

Institut für Parallele und Verteilte Systeme

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Diplomarbeit Nr. 3485

Erfassung von Innenraummodellen mittels Smartphones

Christoph Dibak

Studiengang: Informatik
Prüfer: Prof. Dr. Kurt Rothermel
Betreuer: Dipl.-Inf. Damian Philipp

Beginn am: 13. Mai 2013
Beendet am: 12. November 2013

CR-Nummer: C.2.4

Kurzfassung

Unter Verwendung von aufgezeichneten Bewegungsspuren eines Inertialsensors ist es möglich, ein Innenraummodell eines Gebäudes zu generieren. Das Innenraummodell unterscheidet hierbei Räume und Korridore.

Die Bewegungsspuren werden durch auf dem Fuß platzierte Inertialsensoren per ZUPT erfasst. Es wird von einer Vielzahl an Benutzern ausgegangen, welche sich in alltäglichen Situationen durch Gebäude bewegen und mit den Sensoren ihres Smartphones opportunistisch Daten erfassen.

Bewegungsspuren werden in gerade Segmente unterteilt. Durch eine Äquivalenzrelation wird festgestellt, ob sich der Benutzer beim Erfassen der Spuren auf demselben Korridor befunden hatte. Die Geometrie von Korridoren wird durch Quantile und die empirische Verteilungsfunktion bestimmt. Durch die Ausrichtung der Spuren anhand der Geometrie der Korridore, können überstehende Abschnitte durch geeignete Kriterien als Räume erkannt werden.

Für die Evaluation wurden von vier Testpersonen über 200 Spuren in alltäglichen Szenarien aufgenommen. Wählt man aus diesen Spuren 90 aus, so werden im Durchschnitt über 90% aller Korridore des Stockwerks erkannt. In 65% der so generierten Innenraummodelle war die durchschnittliche Verschiebung der Korridore kleiner als 1,5 m.

Abstract

It is possible to generate indoor models by using traces recorded by inertial measurement units. The generated indoor model distinguishes between rooms and corridors.

Traces will be collected by foot-mounted inertial measurement units via ZUPT. The data will be collected in a crowd based approach via Smartphones and sensor units carried by users. Users will walk inside the building in all-day situations, collecting data opportunistically.

The collected traces will be segmented into parts where the user walked straight. Using an equivalence relation, segments collected from the same corridor can be combined. Reconstructing the geometry of corridors will use quantiles and the empirical distribution function. Using a method to correct traces via the constructed corridor geometry, rooms can be found by protruding parts of traces.

To evaluate the system, four volunteers collected over 200 traces in everyday scenarios. Choosing 90 out of them, in average 90% of all corridors will be found. In 65% of this constructed indoor models, the average shift of corridors was less than 1.5 m.

Inhaltsverzeichnis

1	Einleitung	11
1.1	Motivation	11
1.1.1	Karten für Location Based Services	11
1.1.2	Karten zur Positionierung in Gebäuden	11
1.2	Zielsetzung	12
1.3	Ansatz	12
1.4	Aufbau der Ausarbeitung	13
2	Grundlagen	15
2.1	Erfassung von Bewegungsspuren mittels Inertialsensoren	15
2.1.1	Positionierung mittels Koppelnavigation	15
2.1.2	Inertialsensoren	16
2.1.3	Zero-velocity updates (ZUPT)	17
2.2	Verarbeitung von Bewegungsspuren	18
2.2.1	Segmentierung der Bewegungsspuren	18
2.2.2	Korridorsegmente	18
2.2.3	Ausrichtung parallel oder orthogonal zu Außenwänden	19
2.3	Grundlagen aus der Statistik	20
2.3.1	Quantile und Interquartilsabstand	20
2.3.2	Empirische Verteilungsfunktion	22
2.4	Public Sensing	22
2.4.1	Einbeziehen des Benutzers	23
2.4.2	Public, Social und Personal Sensing	23
3	Verwandte Arbeiten	25
3.1	CrowdInside	25
3.1.1	Erkennen des Gebäudes und Schätzen der Position der Eingangstüre	25
3.1.2	Ankerpunkte	26
3.1.3	Erfassen von Korridoren und Räumen	27
3.2	FootSLAM	28
3.2.1	Kartenerstellung mit Hilfe eines dynamischen Bayes-Netzes	28
3.2.2	Erweiterungen	29
3.3	Innenraummodelle durch abfotografieren von Fluchtplänen	30
4	Systemmodell und Anforderungen	31
4.1	Systemmodell	31
4.1.1	Bewegungsspuren	31

4.1.2	Mobile Einheiten	32
4.1.3	Mobilität der mobilen Einheit	32
4.1.4	Server	33
4.1.5	Außenraummodelle	33
4.1.6	Räume und Korridore	33
4.2	Anforderungen	34
4.2.1	Opportunistisches Aufzeichnen von Bewegungsspuren	34
4.2.2	Erfassen des Korridorskeletts	35
4.2.3	Erfassen der Räume	35
4.2.4	Umgang mit ungenauen Sensordaten	36
5	Entwurf	37
5.1	Übersicht über den Kartenerstellungsprozess	37
5.2	Erfassung der Daten	39
5.3	Segmentierung und Außenwandkorrektur	39
5.3.1	Segmentierung der Bewegungsspuren	40
5.3.2	Außenwandkorrektur	42
5.4	Erstellen eines Basisplans	42
5.4.1	Die Korridorrelation	43
5.4.2	Finden geeigneter Spuren für den Basisplan	45
5.4.3	Zusammenstellen des Basisplans	47
5.5	Iteratives Hinzufügen von Spuren auf das Korridorskelett	47
5.5.1	Bewertungsfunktionen	49
5.5.2	Bewertung anhand der Länge auf bestehenden Korridoren	49
5.5.3	Bewertung anhand des ersten und letzten Korridorsegments	49
5.5.4	Kombination von Bewertungsfunktionen	50
5.6	Reparaturschritt	50
5.6.1	Fehlertypen und Reparatur	51
5.6.2	Detektierung des Fehlertyps einer Spur	52
5.7	Geometrische Repräsentation von Korridoren	53
5.7.1	Bestimmung der Breite eines Korridors	53
5.7.2	Bestimmung der Korridormitte	54
5.7.3	Bestimmung der Länge eines Korridors	55
5.7.4	Verlängern der Korridore	58
5.8	Korrektur der Spuren durch die Korridormitte	59
5.9	Erkennen von Räumen	60
5.9.1	Erkennen der Räume aus der Geometrie des Flurskeletts	60
5.9.2	Bestimmung von Geometrie und Türposition	61
5.10	Zusammenfügen von Räumen	62
6	Implementierung	65
6.1	Architektur	65
6.2	Mobile Sensoreinheiten	66
6.2.1	Die Sensoreinheit	66
6.2.2	Die ComNSenseClient App	68

6.3	Datenbankschema für Bewegungsspuren	72
6.3.1	Beschreibung des Datenbankschemas	72
6.4	Bereitstellen und Abspeichern von Bewegungsspuren	74
6.5	Implementierung des Dienstes zur Kartenerstellung	74
6.5.1	Architektur	75
6.5.2	Caching von Spuren und konstruierten Korridorskeletten	79
6.5.3	Mapper	79
7	Evaluation	81
7.1	Methodologie	81
7.1.1	Szenarien der Datenerfassung	81
7.1.2	Die erfassten Bewegungsspuren	84
7.1.3	Vorgehen bei der Evaluation	85
7.2	Voruntersuchungen	85
7.2.1	Vergleich der Segmentierungsmethoden	85
7.2.2	Vergleich der Verfahren zur Bestimmung der Korridormitte	86
7.2.3	Proof of Concept	87
7.3	Anzahl der gefundenen Korridore	89
7.4	Geometrischer Fehler von Korridoren	90
7.5	Anzahl der gefundenen Räume	92
8	Zusammenfassung und Ausblick	95
	Literaturverzeichnis	99

Abbildungsverzeichnis

2.1	Segmentierung anhand des Winkels	19
2.2	Beispiel einer empirischen Verteilungsfunktion.	22
3.1	Skizze der Situation am Nordeingang des Informatikgebäudes	26
3.2	Dynamisches Bayes-Netz von FootSLAM	29
4.1	Außenraummodell des Informatikgebäudes	33
4.2	Beispiel eines Korridors	34
5.1	Überblick über die einzelnen Schritte der Innenraummodellierung.	38
5.2	Segmentierung durch Polynominterpolation	40
5.3	Segmentierung anhand der Höhe des Dreiecks	41
5.4	Beispiel einer Segmentierung	41
5.5	Beispiel einer an Außenwänden ausgerichteten Spur	42
5.6	Beispiel eines x -umschließenden Rechtecks	44
5.7	Beispiel zur Korridorrelation	45
5.8	Projektion der absoluten Punkte zur Bestimmung der Korridorbreite	53
5.9	Beispiel zur Bestimmung der Korridormitte	54
5.10	Projektion auf Intervalle	56
5.11	Verteilung und Bestätigung von Korridorsegmenten	57
5.12	Beispiele zur topologischen Erweiterung	59
6.1	Übersicht über die verschiedenen Komponenten und deren Kommunikationsschnittstellen, welche für die Implementierung verwendet wurden.	66
6.2	Die prototypische Sensoreinheit	67
6.3	Smartphone mit der App	68
6.4	Datenbankschema der SQLite Datenbank in der Android App	69
6.5	Datenbankmodell für die Abspeicherung der Bewegungsspuren	72
6.6	UML Diagramm des Kartenerstellungsdienstes	77
7.1	Übersicht über das zweite Stockwerk des Informatikgebäudes der Universität Stuttgart	82
7.2	Unkorrigierte, per ZUPT erfasste Bewegungsspuren	84
7.3	Exemplarischer Vergleich der Segmentierungsmethoden	86
7.4	Exemplarischer Vergleich der Verfahren zum Bestimmen der Korridormitte	87
7.5	Aus allen Spuren abgeleitetes Flurskelett	88
7.6	Aus allen Spuren abgeleitetes Innenraummodell	88

