteigerung innerhalb von Räumen. Dies ist erst bei einer Bewegung durch mehrere aufeinanderfolgende Räume innerhalb eines Gebäudes der Fall. Im optimalen Fall wird die tatsächlich zurückgelegte Raumabfolge erkannt und die Trajektorie bestmöglich in die Räume verschoben. Ausreißer in benachbarte und logisch unerreichbare Räume werden korrigiert und verdeutlichen den zurückgelegten Pfad im Gebäude.

Diese Korrektur kann eine Indoornavigation unterstützen, indem im Vorfeld ungültige Ortungspunkte gefiltert werden. Ein weiteres Anwendungsgebiet besteht für die Analyse der Trajektorien. Sollten die Ortungspunkte entsprechend des Gebäude- und Bewegungsmodells korrigiert worden sein, so kann eine Mustererkennung den zurückgelegten Pfad leichter kategorisieren und daraus eine zuvor definierte Aktivität ableiten. Auf diesem *Activity Pattern Recognition* können dann *high-level*-Interpretationen durchgeführt werden.

6.3. Ausblick

Die in der Evaluation stellvertretenden Toleranzen von 1, 5 und 10 Metern haben gezeigt, dass eine zufriedenstellende Korrektur für fünf Meter noch erreicht wurde. Bei zehn Metern jedoch zu häufig die tatsächliche Trajektorie nicht mehr erkannt wird. Es sollten daher noch

mehr Toleranzen feingranularer und mit umfangreicheren Durchläufen automatisiert überprüft werden, um die Schwelle der Korrigierbarkeit weiter einzugrenzen.

Bei dem simulierten Ortungssystem wurde eine gleichmäßige Zufallsverteilung mit einem maximalen Radius für den Jitter verwendet. Dieses Verhalten muss nicht zwangsläufig einem realen Ortungssystem entsprechen. Daher wäre es interessant die Tests zunächst mit einer Normal- bzw. Gauß-Verteilung für die simulierte Streuung zu wiederholen und dessen maximale korrigierbare Toleranz zu bestimmen. In einem weiteren Schritt sollte dann ein reales Ortungssystem durch die Algorithmen korrigiert werden, um anhand dieser Testdaten weitere Kalibrierungen und Verbesserungen an den Verfahren vorzunehmen.

Für die generelle Weiterentwicklung der Algorithmen kann eine Trajektorienvorhersage (Path-Prediction) einen Mehrwert liefern. Ausgehend von der bisher zurückgelegten Strecke und deren Raumwechseln kann eine generelle Bewegungsrichtung und Geschwindigkeit abgeleitet werden. Der jeweils aktuell eintreffende Ortungspunkt wird entsprechend dieser groben Pfadentwicklung zusätzlich in seiner Wahrscheinlichkeit bewertet und korrigiert. Eine Betrachtung von mehreren unkorrigierten Ortungspunkten in ihrem Gesamtkontext würde eine Tendenz der Bewegungsentwicklung ermöglichen und lokale Ausreißer minimieren. Um eine festgefahrene Fehlkorrektur aufzulösen würde sich eine kooperative Integration von Sensordaten (siehe [Ruser und Puente León \(2007\)](#)) anbieten, mit deren Hilfe anhand von Stützstellen die Korrektur repositioniert werden könnte.

Da die Verfügbarkeit von nutzbaren digitalen Gebäudemodellen gering und die Erstellung sehr zeitaufwändig ist, bietet es sich an, diesen Schritt zu automatisieren. [Schafer u. a. \(2011\)](#) schlagen vor, die nötigen Informationen aus einem CAD-Plan des Gebäudes zu extrahieren. Diese Gebäudepläne können im Anschluss nicht nur für die Ortungsverbesserung verwendet werden, sondern bieten einen nutzbaren Mehrwert für andere Bereiche wie z. B. das Gebäudemanagement.

Im Laufe der Entwicklung haben sich einige Parallelen zu dem in Kapitel [2.2.3](#) vorgestellten Backtracking Particle Filter gezeigt. Bei der eigenen Realisierung wird nur jeweils ein *Partikel* verwendet und dieses dann im Raum zur wahrscheinlichsten Position verschoben. Der BPF setzt auf das Prinzip des *Survival of the fittest* und generiert ein Cluster von Partikeln, für welche die jeweils nächste Position mit einer zufälligen Streuung berechnet wird. Illegale Partikel werden rückwirkend entfernt und sorgen bei einer mehrdeutigen Raumentwicklung für die nachträgliche Erkennung des korrekten Pfades. Anhand der Vielzahl an Partikeln und der zufälligen Berechnung der nächsten Position werden mit hoher Wahrscheinlichkeit alle möglichen Bewegungspfade automatisch abgedeckt. Eine aufwändige rekursive Berechnung der erreichbaren Räume entfällt, da ungültige Partikel einfach gelöscht werden und den Schwer-

punkt des Clusters verschieben. Das grundlegende Problem, dass die Korrektur zur Laufzeit erfolgt und erst zu einem späteren Zeitpunkt eine alternative Route wahrscheinlicher wird, ist mit diesem Ansatz jedoch auch nicht gelöst. Die Korrektur wird rückwirkend verändert und führt bei der aktuellen Position zu einem Sprung in der Trajektorie (ähnlich wie beim eigenen *fehlertoleranten* Ansatz).

Es wäre daher interessant zu sehen, wie sich der BPF im Vergleich zur eigenen Lösung bei den gleichen Testszenarien verhält und wo dessen Stärken und Schwächen liegen. Hierfür müsste das ISIS-Framework erweitert werden, um einen Partikelfilter zu integrieren und zusätzlich die genaue Funktionsweise und der Programmcode vom Entwickler in Erfahrung gebracht werden.

A. Anhang

A. Anhang

$\overline{P_t P_u}$	$\overline{P_t P_k}$	$\overline{P_t P_u} - \overline{P_t P_k}$	$\frac{\overline{P_t P_k}}{\overline{P_t P_u}}$
1,612451549659709	1,3935205879585875	0,21893096170112147	0,8642247813602061
2,445914961727001	1,6656965070725889	0,780218454654412	0,6810116186118267
0,34999999999999787	0,34999999999999787	0,0	1,0
2,3900836805434222	0,6699999999999982	1,720083680543424	0,2803249130790531
1,9235384061671348	1,9235384061671348	0,0	1,0
2,5460754112948023	2,5460754112948023	0,0	1,0
1,9506409203131152	1,9506409203131152	0,0	1,0
2,7280945731407473	1,6368872899500466	1,0912072831907007	0,6000111968499549
1,6031219541881412	1,6031219541881412	0,0	1,0
2,1982947936980595	1,308806983217145	0,8894878104809145	0,5953737355741163
0,7516648189186452	0,7516648189186452	0,0	1,0
1,2165525060596467	1,2165525060596467	0,0	1,0
2,0081085628023194	2,0081085628023194	0,0	1,0
1,610124218810462	1,4504125045756087	0,15971171423485342	0,9008078306201454
2,7244265451650542	2,2806131704936923	0,44381337467136195	0,8370984251864003
1,6867127793433025	1,6867127793433025	0,0	1,0
1,2103718436910198	1,2103718436910198	0,0	1,0
2,25	1,095991073608399	1,154008926391601	0,48710714382595516
2,0303940504246936	1,2611504178155666	0,769243632609127	0,6211357926072401
3,1468237955119123	1,1989209502467812	1,947902845265131	0,3809939889092982
1,6985287751463045	1,6985287751463045	0,0	1,0
2,5303161857759995	2,239357058300329	0,29095912747567043	0,8850107630377273
1,7356554957709778	1,7207207750624107	0,01493472070856705	0,9913953427134842
2,2638462845343543	2,2509102920058375	0,012935992528516849	0,9942858344151323
2,5347583711273165	1,9247078993129096	0,6100504718144069	0,7593259859545938
2,6655205870523706	2,466443643002568	0,19907694404980258	0,9253140474634454
2,258871399615304	2,258871399615304	0,0	1,0
2,57147817412476	2,57147817412476	0,0	1,0
2,7504545078950127	2,7504545078950127	0,0	1,0
2,631539473388153	2,296760722543287	0,3347787508448663	0,8727821663971346
0,5024937810560443	1,4449352352945348	-0,9424414542384906	2,8755285931257686
2,830194339616983	2,927091705391856	-0,09689736577487285	1,034237000766522
2,136000936329383	2,860977782044435	-0,724976845715052	1,3394084868524874
0,3640054944640277	2,317826975192504	-1,9538214807284762	6,3675604089584965
1,806931099959265	1,806931099959265	0,0	1,0
1,8062391868188452	1,8062391868188452	0,0	1,0
2,6400757564888164	2,6400757564888164	0,0	1,0
2,7973201461398713	2,7973201461398713	0,0	1,0
1,7804493814764852	1,7804493814764852	0,0	1,0
1,553222456700906	1,553222456700906	0,0	1,0
1,3901438774457835	1,133710721480571	0,25643315596521243	0,8155348089318809