7.7	Anzahl der gefundenen Korridore mit unterschiedlicher Anzahl von Bewegungsspuren	90
7.8	Durchschnittlicher Fehler der Bestimmung der Längsseite von Korridoren über der Anzahl der zufällig gewählten Bewegungsspuren	91
7.9	Anteil der gefundenen Räume über der Anzahl der zufällig gewählten Spuren aus dem Gebäude und dem Süd-West Quadranten	92

Tabellenverzeichnis

6.1	Die von der Sensoreinheit via Bluetooth bereitgestellten Befehle.	68
6.2	Vom Spurserver per XML-RPC bereitgestellte Funktionen	74
6.3	Einige Optionen von Mapper	80
7.1	Werte der Parameter, welche bei der Evaluation verwendet wurden	89

Verzeichnis der Algorithmen

2.1	Korrektur der Bewegungsspuren mit Hilfe eines Außenraummodells	20
5.1	Iteratives Hinzufügen von Bewegungsspuren auf den Basisplan	51
5.2	Finden des Startpunkts eines Korridors K mittels Verteilungsfunktion	58
5.3	Zusammenfügen von Räumen aus verschiedenen Bewegungsspuren	63

1 Einleitung

1.1 Motivation

Karten sind ein wichtiges Hilfsmittel, um einen Ort abstrakt beschreiben zu können. Sie dienen Menschen zur Orientierung an unbekanntem Orten und ermöglichen Maschinen möglichst kurze oder schnelle Verbindungen zu finden. Das Erstellen von Karten für den Außenbereich hat daher eine lange Geschichte [Wik13b]. Für den Innenbereich stehen allerdings keine einheitlichen Kartendaten zur öffentlichen Verfügung. Die Gründe dafür sind vielseitig. Angefangen von eventuell rechtlichen Problemen, zu Bedenken über die Privatsphäre, bis hin zu ökonomischen Gründen.

Im Folgenden werden einige Anwendungsbeispiele für Karten und ein System, zur Erstellung von Innenraummodellen aus Bewegungsspuren, aufgeführt.

1.1.1 Karten für Location Based Services

Ein Verwendungszweck für Karten von Innenräumen sind Location Based Services (LBS). Die Karten können hier zum einen zur Visualisierung dienen. So könnten beispielsweise in einem Kaufhaus interessante Angebote und deren Position für Kunden auf einer Karte angezeigt werden. Zum anderen kann durch Karten eine Navigation in großen unübersichtlichen Gebäuden, wie beispielsweise Krankenhäuser oder Universitäten, ermöglicht werden.

Eine weitere Anwendung ist das Kartographieren von Gebäuden in Katastrophenszenarien, wie etwa einer großen brennenden Anlage. Karten helfen hier der Einsatzleitung beim Koordinieren der Einsatzkräfte. So können zusätzliche Informationen in die Karten eingezeichnet und so der Überblick über das Geschehen behalten werden.

1.1.2 Karten zur Positionierung in Gebäuden

Gerade für Feuerwehrleute im vorherigen Beispiel ist eine Positionierung beim Verlust der Orientierung im Qualm des Feuers sehr wichtig [Har13]. Aber auch für die zivile Nutzung ist eine Positionierung in Innenräumen immer noch nicht flächendeckend verfügbar. Zwar gibt es Möglichkeiten zur Bestimmung der Position mittels Triangulation durch Basisnoten [LDBL07], jedoch ist deren Betrieb und Installation mit Kosten verbunden, welche vom Betreiber getragen und nicht ohne weiteres auf den Benutzer umgelegt werden können.

Wären Karten von Innenräumen flächendeckend verfügbar, so ließe sich die Positionierung mittels Inertialsensoren lösen. Wie später genauer beschrieben wird, ist es mit diesen Sensoren möglich, eine Bewegungsspur des Benutzers zu erfassen. Diese Spur driftet über die Zeit ab. Durch Karten kann dieser Drift verhindert und somit eine Positionierung zu jedem Zeitpunkt auf der Spur ermöglicht werden [WHo8, BK⁺08].

1.2 Zielsetzung

Das Ziel dieser Arbeit ist es, ein Verfahren zu entwickeln, um mit Hilfe von Bewegungsspuren von Fußgängern ein Innenraummodell eines Gebäudes abzuleiten. Die Spuren der Benutzer werden dabei von Smartphones und einer auf dem Fuß platzierten Sensoreinheit ermittelt.

Wesentliches Problem ist die Drift in den Spuren, welche durch ungenaue Sensordaten und die Verwendung von Koppelnavigation erzeugt wird. Es muss also eine Methode entwickelt werden, um die Bewegungsspuren zu korrigieren und damit für die weitere Verarbeitung brauchbar zu machen.

Ein weiteres Ziel ist es, eine höhere Semantik aus Spuren zu extrahieren. Es sollen Korridore und Räume erkannt werden. Dazu soll eine Methode erarbeitet werden, um zu erkennen, wann sich verschiedene Spuren auf dem selben Korridor oder im selben Raum befunden haben.

Das letztendlich erzeugte Innenraummodell soll möglichst genau mit der Realität übereinstimmen. Zum einen soll es die Topologie aus dem realen Gebäude erhalten. Korridore und Räume sollen also in der korrekten Nachbarschaftsbeziehung zueinander stehen. Zum anderen soll auch die Geometrie möglichst genau erfasst werden und somit ein Weg gefunden werden, die Ausdehnungen von Räumen und Korridoren möglichst exakt zu bestimmen.

1.3 Ansatz

In der vorliegenden Arbeit wird ein System vorgestellt, welches aufgezeichnete Bewegungsspuren von Fußgängern verwendet, um ein Innenraummodell zu erstellen. Bewegungsspuren können dabei als eine Aufzeichnung der Bewegung im Verlauf der Zeit gesehen werden.

Für die Erfassung der Bewegungsspuren wird ein Inertialsensor verwendet, welcher auf einem Fuß befestigt wird. Für die Erfassung solcher Spuren kam ein Prototyp zum Einsatz. Zwar ist die Verwendung des Prototypen im Alltag eher unpraktisch, doch existieren Projekte, um solche Sensoreinheiten besser an die Bedürfnisse der Benutzer anzupassen [Fox05, NSHH12].

Die Positionierung des Sensors auf dem Fuß ermöglicht die Ausnutzung von Eigenschaften des menschlichen Gangs, so dass die Positionsänderung des Fußes bei einem Schritt genauer erfasst werden kann. Durch bessere Sensoren und weitere Erkenntnisse der Forschung im Bereich der Inertialnavigation ist davon auszugehen, dass Bewegungsspuren in ähnlicher

Qualität in einiger Zeit auch allein aus den Daten der internen Sensoren eines Smartphones verwendet werden können.

1.4 Aufbau der Ausarbeitung

Diese Ausarbeitung gliedert sich in acht Kapitel. In Kapitel 2 werden Grundlagen erklärt, welche für das Verständnis der restlichen Arbeit notwendig sind. Kapitel 3 behandelt verwandte Arbeiten, welche ähnliche Probleme lösen. In Kapitel 4 folgt schließlich das Systemmodell und die Anforderungen an den Entwurf. Der Entwurf ist in Kapitel 5 beschrieben. Es folgen einige Details über die Implementierung des Entwurfs in Kapitel 6. In Kapitel 7 wird der Ansatz durch die Implementierung getestet und eine prototypische Evaluation durchgeführt. Schließlich endet die Ausarbeitung in Kapitel 8 mit einer Zusammenfassung und einem Ausblick über mögliche anschließende Arbeiten.

2 Grundlagen

In diesem Kapitel werden Grundlagen zusammengefasst, welche für das Verständnis bisheriger und der vorliegenden Arbeit nötig sind.

Zunächst werden einige Methoden zur Erfassung und Korrektur von Bewegungsspuren mit Inertialsensoren vorgestellt. Hierzu wird neben einer kurzen Einführung über Inertialsensoren eine Möglichkeit vorgestellt aus den kontinuierlichen Daten, welche der Sensor liefert, Schrittvektoren zu extrahieren.

Sind Schrittvektoren gegeben, so können diese in Segmente, also gerade Abschnitte, unterteilt werden. Durch Verwendung eines Außenraummodells wird eine Möglichkeit beschrieben, wie diese Segmente korrigiert werden können.

Neben einigen statistischen Grundlagen wird im weiteren Verlauf des Kapitels auch die Taxonomie von Public Sensing Systemen eingeführt, um das System später besser in diese Gruppe von Systemen einordnen zu können.

2.1 Erfassung von Bewegungsspuren mittels Inertialsensoren

Bewegungsspuren beschreiben die zurückgelegte Strecke eines Fußgängers. Darstellen lässt sich diese durch eine initiale Position p_0 und eine Menge von relativen Positionsänderungen $S = \{s_1, \dots, s_n\}$. Die Änderung der Position wird hierbei als 2D Vektor beschrieben. Wird pro Schritt ein solcher Änderungsvektor aufgezeichnet, so spricht man auch von *Schrittvektoren*.