Tabelle A.1.: Messergebnisse für den 2. Algorithmus- Szenario 3 - 5 m Toleranz - Durchlauf 4

Literaturverzeichnis

- [10gen, Inc. 2013] 10GEN, INC.: *mongoDB*. Webseite. 2013. – URL <http://www.mongodb.org/downloads>. – Letzter Aufruf am 26. April 2013
- [Abdulrahim u. a. 2010] ABDULRAHIM, K. ; HIDE, C. ; MOORE, T. ; HILL, C.: Aiding MEMS IMU with building heading for indoor pedestrian navigation. In: *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2010 IEEE* (Veranst.), 2010, S. 1–6
- [Action 1999] ACTION, C.: *Digital Mobile Radio Towards Future Generation Systems: Final Report*. Directorate General Telecommunications, Information Society, Information Market, and Exploitation Research, 1999
- [Angermann u. a. 2008] ANGERMANN, M. ; KHIDER, M. ; ROBERTSON, P.: Towards operational systems for continuous navigation of rescue teams. In: *Position, Location and Navigation Symposium, 2008 IEEE/ION*, may 2008, S. 153 –158
- [Apache.org 2013] APACHE.ORG: *ActiveMQ*. Webseite. 2013. – URL <http://activemq.apache.org/>. – Letzter Aufruf am 26. April 2013
- [Ascher u. a. 2010] ASCHER, C. ; KESSLER, C. ; WANKERL, M. ; TROMMER, G.F.: Dual IMU Indoor Navigation with particle filter based map-matching on a smartphone. In: *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, sept. 2010, S. 1 –5
- [Autodesk 2012] AUTODESK: *Revit-Produkte*. Webseite. 2012. – URL <http://www.autodesk.de/products/autodesk-revit-family/overview>. – Letzter Aufruf am 26. April 2013
- [Bahl und Padmanabhan 2000] BAHL, P. ; PADMANABHAN, V.N.: RADAR: an in-building RF-based user location and tracking system. In: *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* Bd. 2, 2000, S. 775 –784 vol.2. – ISSN 0743-166X

- [Bahl u. a. 2000] BAHL, Paramvir ; ; BAHL, Paramvir ; PADMANABHAN, Venkata N. ; BALACHANDRAN, Anand: Enhancements to the RADAR User Location and Tracking System / Microsoft Research. 2000. – Forschungsbericht
- [Bauhaus-Universität Weimar 2013] BAUHAUS-UNIVERSITÄT WEIMAR: *Open IFC Tools*. Webseite. 2013. – URL http://www.openifctools.org/Open_IFC_Tools/Home.html. – Letzter Aufruf am 26. April 2013
- [Beauregard 2007] BEAUREGARD, S.: Omnidirectional Pedestrian Navigation for First Responders. In: *Positioning, Navigation and Communication, 2007. WPNC '07. 4th Workshop on, march 2007*, S. 33 –36
- [Beauregard u. a. 2008] BEAUREGARD, S. ; WIDYAWAN ; KLEPAL, M.: Indoor PDR performance enhancement using minimal map information and particle filters. In: *Position, Location and Navigation Symposium, 2008 IEEE/ION, may 2008*, S. 141 –147
- [Beauregard 2009] BEAUREGARD, Stephane: *Infrastructureless Pedestrian Positioning*, University of Bremen, Diplomarbeit, 2009. – URL <http://elib.suub.uni-bremen.de/diss/docs/00011529.pdf>. – Letzter Aufruf am 26. April 2013
- [Bernoulli u. a. 2008] BERNOULLI, T. ; GLANZER, G. ; WIESSFLECKER, T. ; SCHÜTZ, R. ; WALDER, U.: A building information model for a context-adaptive disaster management system. In: *Intelligent Computing in Engineering 1* (2008), S. 402–409
- [Borenstein und Ojeda 2010] BORENSTEIN, J. ; OJEDA, L.: Heuristic drift elimination for personnel tracking systems. In: *Journal of Navigation* 63 (2010), Nr. 04, S. 591–606
- [von Both und Koch 2009] BOTH, P. von ; KOCH, V. ; UNIVERSITÄT KARLSRUHE (TH) (Hrsg.): *Forum Bauinformatik 2009*. Universitätsverlag Karlsruhe, 2009. – ISBN: 978-3-86644-396-9
- [Broadcom] BROADCOM: *New Location Architecture With BCM4752*. Webseite. – URL <http://www.broadcom.com/products/features/GNSS.php>. – Letzter Aufruf am 26. April 2013
- [Bruning u. a. 2007] BRUNING, S. ; ZAPOTOCZKY, J. ; IBACH, P. ; STANTCHEV, V.: Cooperative Positioning with MagicMap. In: *Positioning, Navigation and Communication, 2007. WPNC '07. 4th Workshop on, march 2007*, S. 17 –22
- [Chai u. a. 2011] CHAI, W. ; ZHOU, J. ; CHEN, C. ; NIES, H. ; LOFFELD, O.: Continuous Indoor Localization and Navigation Based on Low-cost INS/Wi-Fi Integration. In: *2011 International Conference on Indoor Positioning and Indoor Navigation (IPIN 2011)*, 2011