Im Folgenden wird erklärt, wie man mit Hilfe eines Inertialsensors Bewegungsspuren aufzeichnen kann.

2.1.1 Positionierung mittels Koppelnavigation

Um Bewegungsspuren zu erfassen, benötigt man eine Abschätzung der Position zu bestimmten Zeitpunkten. Zur Positionierung im Außenbereich hat sich mit GPS eine Technik entwickelt, welche absolute Punkte zur Bestimmung liefert. Systeme zur absoluten Positionsbestimmung arbeiten meist mit Triangulation [LDBL07]. Hierfür sind stationäre Basisknoten notwendig, deren Position bekannt sein muss. Der Betrieb und das Aufsetzen dieser Basisknoten ist mit entsprechenden Kosten verbunden. Diese Kosten rechtfertigen sich meist nicht in Gebäuden, in denen eine Positionierung nur selten benötigt wird.

Ein anderer Ansatz zur Positionierung wird in der Seefahrt verwendet. Man verwendet die Geschwindigkeit, die Richtung und die Zeit, um eine relative Positionsänderung zu berechnen. Diese verrechnet man schließlich mit der vorherig bekannten Position. Dieses Verfahren nennt sich Koppelnavigation (englisch: dead reckoning) und lässt sich als Koppelnavigation für Fußgänger (englisch: pedestrian dead reckoning, PDR) zur Positionsbestimmung in Innenräumen übertragen.

Bei PDR wird durch jeden Schritt ein *Schrittvektor* erfasst. Dieser gibt die relative Positionsänderung eines Schrittes wieder. Mit Hilfe einer Startposition und aller vorherigen Schrittvektoren ist es somit möglich die Position des Benutzers zu bestimmen.

Mit reiner Koppelnavigation wie PDR, entfallen teure stationäre Knotenpunkte. Dafür erhält man über die Zeit eine größer werdende Ungenauigkeit in der Positionierung. Bei jedem neu bestimmten Punkt wird der Fehler der vorherigen Position fortgesetzt. Hinzu kommt ein weiterer Fehler, welcher beim Messen der relativen Positionsänderung entsteht. Es ist also besonders wichtig mittels Sensoren die relative Änderung in einem Zeitintervall möglichst genau zu erfassen. Von solchen Sensoren handelt der folgende Abschnitt.

2.1.2 Inertialsensoren

Es gibt sechs physikalische Freiheitsgrade, durch die man ein Objekt im dreidimensionalen Raum bewegen kann. Man unterscheidet hierbei zwischen den drei Freiheitsgraden der Translation, welche die Bewegung des Schwerpunktes in den drei Raumrichtungen beschreiben und den drei Freiheitsgraden der Rotation, welche Drehbewegungen um den Schwerpunkt festlegen. Zur Messung der Beschleunigung auf diesen sechs Freiheitsgraden werden inertielle Messeinheiten (englisch: inertial measurement unit, IMU) verwendet. Technisch bestehen diese Einheiten aus drei Gyroskopen zur Messung der Rotation und drei Beschleunigungssensoren zur Messung der Translation.

Nahezu alle modernen Smartphones haben eine IMU integriert, um zu erkennen, wann der Benutzer das Mobilgerät zur Seite kippt, um daraufhin den Bildschirm mitzudrehen. Für diese relativ einfache Aufgabe müssen die verbauten Sensoren nicht sehr genau sein. PDR mit den internen Sensoren zu betreiben ist daher schon aufgrund der Sensorungenauigkeit schwierig. Da der Benutzer das Gerät an beliebigen Stellen am Körper tragen kann, ist es zudem schwierig zu erkennen, in welcher Phase eines Schrittes sich der Benutzer befindet. Techniken wie das später beschriebene ZUPT zur Verbesserung der Genauigkeit von Schrittvektoren können daher nicht angewandt werden.

Der hier vorgestellte Entwurf basiert auf der Erfassung der Bewegungsspuren mittels auf dem Fuß befestigter IMU. Durch diese fixe Position am Körper ist es einfacher Eigenschaften des menschlichen Ganges zur Verbesserung der Genauigkeit der Positionsänderung auszunutzen.

Mit NavShoe [Fox05] und OpenShoe [NSHH12] existieren bereits Projekte, welche fertige Sensoreinheiten mit Inertialsensoren auf dem Fuß für die Forschung anbieten wollen. Setzt

sich diese Positionierungsmethode in Innenräumen durch, so ist zu erwarten, dass es schon bald Anbieter von fertigen Lösungen für die zivile Nutzung geben wird.

Neu entwickelte IMUs werden genauer und mit großer Verbreitung von Smartphones auch billiger. Eventuell könnte schon bald PDR mittels den internen Sensoren eines Smartphones Bewegungsspuren von ähnlicher Qualität liefern, wie man sie heute durch Sensoren auf dem Fuß erfassen kann.

2.1.3 Zero-velocity updates (ZUPT)

IMUs messen die Beschleunigung, welche der Sensor erfährt. Zur Bestimmung der neuen Position benötigt man jedoch die Positionsänderung. Ein naiver Ansatz um aus der Beschleunigung an die relative Änderung zu bekommen, ist zweifach zu integrieren. Nach der ersten Integration der Beschleunigung bekommt man die relative Geschwindigkeit. Integriert man über die Geschwindigkeit, so bekommt man die Änderung des Ortes.

Bleibt der Benutzer stehen, so beträgt seine Geschwindigkeit in alle Richtungen 0. Bereits durch kleine Messfehler besitzt das Integral für die Geschwindigkeit nach obiger Implementierung einen anderen Wert als 0. In der erfassten Spur würde sich der Benutzer also weiterbewegen, obwohl er eigentlich steht. Die Daten aus IMUs unterliegen, wie Daten aus anderen Sensoren, einer gewissen Ungenauigkeit. Diese verstärkt sich über die Zeit hinweg. Auch liefert eine Implementierung nach obiger Beschreibung kontinuierliche Daten und keine Schrittvektoren.

Zero-velocity updates (ZUPT) lösen beide Probleme [Fox05, NSHH12, FTSH13]. Beim menschlichen Gang beschreibt jeder Fuß zwei Phasen. Man unterscheidet zwischen einer Stand- und einer Geh-Phase. Beide Phasen dauern bei typischer Gangart ungefähr 0.5 Sekunden an. Bei der Stand-Phase trägt ein Fuß das komplette Gewicht des Benutzers, wodurch der andere Fuß die Geh-Phase absolvieren kann.

Mit ZUPT wird nun versucht die Stand-Phase zu erkennen. Die Geschwindigkeit wird dann auf 0 zurückgesetzt. Durch die Kenntnis der Differenz unter dem Integral der Beschleunigung kann man außerdem mit Hilfe eines Kalman-Filters die Positionsänderung genauer erfassen [Fox05]. Hierzu gehen zum einen die naiv integrierten, als auch durch ZUPT gewonnenen Werte in den Filter ein. Dieser verknüpft die berechneten Positionen und verwendet somit den Fehler in der Geschwindigkeit, um die Position genauer zu bestimmen [Wik13a].

Zum Erkennen der Stand-Phase gibt es unterschiedliche Ansätze. Es gibt die Möglichkeit dies durch Taster in den Fußsohlen zu detektieren. Im hier beschriebenen Entwurf wird diese Phase jedoch nur mit der IMU selbst erkannt. Steht man auf einem Bein, so ist dessen Beschleunigung in alle Richtungen 0. Hier spielt also die Platzierung des Sensors auf dem Fuß eine besondere Rolle. Trotzdem kann es bei der Detektierung der Stand-Phase mit der IMU zu fehlerhaften Erkennungen kommen, etwa wenn man den Fuß mit konstanter Geschwindigkeit bewegt oder durch Rauschen im Sensor eine Stand-Phase nicht erkannt wird. In gemessenen Daten tauchen daher in seltenen Fällen Doppelschritte, also Schritte, in

denen ein Fuß eigentlich zweifach den Boden berührt hatte, oder eine Unterteilung eines Schrittes in mehrere kleine auf.

Generell weisen die so berechneten Schrittvektoren Fehler auf, welche die Bewegungsspuren über die Zeit zunehmend ungenauer werden lassen. Im Folgenden werden daher die Grundlagen für eine Korrektur der Spur vorgestellt.

2.2 Verarbeitung von Bewegungsspuren

Auch wenn sich durch ZUPT eine Bewegungspur genauer erfassen lässt, so unterliegt diese trotzdem einer Drift. Mit der Zeit wird die Position ungenauer.

In diesem Abschnitt wird ein Verfahren vorgestellt, um durch ein Außenraummodell diese Drift einzugrenzen. Hierbei hilft die Beobachtung, dass sich Korridore meist parallel oder orthogonal zu Außenwänden erstrecken [JSZ⁺12]. Trifft dies auf das Gebäude zu, so sind die möglichen Ausrichtungen durch ein Außenraummodell, in welchem die Außenwände eingezeichnet sind, vorgegeben.

Um herauszufinden, in welchen Abschnitten einer Bewegungspur der Benutzer auf einem Korridor gegangen ist, wird eine solche Spur in Segmente unterteilt. Dies sind Abschnitte, in denen der Benutzer keine Kurve gegangen ist. Durch die Länge der Segmente kann dann herausgefunden werden, ob sich der Benutzer auf einem Korridor aufgehalten hatte.