- [Chen 2003] CHEN, Z.: Bayesian filtering: From Kalman filters to particle filters, and beyond. In: *Statistics 1* (2003), S. 1–69
- [CityGML] CITYGML: *Features*. Webseite. – URL <http://www.citygml.org/index.php?id=1532>. – Letzter Aufruf am 26. April 2013
- [Daum 2005] DAUM, F.: Nonlinear filters: beyond the Kalman filter. In: *Aerospace and Electronic Systems Magazine, IEEE 20* (2005), aug., Nr. 8, S. 57 –69. – ISSN 0885-8985
- [DoD 2008] DoD, U S.: Global Positioning System Standard Positioning Service Performance Standard. In: *Distribution 4th Edition* (2008), Nr. September, S. 160. – URL <http://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf>. – Letzter Aufruf am 26. April 2013
- [Dorveaux und Petit 2011] DORVEAUX, E. ; PETIT, N.: Harmonization of a multi-sensor navigation system. In: *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, sept. 2011, S. 1 –7
- [Driewer u. a. 2004] DRIEWER, F. ; BAIER, H. ; SCHILLING, K. ; PAVLICEK, J. ; PREUCIL, L. ; RUANGPAYOONGSAK, N. ; ROTH, H. ; SAARINEN, J. ; SUOMELA, J. ; HALME, A. u. a.: Hybrid telematic teams for search and rescue operations. In: *IEEE International Workshop on Safety, Security, and Rescue Robotics, SSRR 2004*, 2004
- [Eisner u. a. 2011] EISNER, J. ; FUNKE, S. ; HERBST, A. ; SPILLNER, A. ; STORANDT, S.: Algorithms for matching and predicting trajectories. In: *Proc. of the 13th Workshop on Algorithm Engineering and Experiments (ALENEX) Citeseer* (Veranst.), 2011
- [Fang u. a. 2005] FANG, Lei ; ANTSAKLIS, P.J. ; MONTESTRUQUE, L.A. ; McMICKELL, M.B. ; LEMMON, M. ; SUN, Yashan ; FANG, Hui ; KOUTROULIS, I. ; HAENGGI, M. ; XIE, Min ; XIE, Xiaojuan: Design of a wireless assisted pedestrian dead reckoning system - the NavMote experience. In: *Instrumentation and Measurement, IEEE Transactions on 54* (2005), dec., Nr. 6, S. 2342 – 2358. – ISSN 0018-9456
- [Fischer 2009] FISCHER, J.: NFC in cell phones: The new paradigm for an interactive world [Near-Field Communications]. In: *Communications Magazine, IEEE 47* (2009), June, Nr. 6, S. 22 –28. – ISSN 0163-6804
- [Gilliéron u. a. 2004] GILLIÉRON, Pierre-Yves ; BÜCHEL, Daniela ; SPASSOV, Ivan ; MERMINOD, Bertrand: Indoor Navigation Performance Analysis. In: *Proceedings of the 8th European Navigation Conference GNSS*, 2004