2.2.1 Segmentierung der Bewegungsspuren

Um eine gröbere Übersicht über die Bewegungspur zu erhalten, wird diese in gerade Abschnitte, sogenannte Segmente unterteilt. Ein Segment ist dabei ein Teil einer Spur, in welcher der Benutzer geradeaus ging. Segmente werden also durch Kurven unterbrochen.

Zum Durchführen einer Segmentierung werden meist zwei aufeinanderfolgende Schritte herangezogen, um zu entscheiden, ob sich diese im selben Segment befinden. Gängig ist es, den Winkel $\alpha \in (-\pi, \pi]$, welcher die Richtungsänderung von zwei aufeinanderfolgenden Schrittvektoren angibt, zu betrachten (siehe Abbildung 2.1) [AY12]. Durch einen Parameter α_{\max} , welcher die maximale Abweichung festlegt, wird unterschieden, ob zwei aufeinanderfolgende Schritte zum selben Segment gehören (falls $|\alpha| \leq \alpha_{\max}$) oder ob sich diese in unterschiedlichen Segmenten befinden (falls $|\alpha| > \alpha_{\max}$). Dementsprechend werden die Schritte zu Segmenten zusammengefasst.

2.2.2 Korridorsegmente

Um zu unterscheiden, ob ein Segment auf einem Flur gegangen oder in einem Raum erfasst wurde, kann man die Länge von diesem heranziehen [AY12]. Kleinere Segmente können auch in einem Raum gegangen worden sein. Bei längeren Segmenten kann man hingegen

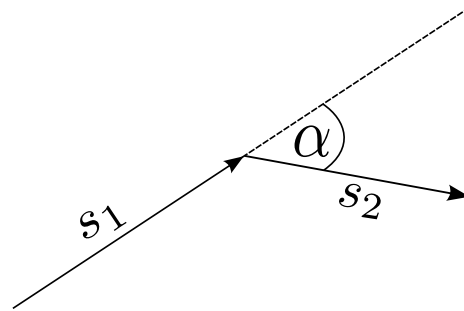


Abbildung 2.1: Segmentierung anhand des Winkels. Die Schritte s_1 und s_2 gehören zum selben Segment, wenn der Betrag des Winkels $\alpha \in (-\pi, \pi]$, welcher die Richtungsänderung angibt, kleiner als ein Maximalwert α_{\max} ist.

ausschließen, dass diese in einem Raum gegangen wurden, da Räume meist viel kleiner als Korridore sind.

Als Länge eines Segments verwendet man dazu die euklidische Länge der Summe aller zugehörigen Schrittvektoren. Zur Unterscheidung setzt man eine Minimallänge L_{\min} für Segmente, welche vermutlich auf Korridoren gegangen wurden, fest. Alle Segmente, welche länger als L_{\min} sind, nennt man *Korridorsegmente*.

Man könnte erwarten, dass man L_{\min} in Abhängigkeit der maximalen Raumdiagonalen wählen muss, um wirklich zwischen Korridoren und Räumen unterscheiden zu können. In der Praxis versperren allerdings meist Gegenstände und Möbel die Raumdiagonale. Auch geht man in Räumen meist nicht so zielstrebig geradeaus wie auf Korridoren. Man kann L_{\min} also unabhängig vom Gebäude auf einen Wert festlegen.

2.2.3 Ausrichtung parallel oder orthogonal zu Außenwänden

Ein wichtiges Hilfsmittel zur Korrektur der Bewegungsspuren ergibt sich durch Ausnutzung der sehr verbreiteten Eigenschaft, dass sich Flure meist parallel oder orthogonal zu Außenwänden erstrecken [JSZ⁺12]. Mittels Kartendiensten, wie OpenStreetMap, kann ein Außenraummodell des Gebäudes und somit auch die Ausrichtung der Außenwände abgefragt werden. Durch diese Information kann man schließlich die Korridorsegmente entsprechend einer Außenwand korrigieren.

Ist ein Korridorsegment parallel oder orthogonal zu einer Außenwand ausgerichtet, so heißt dieses *nach den Außenwänden orientiert*. Die Menge der möglichen Ausrichtungen durch ein Außenraummodell heißt entsprechend *Menge der Orientierungen*.

Um ein Korridorsegment zu orientieren, verwendet man als resultierenden Vektor des Segments die Differenz zwischen dessen Start- und Endpunkt. Man bestimmt den kleinsten

Algorithmus 2.1 Korrektur der Bewegungsspuren mit Hilfe eines Außenraummodells

```
procedure SPURKORREKTUR_AUSSENWAND(spur, orientierungen)
  Korridorsegmente  $\leftarrow$  Liste der geordneten Korridorsegmente in spur
  for  $s \in$  Korridorsegmente do
    Bestimme absolut minimales  $\alpha$ , so dass  $s$  nach Rotation um  $\alpha$  nach  $o \in$  orientierungen
    orientiert ist.
    Drehe  $s$  und alle folgenden Schrittvektoren der spur um  $\alpha$ 
  end for
end procedure
```

absoluten Winkel, so dass der resultierende Vektor des Korridorsegments parallel oder orthogonal zu einer Außenwand steht. Schließlich dreht man alle Schrittvektoren entsprechend. Das Korridorsegment ist nun nach den Außenwänden orientiert.

Die Position von aufeinanderfolgenden Korridorsegmenten hängt voneinander ab. Daher ist es wichtig, dass die Orientierung anhand von Außenwänden geordnet nach der zeitlichen Reihenfolge der Korridorsegmente durchgeführt wird.

Des Weiteren sollte durch die Kausalität bei der Korrektur eines Korridorsegments nicht nur dieses, sondern auch alle darauffolgenden Schrittvektoren mitgedreht werden. Dadurch werden auch Schrittvektoren korrigiert, die in keinem Korridorsegment enthalten sind, wie etwa in Räumen oder von Kurven.

Algorithmus 2.1 fasst das Verfahren im Pseudocode zusammen. Als Eingabe wird neben einer Bewegungsspur eine Menge an möglichen Orientierungen gegeben. Die Elemente können zum Beispiel durch den Winkel zu einem Vektor, welcher Richtung Norden zeigt, repräsentiert werden. Diese Menge kann aus einem Außenraummodell gewonnen werden.

2.3 Grundlagen aus der Statistik

Für den Umgang mit den erfassten Daten, besonders zum Festlegen der Geometrie des Korridors, werden einige Grundlagen aus der Statistik benötigt. Gerade bei sehr ungenauen Sensordaten liefert die Statistik wichtige Methoden, um die Daten zu verknüpfen und daraus neue Schlüsse ziehen zu können. In diesem Abschnitt wird der Begriff eines p -Quantils, des Interquartilabstands und der empirischen Verteilungsfunktion einer Messreihe eingeführt.

2.3.1 Quantile und Interquartilsabstand

Quantile sind eine Art Verallgemeinerung des Medians. Der Median m einer Messreihe $X = \{x_1, \dots, x_n\}$ ist der Wert, für welchen eine Hälfte aller Beobachtungen kleiner oder

gleich und die andere Hälfte der Beobachtungen größer oder gleich m ist. Es gelten also folgende Formeln

$$|\{i : x_i \leq m\}| \geq \frac{n}{2}$$

$$|\{i : x_i \geq m\}| \geq \frac{n}{2}$$

Zur Verallgemeinerung wird nun ein Anteil von p vorgeschrieben, welcher kleiner als das Quantil sein sollen. Entsprechend soll ein Anteil von $1 - p$ größer sein als das Quantil. Der Median ist damit das 0.5-Quantil. Im Folgenden wird eine formale Definition für Quantile gegeben.

Formale Definition von Quantilen

Es sei $X = \{x_1, \dots, x_n\}$ eine Messreihe und p Element aus dem offenen Intervall $(0, 1)$. Eine Zahl q ist als p -Quantil definiert, wenn circa $p \cdot 100\%$ der Beobachtungen kleiner oder gleich q und circa $(1 - p) \cdot 100\%$ der Beobachtungen größer oder gleich q sind.

Formal muss für q als p -Quantil folgendes gelten:

$$\frac{1}{n} |\{i : x_i \leq q\}| \geq p \quad \text{und} \quad \frac{1}{n} |\{i : x_i \geq q\}| \geq 1 - p$$

Für ein p -Quantil einer Messreihe X schreibt man auch $q_p(X)$.

Der Interquartilsbereich

Neben dem Median werden vor allem die 0.25- und 0.75-Quantile häufig genutzt. Diese nennt man auch Quartile. Der Abstand zwischen diesen Quartilen nennt man den *Interquartilsabstand*. Man erhält einen solchen Interquartilsabstand auch, wenn man die Messreihe sortiert, 25% der kleinsten und 25% der größten Werte abschneidet und danach die maximale Distanz zwischen zwei Messergebnissen berechnet.

Die Zahlenwerte zwischen einem 0.25- und einem 0.75-Quantil nennt man auch den *Interquartilsbereich*. Formal lässt sich dieser auch durch ein Intervall darstellen: $[q_{0.25}(X), q_{0.75}(X)]$.

Da Quantile nicht immer eindeutig festgelegt sind, gibt es verschiedene Interquartilsabstände und Interquartilsbereiche. Zum Beispiel ist der Median von vier Werten nicht eindeutig definiert. Um den Interquartilsabstand später für die Anwendung festzulegen, wird für das 0.25-Quantil jeweils der größtmögliche Wert und für das 0.75-Quantil der kleinstmögliche Wert verwendet.