- [Gregor 2006] GREGOR, Sebastian: *Entwicklung einer Hardwareplattform für die Ermittlung von Positionsdaten innerhalb von Gebäuden*, Hochschule für Angewandte Wissenschaften (HAW) Hamburg, Bachelorarbeit, 2006. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/gregor.pdf>. – Letzter Aufruf am 26. April 2013
- [Grubbs 1969] GRUBBS, F.E.: Procedures for detecting outlying observations in samples. In: *Technometrics* 11 (1969), Nr. 1, S. 1–21
- [Hartung u. a. 2005] HARTUNG, J. ; ELPALT, B. ; KLÖSENER, K.H.: *Statistik*. Oldenbourg Wissenschaftsverlag, 2005. – URL <http://books.google.de/books?hl=en&lr=&id=DKcVnmCeskkC&oi=fnd&pg=PA1&dq=hartung+statistik&ots=VmyCzKtqAw&sig=hEe6jbkRwnZFigYkHuAt8PeInp0#v=onepage&q&f=false>. – Letzter Aufruf am 26. April 2013
- [Hattori u. a. 2011] HATTORI, Kiyohiko ; NAKAMURA, Kyohei ; NAKAJIMA, Nobuo ; FUJI, Tetsuya ; KADO, Youiti ; BING, Zhang ; TAKADAMA, Keiki: *Research on Improvement of the Hybrid Location Estimation System Using 2D Marker and WiFi*. Webseite. 2011. – URL http://ipin2011.dsi.uminho.pt/PDFs/Shortpaper/117_Short_Paper.pdf. – Letzter Aufruf am 26. April 2013
- [Hicks 1993] HICKS, S.: Advanced Cruise Missile guidance system description. In: *Aerospace and Electronics Conference, 1993. NAECON 1993., Proceedings of the IEEE 1993 National*, may 1993, S. 355 –361 vol.1
- [Hochschule Luzern 2012] HOCHSCHULE LUZERN: *iHomeLab - Schweizer Denkfabrik Forschungslabor für Gebäudeintelligenz*. Webseite. 4 2012. – URL http://www.ihomelab.ch/fileadmin/Dateien/PDF/Publikationen/2012/DBZ_Deutsche_BauZeitschrift_Bautechnik_17210_292_1_.pdf. – Letzter Aufruf am 26. April 2013
- [Humboldt-Universität zu Berlin a] HUMBOLDT-UNIVERSITÄT ZU BERLIN: *MagicMap*. Webseite. – URL <http://www.magicmap.de>. – Letzter Aufruf am 26. April 2013
- [Humboldt-Universität zu Berlin b] HUMBOLDT-UNIVERSITÄT ZU BERLIN: *MagicMap Architektur*. Webseite. – URL http://wiki.informatik.hu-berlin.de/nomads/index.php/Bild:MagicMapArchitektur_Schnittstellen.png. – Letzter Aufruf am 26. April 2013

- [Humboldt-Universität zu Berlin c] HUMBOLDT-UNIVERSITÄT ZU BERLIN: *MagicMap Pfadbasierte Ortung*. Webseite. – URL http://wiki.informatik.hu-berlin.de/nomads/index.php/Pfadbasierte_Ortung. – Letzter Aufruf am 26. April 2013
- [Humboldt-Universität zu Berlin d] HUMBOLDT-UNIVERSITÄT ZU BERLIN: *MagicMap Plugins*. Webseite. – URL http://wiki.informatik.hu-berlin.de/nomads/index.php/MagicMap_Plugins. – Letzter Aufruf am 26. April 2013
- [Humboldt-Universität zu Berlin e] HUMBOLDT-UNIVERSITÄT ZU BERLIN: *MagicMap Umgebungsmodell-Plugin*. Webseite. – URL <http://wiki.informatik.hu-berlin.de/nomads/index.php/Umgebungsmodell-Plugin>. – Letzter Aufruf am 26. April 2013
- [Ibach u. a. 2004] IBACH, P. ; HÜBNER, T. ; SCHWEIGERT, M.: MagicMap - Kooperative Positionsbestimmung über WLAN. In: *Chaos Communication Congress, Berlin* Bd. 27, 2004, S. 29
- [Jimenez u. a. 2010] JIMENEZ, AR ; SECO, F. ; PRIETO, JC ; GUEVARA, J.: Indoor pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU. In: *Positioning Navigation and Communication (WPNC), 2010 7th Workshop on IEEE* (Veranst.), 2010, S. 135–143
- [Jimenez u. a. 2011] JIMENEZ, A.R. ; SECO, F. ; ZAMPELLA, F. ; PRIETO, J.C. ; GUEVARA, J.: Improved Heuristic Drift Elimination (iHDE) for pedestrian navigation in complex buildings. In: *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, sept. 2011, S. 1 –8
- [Johnson und Montgomery 2008] JOHNSON, A.E. ; MONTGOMERY, J.F.: Overview of Terrain Relative Navigation Approaches for Precise Lunar Landing. In: *Aerospace Conference, 2008 IEEE*, march 2008, S. 1 –10. – ISSN 1095-323X
- [Kaiser u. a. 2011a] KAISER, S. ; KHIDER, M. ; ROBERTSON, P.: A maps-based angular PDF for navigation systems in indoor and outdoor environments. In: *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, sept. 2011, S. 1 –7
- [Kaiser u. a. 2011b] KAISER, Susanna ; KHIDER, Mohammed ; ROBERTSON, Patrick: A human motion model based on maps for navigation systems. In: *EURASIP Journal on Wireless Communications and Networking* 2011 (2011), Nr. 1, S. 60. – URL <http://jwcn.erasipjournals.com/content/2011/1/60>. – Letzter Aufruf am 26. April 2013. – ISSN 1687-1499

- [Karstaedt] KARSTAEDT, Bastian: *LivingPlace Wiki - Ubisense Mockup*. Webseite. – URL http://livingplace.informatik.haw-hamburg.de/wiki/index.php/Tools_und_Mockups#Ubisense_Mockup. – Letzter Aufruf am 26. April 2013
- [Karstaedt 2012] KARSTAEDT, Bastian: *Kontextinterpretation in Smart Homes auf Basis semantischer 3D Gebäudemodelle*, Hochschule für Angewandte Wissenschaften (HAW) Hamburg, Mastersthesis, 2012. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/karstaedt.pdf>. – Letzter Aufruf am 26. April 2013
- [Khider u. a. 2008] KHIDER, M. ; KAISER, S. ; ROBERTSON, P. ; ANGERMANN, M.: A novel movement model for pedestrians suitable for personal navigation. In: *ION NTM*, 2008, S. 819–827
- [Koyuncu und Yang 2010] KOYUNCU, H. ; YANG, H.: A Survey of Indoor Positioning and Object Locating Systems. In: *IJCSNS International Journal of Computer Science and Network Security* 10th Edition (2010), S. 121. – URL http://paper.ijcsns.org/07_book/201005/20100518.pdf. – Letzter Aufruf am 26. April 2013
- [Krach und Roberston 2008] KRACH, B. ; ROBERSTON, P.: Cascaded estimation architecture for integration of foot-mounted inertial sensors. In: *Position, Location and Navigation Symposium, 2008 IEEE/ION*, may 2008, S. 112 –119
- [Krach und Robertson 2008] KRACH, B. ; ROBERTSON, P.: Integration of foot-mounted inertial sensors into a Bayesian location estimation framework. In: *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on*, march 2008, S. 55 –61
- [Levi und Judd 1996] LEVI, R.W. ; JUDD, T.: *Dead reckoning navigational system using accelerometer to measure foot impacts*. Dezember 10 1996. – US Patent 5,583,776
- [Lin und Lin 2005] LIN, Tsung-Nan ; LIN, Po-Chiang: Performance comparison of indoor positioning techniques based on location fingerprinting in wireless networks. In: *Wireless Networks, Communications and Mobile Computing, 2005 International Conference on* Bd. 2, june 2005, S. 1569 – 1574 vol.2
- [Liu u. a. 2007] LIU, Hui ; DARABI, H. ; BANERJEE, P. ; LIU, Jing: Survey of Wireless Indoor Positioning Techniques and Systems. In: *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 37 (2007), nov., Nr. 6, S. 1067 –1080. – ISSN 1094-6977

- [Manh-Hung Vo Le 2009] MANH-HUNG VO LE, Nathan Michael W.: *Indoor Navigation System for Handheld Devices*. Webseite. 10 2009. – URL http://www.wpi.edu/Pubs/E-project/Available/E-project-102209-164024/unrestricted/Indoor_Navigation_System_for_Handheld_Devices.pdf. – Letzter Aufruf am 26. April 2013
- [Martin u. a. 2006] MARTIN, J.D. ; KRÖSCHE, J. ; BOLL, S.: Dynamic GPS-position correction for mobile pedestrian navigation and orientation. In: *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication 2006 (WPNC06)*, 2006
- [Mautz 2011] MAUTZ, Rainer: *IPIN 2011 Keynote - Overview of Indoor Positioning Technologies*. Webseite. 2011. – URL http://www.geometh.ethz.ch/people/rmautz/IPIN_Keynote_Mautz_2011.pdf. – Letzter Aufruf am 26. April 2013
- [Mestre u. a. 2011] MESTRE, P. ; SERODIO, C. ; COUTINHO, L. ; REIGOTO, L. ; MATIAS, J.: Hybrid technique for fingerprinting using IEEE802.11 wireless networks. In: *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, sept. 2011, S. 1 –7
- [Microsoft Research] MICROSOFT RESEARCH: *RADAR*. Webseite. – URL <http://research.microsoft.com/en-us/projects/radar/>. – Letzter Aufruf am 26. April 2013
- [Oh u. a. 2004] OH, Sang M. ; TARIQ, S. ; WALKER, B.N. ; DELLAERT, F.: Map-based priors for localization. In: *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on* Bd. 3, sept.-2 oct. 2004, S. 2179 – 2184 vol.3
- [OpenStreetMap Wiki] OPENSTREETMAP WIKI: *IndoorOSM*. Webseite. – URL <http://wiki.openstreetmap.org/wiki/IndoorOSM>. – Letzter Aufruf am 26. April 2013
- [Retscher 2007] RETSCHER, Günther: Test and Integration of Location Sensors for a Multi-sensor Personal Navigator. In: *The Journal of Navigation* 60 (2007), Nr. 01, S. 107–117. – URL <http://dx.doi.org/10.1017/S037346330700402X>. – Letzter Aufruf am 26. April 2013
- [Ristic u. a. 2004] RISTIC, B. ; ARULAMPALAM, S. ; GORDON, N.: *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House Publishers, 2004
- [Ruser und Puente León 2007] RUSER, Heinrich ; PUENTE LEÓN, Fernando: Informationsfusion - Eine Übersicht (Information Fusion - An Overview). In: *tm - Technisches Messen* 74 (2007),