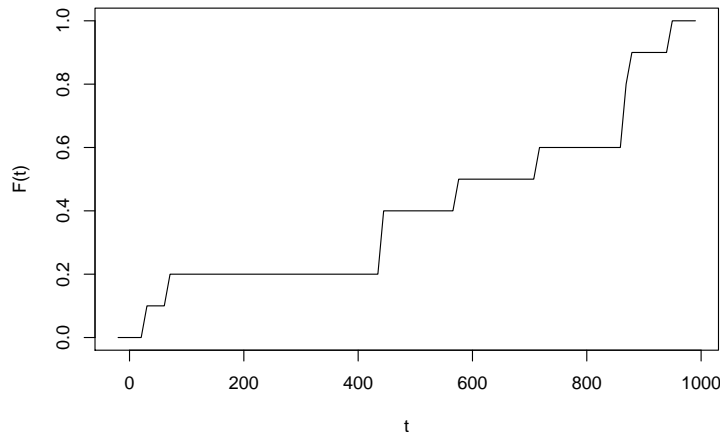


Abbildung 2.2: Beispiel einer empirischen Verteilungsfunktion.

2.3.2 Empirische Verteilungsfunktion

Die Verteilung von ungenauen Messdaten erlaubt es, weitere Schlüsse auf die tatsächlich gemessenen Objekte zuzulassen. Später soll mit Hilfe der Verteilung der Start- und Endpunkte von Korridorsegmenten die geometrische Länge der Korridore geschätzt werden. Um eine Art Verteilungsfunktion einer Messreihe zu erhalten, verwendet man den Anteil der Messpunkte, welcher kleiner als ein vorgegebener Wert ist.

Die *empirische Verteilungsfunktion* F_X einer Messreihe $X = \{x_1, \dots, x_n\}$ ist definiert als

$$F_X(t) = \frac{|\{x \in X : x \leq t\}|}{|X|}.$$

Diese ist monoton wachsend und nimmt Werte zwischen 0 und 1 an. Zusätzlich handelt es sich bei der empirischen Verteilungsfunktion um eine Stufenfunktion. Dadurch ist das Integrieren über diese mit Hilfe einer Summe exakt möglich.

Abbildung 2.2 zeigt ein Beispiel für eine empirische Verteilungsfunktion. Hierzu wurden zufällig 10 ganze Zahlen zwischen 0 und 1000 gezogen. Wie man auch aus der Definition der Funktion ablesen kann, nimmt damit die empirische Verteilungsfunktion für Werte kleiner als 0 den Wert 0 und für Werte größer als 1000 den Wert 1 an.

2.4 Public Sensing

Durch mobile Sensorknoten können Daten über ein größeres Gebiet erfasst werden als durch statische Sensoren [LBD⁺05]. Die Untersuchung von mobilen Sensornetzen und das Aufkommen von Smartphones haben in den letzten Jahren den Begriff des *Public Sensing* geprägt. Durch die weite Verbreitung von Smartphones, deren eingebauten Sensoren und

fast überall vorhandene Anbindung an das Internet, sind viele Szenarien zur Datenerfassung denkbar. Dabei spielt die Mobilität der Geräte eine große Rolle.

Am naheliegendsten ist das Erheben von Daten, welche direkt durch das Smartphone erfasst werden können. Dies beinhaltet zum Beispiel die Umweltbelastung durch Lärm [MSN⁺09]. Durch das Mikrofon eines Smartphones wird erfasst, wie laut die Umgebung ist. Mit Hilfe des eingebauten GPS Sensors wird zudem die Position der Messung festgehalten. Auf diese Weise erhält der Benutzer die Lärmbelastung, welcher er den Tag über ausgesetzt war. Zusätzlich kann er erkennen, wie stark die Belastung an verschiedenen Orten ist. Laden viele solche Benutzer die Daten auf einen Server, so können diese kombiniert werden. Auf diese Weise kann die Lärmbelastung einer ganzen Stadt erfasst werden.

Mit Hilfe von mobilen Sensoren können auch weitere Daten, wie etwa der Kohlenstoffmonoxid Gehalt der Luft [SM08] oder die Temperatur [PDR11, PSA⁺13, PSDR13] erfasst werden.

2.4.1 Einbeziehen des Benutzers

Es gibt zwei unterschiedliche Ansätze, wie man einen Benutzer in ein Public Sensing System einbezieht [LEM⁺08].

In einem Ansatz werden Daten im Hintergrund aufgezeichnet und bei Bedarf an einen Server in der Cloud übertragen. Der Benutzer kann in diesem Fall das Smartphone ganz normal verwenden. Dies nennt man den *opportunistischen Ansatz*.

Der zweite Ansatz bezieht den Benutzer aktiv ein, indem er die Messung der Daten auf dem Mobilgerät auslösen muss. Vorteile der zweiten Methode, welche auch *teilnehmender Ansatz* genannt wird, sind neben der Wahrung der Privatsphäre des Benutzers auch eventuell genauere Daten, da diese dem Benutzer angezeigt werden können und er diese prüfen und gegebenenfalls eine weitere Messung durchführen kann.

2.4.2 Public, Social und Personal Sensing

Neben verschiedenen Paradigmen der Benutzerinteraktion unterscheidet man auch zwischen Personal, Public und Social Sensing [CEL⁺08]. Personal Sensing bezieht sich hierbei nur auf Daten einer Person. Dies können zum Beispiel Daten über die körperliche Fitness beim Sport sein. Unter Social Sensing versteht man das Erfassen und den Gebrauch der Daten von Interessengruppen. Bei Public Sensing werden hingegen Daten für die Öffentlichkeit gesammelt. Es wird angenommen, dass solche Daten für alle nützlich sind.

3 Verwandte Arbeiten

In diesem Kapitel werden andere Arbeiten, welche sich mit dem Problem der Innenraummodellierung beschäftigen, vorgestellt. Zunächst wird mit CrowdInside ein System, welches dem später entworfenen sehr ähnelt, vorgestellt. Danach werden zwei weitere, etwas andere Vorgehensweisen präsentiert. Zum einen sind dies FootSLAM, welches aus der Robotik kommt und mit Hilfe von Bayes-Netzen die richtige Bewegungsspur aus den Sensordaten zu rekonstruieren versucht. Zum anderen wird ein Verfahren vorgestellt, welches auf abfotografieren von Fluchtplänen basiert.

3.1 CrowdInside

Zur Erstellung eines Innenraummodells verwendet CrowdInside Bewegungsspuren aus der im Smartphone verbauten IMU [AY12]. Es wird angenommen, dass einige Spuren von verschiedenen Fußgängern in einem Gebäude erfasst werden. Alle Daten werden an einen Server gesendet. Dieser erstellt von Zeit zu Zeit ein Modell des Gebäudes.

Im Gegensatz zu dem in dieser Arbeit vorgestellten System, werden für die Korrektur und zur Verknüpfung der Spuren einzelne Punkte, sogenannte Ankerpunkte verwendet. Dies sind markante Positionen innerhalb der Bewegungsspuren. Durch statistische Methoden kann durch diese eine genauere Schätzung der wirklichen Position der Gegebenheit erfolgen. Für die Unterscheidung der Ankerpunkte werden neben den Daten aus dem Inertialsensor auch Daten des GPS Sensors, sowie die Signalstärke von naheliegenden W-LAN Access Points herangezogen.

Die korrigierten Spuren werden segmentiert und in Raumsegmente und Korridorsegmente unterteilt. Für diese Unterscheidung wird neben der Länge die Dichte der umliegenden Segmente herangezogen. Die Gebäudehülle, das Korridorskelett und die Geometrie der Räume werden durch eine spezielle geometrische Repräsentation konstruiert. Hierbei wird im Gegensatz zum hier vorgestellten System angenommen, dass alle Spuren zu diesem Zeitpunkt korrekt sind.

3.1.1 Erkennen des Gebäudes und Schätzen der Position der Eingangstüre

Um stromsparend zu erkennen, wann ein Benutzer ein Gebäude betreten hat, verwendet CrowdInside GPS mit einer niedrigen Abfragefrequenz der Position. Fällt das GPS Signal aus, so wird angenommen, dass der Benutzer sich nun im Gebäude befindet. Um festzustellen,

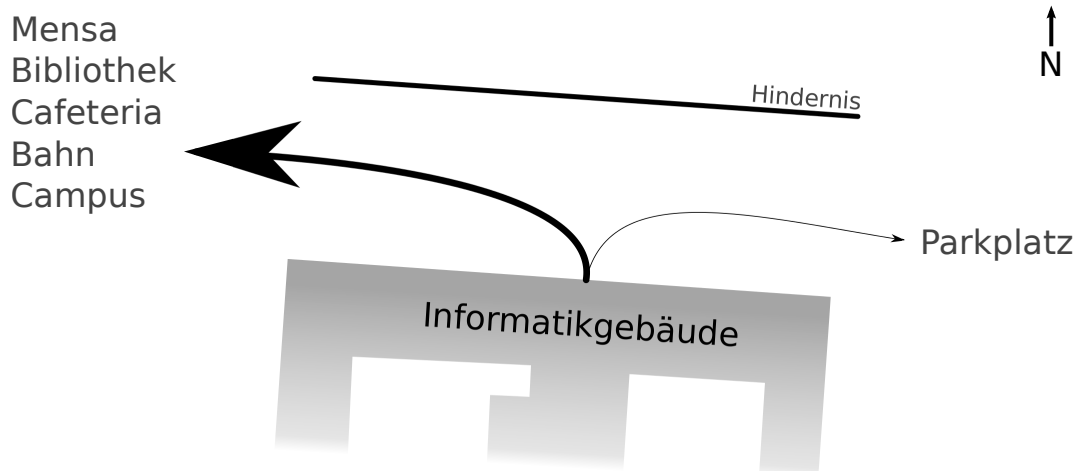


Abbildung 3.1: Skizze der Situation vor dem Nordeingang des Informatikgebäudes der Universität Stuttgart. Der direkte Zugang von Norden wird durch ein Hindernis versperrt. Die meisten Besucher des Gebäudes kommen und gehen von Westen auf das Gebäude zu.

um welches Gebäude es sich handelt wird ein Dienst wie OpenStreetMap verwendet, mit dessen Hilfe das nächstgelegene Gebäude bestimmt wird.

Zum Bestimmen der Position der Eingangstüre werden die Daten des Inertialsensors und die letzte GPS Position verwendet. Die Position der Türe wird hierbei auf halber Strecke von der Position des letzten GPS Signals bis zum Feststellen der Unterbrechung der Satellitenverbindung vermutet. Die letztere Position wird durch die Daten aus dem Inertialsensor ermittelt. Durch Mittelung der geschätzten Türposition über allen Bewegungsspuren soll dann die tatsächlich Position der Türe bestimmt werden.

Diese Methode zur Bestimmung der Türposition setzt voraus, dass ungefähr gleich viele Benutzer aus allen Richtungen auf das Gebäude zugehen. Dies muss nicht immer der Fall sein. Abbildung 3.1 zeigt eine Skizze der Situation vor dem Nordeingang des Informatikgebäudes der Universität Stuttgart. An diesem Eingang geht es Richtung Westen zur Bahn, der Mensa und dem Rest des Campus, während es nach Osten nur zu einer sehr selten genutzten Bushaltestelle und einem kleineren Parkplatz geht. Der Zugang aus dem Norden ist durch ein Hindernis verstellt. Die meisten Menschen kommen also aus Westen auf das Gebäude zu und bewegen sich teilweise auch sehr dicht an dessen Außenwänden. Demnach würde die geschätzte Türposition nach dieser Methode viel zu weit im Westen liegen.

3.1.2 Ankerpunkte

Da CrowdInside sowohl auf einen auf dem Fuß platzierten Sensor, als auch auf eine bestimmte Position des Smartphones am Körper verzichtet, gibt es keine Möglichkeit ZUPT durchzuführen. Stattdessen wird erst später bei der Berechnung des Innenraummodells

versucht aus den Daten zwischen Schritten zu unterscheiden und entsprechend die zurückgelegte Strecke und Richtung zu berechnen. Die somit gewonnenen Daten sind entsprechend schlechter als die im späteren Entwurf betrachteten. Es werden daher *Ankerpunkte* eingeführt, um die Spuren zu korrigieren. Ankerpunkte repräsentieren jeweils eine absolute Position im Gebäude, welche durch ein bestimmtes Muster in den Spuren auffällt. Taucht ein solches Muster in einer Bewegungsspur auf und ist ein entsprechender Ankerpunkt in der Nähe der vermuteten Benutzerposition, so wird die Position des Ankerpunktes neu berechnet und die aktuelle Position des Benutzers auf diese gesetzt. Gibt es mehrere Alternativen des selben Typs von Ankerpunkten, so wird zusätzlich die W-LAN Signalstärke erreichbarer Drahtlosnetzwerke verwendet.

Es wurde untersucht, dass sich Aufzüge, Treppen und Rolltreppen sehr gut als Ankerpunkte eignen. Zusätzlich werden Ankerpunkte bei verfügbarem GPS Signal oder bei Drehungen hinzugefügt.

Die Korrektur erfolgt bei CrowdInside ausschließlich auf Punkten. Das in dieser Arbeit vorgestellte System, welchem per ZUPT erfasste Daten zu Grunde liegen, verwendet hierzu Segmente. Per ZUPT werden gerade Stücke zuverlässiger erkannt als den bei CrowdInside zugrunde liegenden Daten.

3.1.3 Erfassen von Korridoren und Räumen

Um eine Repräsentation der Räume und Korridore zu erhalten, werden die Spuren segmentiert. Gerade Abschnitte werden unterteilt, wenn entweder der Benutzer eine 45° Drehung gegangen ist oder er längere Zeit am selben Ort gestanden hat.

Daraufhin werden Segmente unterteilt, je nachdem ob sie vermutlich in einem Raum oder auf einem Korridor aufgenommen wurden. Dabei wird zum einen die Länge der Segmente, wie auch die Geschwindigkeit des Benutzers und die Dichte der umherliegenden Schritte verwendet. Längere Abschnitte, in denen der Benutzer schneller gegangen ist und die eine hohe Anzahl an benachbarten Spuren haben, werden als Segmente in Korridoren erkannt. Für alle weiteren Segmente wird angenommen, dass diese in einem Raum erfasst wurden.

Um Segmente und Korridore zusammenzufassen, wird eine dichte-basierte Clusteranalyse auf die Punktwolken angewendet. Es werden also wieder Punkte und keine Segmente, wie im späteren Verfahren, betrachtet.

Zur Erfassung der geometrischen Ausdehnungen des Gebäudes, von Räumen und Korridoren werden Alpha Shapes verwendet. Alpha Shapes sind eine Verallgemeinerung der konvexen Hülle [Ede10]. Alle Schritte, welche aus dem entsprechenden Objekt erfasst wurden, werden als Punkte betrachtet. Daraufhin wird das Alpha Shape dieser Punkte berechnet und als geometrische Ausdehnung interpretiert.

Damit sich beim Bestimmen der geometrischen Ausdehnung keine Räume überlappen, wird das Alpha Shape jeweils aus Punkten zwischen den Schritten bestimmt. Zusätzlich wird zur Unterscheidung der Räume die Signalstärke verfügbarer Drahtlosnetzwerke verwendet.

Durch die Repräsentation der Geometrie mittels Alpha Shapes ist CrowdInside sehr fehleranfällig. Kann eine Spur nicht korrigiert werden und ragt sie über das Gebäude oder einen Korridor hinaus, so hat dies direkten Einfluss auf die erkannte Geometrie. Es gehen hier keine weiteren statistischen Methoden ein, um eine Bestimmung der Geometrie auch mit eventuell fehlerhaften Bewegungsspuren durchführen zu können.

3.2 FootSLAM

Das aus dem Bereich der Robotik kommende FootSLAM verwendet einen grundlegend anderen Ansatz [AR12, GPRH12, PRA13]. In diesem wird der zweidimensionale Raum durch Sechsecke unterteilt. Mittels eines Inertialsensors am Fuß des Benutzers und unter Verwendung von ZUPT werden wieder Schrittvektoren erfasst. Diese werden nun allerdings in ein dynamisches Bayes-Netz übertragen. Aus diesem wird schließlich eine Karte erzeugt.

3.2.1 Kartenerstellung mit Hilfe eines dynamischen Bayes-Netzes

In einem *Bayes-Netz* werden Abhängigkeiten von verschiedenen Daten oder Gegebenheiten modelliert. Zusätzlich dazu können in einem *dynamischen Bayes-Netz* diese Abhängigkeiten über Zeitschritte hinweg modelliert werden.

Abbildung 3.2 zeigt das dynamische Bayes-Netz, welches zur Kartenerstellung von FootSLAM verwendet wird. Alleine durch die Eingabe der gemessenen Bewegung kann mit diesem Netz für jede Kante eines Sechsecks eine Wahrscheinlichkeit angegeben werden, mit welcher diese vom Benutzer überschritten worden ist.

Ein Fußgänger verfolgt eine Absicht, für welche er mit Hilfe seiner Wahrnehmung und dem visuellen Eindruck seine Position und Richtung ändert. Die Wahrnehmung ist abhängig von der Karte. Diese spiegelt gewisse Gegebenheiten wie Wände wieder.

Die tatsächliche Bewegung ist abhängig von der Absicht im vorherigen Schritt. Letztendlich ist die gemessene Bewegung, welche vom Sensor erfasst wird, abhängig von einem Korrekturterm und der tatsächlichen Bewegung.

Mit Hilfe des dynamischen Bayes-Netzes kann nun unter der Verwendung eines Particle Filter die Wahrscheinlichkeit für jeden Übergang von Sechsecken angegeben werden. Dabei werden vorher durchlaufene Kanten bevorzugt. Hierdurch lassen sich zum einen Hindernisse wie Wände erkennen. Zum anderen verhindert man dadurch ein Abdriften über die Zeit, wenn Orte in regelmäßigen Abständen wieder besucht werden und somit ein „Einrasten“ an diese möglich ist.

Im Gegensatz zu CrowdInside und dem in dieser Arbeit vorgestellten Entwurf werden bei FootSLAM alle erfassten Daten auf einmal betrachtet. Dadurch ist eine Korrektur der Bewegungsspuren durch das dynamische Bayes-Netz auch rückwärts möglich.