- März, Nr. 3, S. 93–102. – URL <http://dx.doi.org/10.1524/teme.2007.74.3.93>. – Letzter Aufruf am 26. April 2013. – ISSN 0171-8096
- [Saarinen und Heikkila 2005] SAARINEN, J. ; HEIKKILA, S.: Laser based personal navigation system. In: *Computational Intelligence in Robotics and Automation, 2005. CIRA 2005. Proceedings. 2005 IEEE International Symposium on*, june 2005, S. 315 – 320
- [Schafer u. a. 2011] SCHAFFER, M. ; KNAPP, C. ; CHAKRABORTY, S.: Automatic generation of topological indoor maps for real-time map-based localization and tracking. In: *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, sept. 2011, S. 1 –8
- [Serodio u. a. 2011] SERODIO, L.C. ; PINTO, H. ; MESTRE, P.: A comparison of multiple algorithms for fingerprinting using IEEE802. 11. In: *Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering*, 2011
- [Spassov u. a. 2007] SPASSOV, I. ; MERMINOD, B. ; BIERLAIRE, M.: *Algorithms for map-aided autonomous indoor pedestrian positioning and navigation*, Ph. D. thesis, Ecole polytechnique fédérale de Lausanne (EPFL), Dissertation, 2007
- [Sternberg 2010] STERNBERG, Prof. H.: *IPIN: Foot Mounted Pedestrian Navigation*. Webseite. 2010. – URL http://www.geometh.ethz.ch/ipin/program/IPIN2010_Abstract_foot_mounted_Pedestrian_Navigation.pdf. – Letzter Aufruf am 26. April 2013
- [Strauss u. a. 1999] STRAUSS, O. ; COMBY, F. ; ALDON, M.J.: Multibeam sonar image matching for terrain-based underwater navigation. In: *OCEANS '99 MTS/IEEE. Riding the Crest into the 21st Century* Bd. 2, 1999, S. 882 –887 vol.2
- [Thiagarajan u. a. 2011] THIAGARAJAN, A. ; RAVINDRANATH, L. ; BALAKRISHNAN, H. ; MADDEN, S. ; GIROD, L.: Accurate, low-energy trajectory mapping for mobile devices. In: *Proceedings of the 8th USENIX conference on Networked systems design and implementation* USENIX Association (Veranst.), 2011, S. 20–20
- [TU Graz 2009] TU GRAZ: *CADMS Description*. Webseite. 2009. – URL http://portal.tugraz.at/portal/page/portal/Files/i2330/cadms/CADMS_description.pdf. – Letzter Aufruf am 26. April 2013
- [Ubisense 2013] UBISENSE: *RTLS Solutions*. Webseite. 2013. – URL <http://www.ubisense.net/en/rtls-solutions/>. – Letzter Aufruf am 26. April 2013