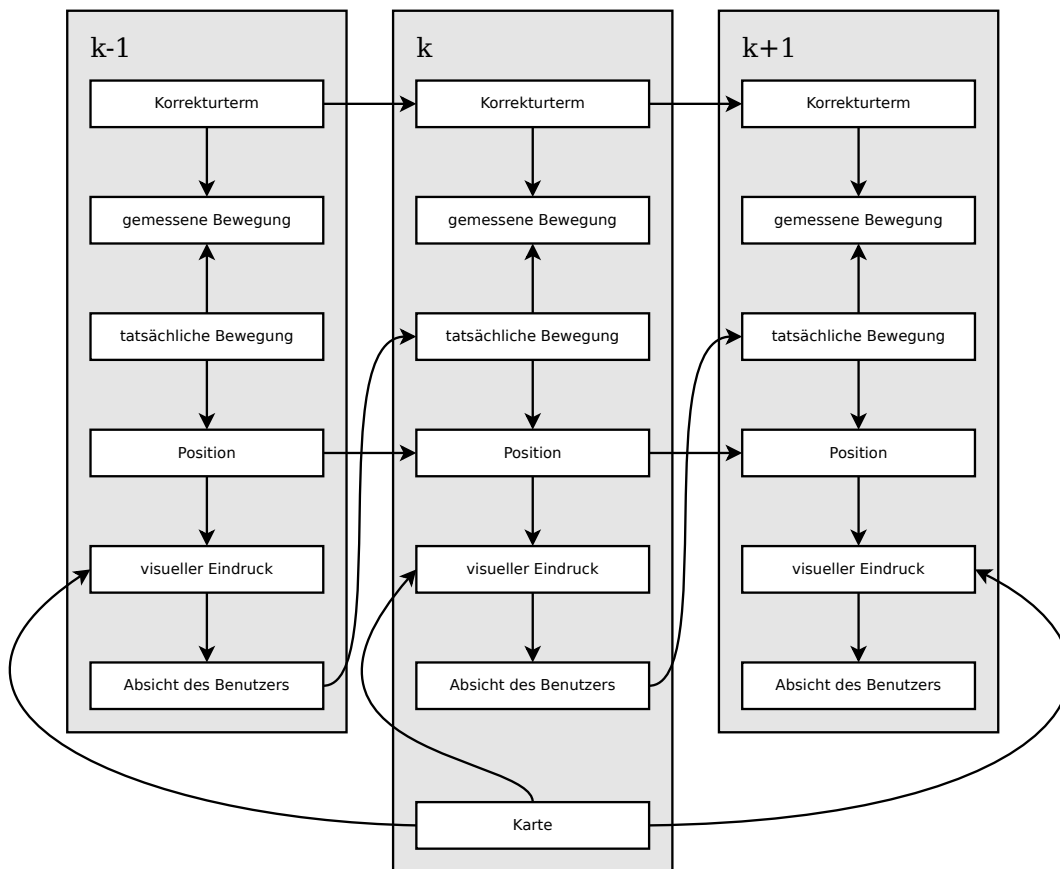


Abbildung 3.2: Das dynamische Bayes-Netz, mit welchem in FootSLAM die menschliche Bewegung modelliert wird. Die einzige Eingabe in das Netz ist die gemessene Bewegung.

Durch den sehr abstrakten Entstehungsprozess der Karte mit Hilfe des Netzes ist es nur schwer möglich, weitere Informationen, wie etwa ein Außenraummodell, in das System mit aufzunehmen. Dies erfordert eine komplette Umstrukturierung des Bayes-Netz. Auch liefert FootSLAM keine höhere Semantik über Räume oder Korridore.

3.2.2 Erweiterungen

Um andere Gegebenheiten oder Merkmale in FootSLAM einzubauen, muss das Bayes-Netz modifiziert werden. Auf diese Weise kann zum Beispiel die Signalstärke von in der Nähe befindlichen W-LANs in den Kartenerstellungsprozess einfließen [BR11]. Hierzu wird neben der Karte auch eine *Karte der W-LANs* modelliert. Im Bayes-Netz wirkt sich diese über eine weitere Zufallsvariable zur Korrektur auf die Position und Richtung des Benutzers aus. Die Daten über die Signalstärken gehen somit nur in die Bestimmung der Position und nur indirekt in die Karte ein.

Eine zusätzliche Erweiterung von FootSLAM ist PlaceSLAM [RAK10]. Hierbei gehen Merkmale, welche vom Benutzer erfasst werden, wie zum Beispiel Türen oder Wandbilder, in das modifizierte Bayes-Netz ein. Auch können auf diese Weise von Maschinen lesbare Gegebenheiten, wie RFID Tags in den SLAM Prozess eingebunden werden. Voraussetzung ist, dass diese Eigenschaften wie Objekte, die vom Benutzer auf zwei Meter genau erkannt werden können, besitzen.

FeetSLAM ist eine Erweiterung von FootSLAM, welche die kooperative Erfassung des Innenraummodells durch mehrere Benutzer ermöglicht [RPA11]. Hierbei werden die einzelnen Fußgänger wie in FootSLAM modelliert. Der einzige Unterschied besteht in der Modellierung der Karte. Diese wird als ein einzelner Knoten, von welchem die visuellen Eindrücke aller Benutzer abhängig sind, modelliert.

3.3 Innenraummodelle durch abfotografieren von Fluchtplänen

Eine weitere, sehr einfache, aber durchaus interessante Methode um Innenraummodelle von Gebäuden zu erhalten, ist deren Fluchtpläne abzufotografieren [PHF11]. Fluchtpläne enthalten viele Informationen, wie das Stockwerk, die Ausrichtung und Topologie des Gebäudes.

Diese Methode scheitert allerdings zum einen, wenn es keinen Fluchtplan gibt, wie etwa in privaten oder nicht-öffentlichen Gebäuden. Zum anderen kann sich das Gebäudemodell, zum Beispiel durch neu oder umgestellte Hindernisse, ändern, so dass der Fluchtplan aktualisiert werden muss. Es kann dann einige Zeit vergehen, bis dieser geänderte Umstand einem Benutzer auffällt und er den neuen Fluchtplan abfotografiert und die entsprechenden Änderungen in das System übernommen werden.

Fluchtpläne geben nur die Topologie, nicht aber die genaue Geometrie wieder. Zudem sind in diesen nicht alle Türen, sondern nur Fluchttüren eingezeichnet. Einige Details, wie die Orientierung von Treppen (nach oben oder unten) lassen sich nur mit Tricks, wie zum Beispiel der Annahme, dass Fluchtwege meist nach unten führen, herausfinden.

4 Systemmodell und Anforderungen

In diesem Kapitel wird das Systemmodell und die Anforderungen an den Entwurf des Systems zur Erfassung von Innenraummodellen beschrieben.

4.1 Systemmodell

Das System besteht aus *mobilen Einheiten*, welche durch zu Fuß gehende Benutzer in *alltäglichen Situationen* durch ein Gebäude getragen werden. Die mobile Einheit besteht aus einem *Smartphone* und eine auf dem Fuß des Benutzers befestigte *Sensoreinheit*. Durch diese Sensoreinheit wird die Bewegung des Benutzers aufgezeichnet. Die Daten werden über das Smartphone an einen *Server* in der Cloud übertragen. Auf diesem wird ein Innenraummodell des Gebäudes konstruiert. Im Folgenden werden die einzelnen Komponenten im Detail beschrieben.

4.1.1 Bewegungsspuren

Bewegungsspuren bestehen aus einer Startposition p_0 und einer Liste von geordneten Schrittvektoren $S = (s_1, \dots, s_n)$. Zusätzlich wird beim Start eines Schrittes die Zeit als $s_i.zeit$ festgehalten.

Die erfasste Position des Benutzers p_k nach dem Schritt k lässt sich berechnen durch

$$p_k = p_0 + \sum_{i=1}^k s_i.$$

Die Liste der absoluten Punkte $(p_i)_{i=0, \dots, n}$ nennt man auch die *absoluten Punkte einer Spur*.

Ein *Segment* ist ein Abschnitt in einer Spur, in welcher der Benutzer keine Kurve gegangen ist. Segmente in einer Spur lassen sich als ein Tupel von Schrittnummern darstellen. Das Segment (j, k) fängt mit dem Start von Schritt j an und hört mit dem Ende von Schritt k auf. Als *absolute Punkte eines Segments* versteht man entsprechend die Liste $(p_i)_{i=j, \dots, k}$.

Korridorsegmente sind Segmente, in welchen die Summe der Schrittlängen $|s_i|$ länger als L_{\min} sind. Die Menge aller Korridorsegmente einer Spur s wird mit $K(s)$ bezeichnet.

Es ist anzunehmen, dass die erfasste Spur des Benutzers $(p_i)_{i=0, \dots, k}$ mit der Zeit immer weiter von der real gegangenen Spur $(\hat{p}_i)_{i=0, \dots, \hat{k}}$ abweicht. Dies beruht auf ungenauen Messungen von s_i . Auch kann die initiale Position p_0 von der realen Startposition \hat{p}_0 abweichen. Selbst

die Anzahl der erfassten Schritte k muss nicht der Anzahl der real gegangenen Schritte \hat{k} übereinstimmen.

4.1.2 Mobile Einheiten

Sensoreinheit

Die Sensoreinheit zeichnet die Positionsänderungen, welche als $S = (s_1, \dots, s_n)$ in die Bewegungsspuren eingehen, auf. Hierzu ist in die Sensoreinheit ein Inertialsensor eingebaut. Mit diesem können per ZUPT die Schrittvektoren s_i bei jedem Schritt berechnet werden. Zusätzlich zu der Positionsänderung wird die Zeit bei jedem Start eines Schrittes festgehalten.