- [Vazirani 2004] VAZIRANI, Vijay V.: *Approximation Algorithms*. Springer, März 2004. – URL <http://www.worldcat.org/isbn/3540653678>. – Letzter Aufruf am 26. April 2013. – ISBN 3540653678
- [Vildjiounaite u. a. 2002] VILDJIOUNAITE, Elena ; MALM, Esko-Juhani ; KAARTINEN, Jouni ; ALAHUHTA, Petteri: Location Estimation Indoors by Means of Small Computing Power Devices, Accelerometers, Magnetic Sensors, and Map Knowledge. In: MATTERN, Friedemann (Hrsg.) ; NAGHSHINEH, Mahmoud (Hrsg.): *Pervasive Computing* Bd. 2414. Springer Berlin / Heidelberg, 2002, S. 5–12. – URL http://dx.doi.org/10.1007/3-540-45866-2_17. – Letzter Aufruf am 26. April 2013. – ISBN 978-3-540-44060-4
- [Vividsolutions 2013] VIVIDSOLUTIONS: *Java Topology Suite*. Webseite. 2013. – URL <http://www.vividsolutions.com/jts/JTSHome.htm>. – Letzter Aufruf am 26. April 2013
- [Voskuhl 2012] VOSKUHL, Sören: *Modellunabhängige Kontextinterpretation in einer Smart Home Umgebung*, Hochschule für Angewandte Wissenschaften (HAW) Hamburg, Masterthesis, 2012. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/voskuhl.pdf>. – Letzter Aufruf am 26. April 2013
- [Walder 2011] WALDER, 8561 Söding A.: *Method and system for detection of a zero velocity state of an object*. November 2011. – URL <http://www.freepatentsonline.com/EP2386828A1.html>
- [Wang 2011] WANG, Hui: Bayesian radio map learning for robust indoor positioning. In: *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, sept. 2011, S. 1 –6
- [Wang u. a. 2007] WANG, Hui ; LENZ, H. ; SZABO, A. ; BAMBERGER, J. ; HANEBECK, U.D.: WLAN-Based Pedestrian Tracking Using Particle Filters and Low-Cost MEMS Sensors. In: *Positioning, Navigation and Communication, 2007. WPNC '07. 4th Workshop on*, march 2007, S. 1 –7
- [Wehrle 2011] WEHRLE, Jo Agila Bitsch Link;Paul S.: FootPath: Accurate Map-based Indoor Navigation Using Smartphones. In: *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on* 1 (2011), S. 1 – 8. – URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=6071934. – Letzter Aufruf am 26. April 2013
- [White u. a. 2000] WHITE, C.E. ; BERNSTEIN, D. ; KORNHAUSER, A.L.: Some map matching algorithms for personal navigation assistants. In: *Transportation Research Part C: Emerging*

- Technologies* 8 (2000), Dezember, Nr. 1-6, S. 91–108. – URL [http://dx.doi.org/10.1016/S0968-090X\(00\)00026-7](http://dx.doi.org/10.1016/S0968-090X(00)00026-7). – Letzter Aufruf am 26. April 2013. – ISSN 0968090X
- [Widyawan u. a. 2008a] WIDYAWAN ; KLEPAL, M. ; BEAUREGARD, S.: A Backtracking Particle Filter for fusing building plans with PDR displacement estimates. In: *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on*, march 2008, S. 207 –212
- [Widyawan u. a. 2007] WIDYAWAN ; KLEPAL, M. ; PESCH, D.: A Bayesian Approach for RF-Based Indoor Localisation. In: *Wireless Communication Systems, 2007. ISWCS 2007. 4th International Symposium on*, oct. 2007, S. 133 –137
- [Widyawan u. a. 2008b] WIDYAWAN ; KLEPAL, Martin ; BEAUREGARD, Stéphane: A novel backtracking particle filter for pattern matching indoor localization. In: *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*. New York, NY, USA : ACM, 2008 (MELT '08), S. 79–84. – URL <http://doi.acm.org/10.1145/1410012.1410031>. – Letzter Aufruf am 26. April 2013. – ISBN 978-1-60558-189-7
- [Wießflecker 2009] WIESSFLECKER, T.: Neue Wege bei der Navigation in Gebäuden. In: *Forum Bauinformatik 2009: 23. bis 25. September 2009, Universität Karlsruhe (TH)* KIT Scientific Publishing (Veranst.), 2009, S. 335
- [Wießflecker u. a. 2008] WIESSFLECKER, T. ; BERNOULLI, T. ; GLANZER, G. ; SCHÜTZ, R. ; WALDER, U.: Context-Adaptive Building Information for Disaster Management. In: *European Conference on Product and Process Modelling, Sophia Antipolis, France*, 2008, S. 351–361
- [Woodman 2007] WOODMAN, O.J.: An introduction to inertial navigation. In: *University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696* 696 (2007), S. 1–37
- [Woodman und Harle 2008] WOODMAN, Oliver ; HARLE, Robert: Pedestrian localisation for indoor environments. In: *Proceedings of the 10th international conference on Ubiquitous computing*. New York, NY, USA : ACM, 2008 (UbiComp '08), S. 114–123. – URL <http://doi.acm.org/10.1145/1409635.1409651>. – Letzter Aufruf am 26. April 2013. – ISBN 978-1-60558-136-1

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 2. Mai 2013

Johannes Meyer