Neben dem Inertialsensor ist in der Sensoreinheit auch ein Modul für drahtlose Nahbereichskommunikation enthalten. Dieses steht in ständiger Verbindung mit dem Smartphone. Bei jedem von der Sensoreinheit erfassten Schritt wird der Änderungsvektor durch dieses Kommunikationsmodul an das Smartphone gesendet.

Smartphone

Das Smartphone steht zum einen in ständiger Verbindung mit der Sensoreinheit. Zum anderen unterhält es per drahtloser Kommunikation, zum Beispiel via W-LAN oder 3G/4G Mobilfunk, eine Verbindung zum Internet. Das Aufzeichnen der Spur geschieht opportunistisch und ohne weitere Benutzerinteraktion.

Die Daten werden von dem Smartphone auf einen Server hochgeladen. Dies geschieht periodisch oder wenn seit dem letzten Hochladen neue Daten angefallen sind. Der Server ist für die weitere Verarbeitung zuständig.

Es wird angenommen, dass ein geeigneter Mechanismus existiert, um mit dem Smartphone den Startpunkt p_0 festzulegen. Dies kann zum Beispiel durch periodisches Abfragen der GPS Position oder durch Auswählen der aktuellen Position auf einer ersten Karte durch den Benutzer erfolgen.

4.1.3 Mobilität der mobilen Einheit

Die mobile Einheit wird von Benutzern getragen. Benutzer sind Fußgänger, welche zur Erfassung des Gebäudes beitragen möchten. Sie bewegen sich dazu in alltäglichen Situationen in einem Stockwerk eines Gebäudes. Unter alltäglichen Situationen sind etwa der Büroalltag oder das Suchen von Räumen in einem Gebäude zu verstehen. Die körperlichen Aktivitäten der Benutzer sind hierbei auf Gehen, Stehen oder Sitzen beschränkt. Benutzer wechseln von Zeit zu Zeit den Raum und bewegen sich dabei auf Korridoren.

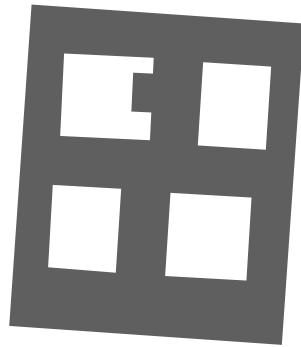


Abbildung 4.1: Außenraummodell des Informatikgebäudes der Universität Stuttgart aus OpenStreetMap. Graue Flächen befinden sich im Inneren des Gebäudes. Die vier Innenhöfe sind gut zu erkennen. Stand: 28. September 2013.

4.1.4 Server

Der Server empfängt die Bewegungsspuren von sämtlichen Benutzern im System. Er speichert die Spuren entsprechend und konstruiert aus diesen periodisch ein neues Gebäudemodell. Dies kann zum Beispiel täglich oder wenn mehr als 1% neue Spuren seit der letzten Innenraummodellierung in einem Gebäude erfasst wurden, passieren. Es wird angenommen, dass der Server über eine ausreichende Internetanbindung sowie genügend Speicherplatz und Rechenkapazität verfügt.

Zur Generierung des Innenraummodells stehen dem Server neben sämtlichen Bewegungsspuren auch das *Außenraummodell* eines Gebäudes zur Verfügung. Es wird angenommen, dass ein Dienst zur Abfrage von Außenraummodellen über das Internet erreichbar ist. Der Server kann über eine absolute Position, wie etwa den Startpunkt p_0 einer Bewegungsspur, das Außenraummodell des naheliegendsten Gebäudes abfragen.

4.1.5 Außenraummodelle

In einem Außenraummodell eines Gebäudes ist verzeichnet, welche Bereiche sich innerhalb eines Gebäude befinden und welche außerhalb. Abbildung 4.1 zeigt als Beispiel das Außenraummodell des Informatikgebäudes der Universität Stuttgart.

4.1.6 Räume und Korridore

Der Innenraum, welcher erfasst werden soll, besteht aus Räumen und Korridoren. Die Ausdehnung von Räumen und Korridoren wird als Rechtecke angenommen.

Räume besitzen exakt eine Türe. Diese Türe verbindet einen Korridor und einen Raum. Türen werden als Punkte auf der gemeinsamen Wand zwischen Korridor und Türe modelliert. Bei

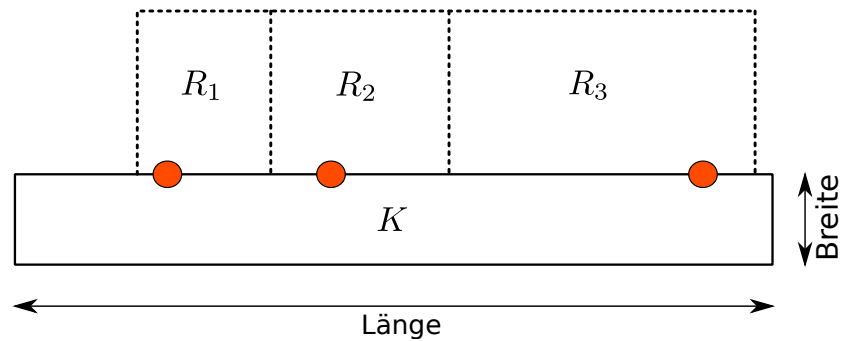


Abbildung 4.2: Beispiel eines Korridors. Türen sind in orange eingezeichnet.

der Repräsentation durch Rechtecke ist eine Seite des Raumes vollständig in einer Seite des Korridors enthalten.

Korridore sind entweder parallel oder orthogonal zu Außenwänden ausgerichtet. Sie haben eine längere und eine kürzere Ausdehnung. Es wird angenommen, dass sich Benutzer häufiger entlang der längeren Ausdehnung bewegen. Die längere Ausdehnung bezeichnet man auch als *Länge des Korridors*, die kürzere Ausdehnung als die *Breite des Korridors*. Die Menge aller Korridore nennt man auch das *Korridorskelett*.

Abbildung 4.2 zeigt ein Beispiel eines Korridors K mit drei anliegenden Räumen R_1 , R_2 und R_3 . Jeder Raum besitzt genau eine Tür zum Korridor. Die längere Ausdehnung, welche die Länge des Korridors definiert, ist deutlich von der kürzeren zu unterscheiden.

4.2 Anforderungen

Es soll ein System zur Erfassung von Innenraummodellen, welches zwischen Räumen und Korridoren unterscheidet, entworfen werden. Dieses muss auch mit ungenauen Sensordaten umgehen können.

4.2.1 Opportunistisches Aufzeichnen von Bewegungsspuren

Um möglichst viele Bewegungsspuren zur Verfügung zu haben, soll die Erfassung der Bewegungsspuren opportunistisch erfolgen. Benutzer, welche Daten für das Erstellen des Innenraummodells bereitstellen möchten, zeichnen diese ohne weitere Interaktion mit ihrem Smartphone auf.

4.2.2 Erfassen des Korridorskeletts

Aus den Bewegungsspuren soll ein möglichst gut mit der Realität übereinstimmendes Korridorskelett konstruiert werden. Neben den topologischen Zusammenhängen sollen hierbei auch die geometrischen Abmessungen möglichst exakt erfasst werden.

Topologische Zusammenhänge

Unter den topologischen Zusammenhängen versteht man, dass die Abzweigungen aus den einzelnen Korridoren mit der Realität übereinstimmen. Die topologischen Zusammenhänge sollen bereits mit wenigen Spuren gefunden werden.

Stehen einige Spuren, welche alle möglichen Abzweigungen aus einem Korridor K in alle angrenzenden Korridore genommen hatten, zur Verfügung, so sollen diese Abbiegungen aus K auch richtig erkannt werden.

Um die Topologie richtig zu erkennen, muss erkannt werden, welche Teile aus unterschiedlichen Spuren im selben Korridor erfasst wurden.

Bestimmung der geometrischen Ausmaße

Zusätzlich zu den topologischen Zusammenhängen der Korridore sollen auch deren geometrische Ausmaße erkannt werden. Wie vorher definiert, müssen Korridore als Rechtecke modelliert werden. Für diese muss eine Breite und eine Länge gefunden werden.

Die Länge kann unter Umständen nur dann richtig erkannt werden, wenn Benutzer auch die volle Länge des Korridors gegangen sind. Sind durch die Spuren nur Abschnitte des Korridors erfasst worden, so wird nur eine Erfassung der Länge dieser Abschnitte gefordert.

4.2.3 Erfassen der Räume

Von Räumen soll neben den geometrischen Ausmaßen auch die Position der Türe erfasst werden. Es soll erkannt werden, an welchen Korridor ein Raum angrenzt.

Erkennen der Topologie von Räumen

Es wird davon ausgegangen, dass Räume nur eine Türe haben. Diese verbindet jeweils einen Raum mit einem Korridor. Raum und Korridor werden durch eine gemeinsame Wand getrennt. Die Nachbarschaftsbeziehung zwischen Räumen soll korrekt erkannt werden.

Befindet sich zum Beispiel Raum a südlich von Raum b auf der selben Seite eines Korridors und wurden beide Räume durch Spuren besucht, so soll sich auch in dem erzeugten Innenraummodell der Raum a südlich von Raum b im entsprechenden Korridor befinden.

Erkennen der Geometrie von Räumen

Für die geometrische Erfassung von Räumen soll neben den Ausdehnungen des Raumes eine Türposition festgelegt werden. Wie bereits definiert wird die Ausdehnung eines Raumes als Rechteck modelliert. Für die Tür soll ein Punkt auf der Wand zwischen Raum und Korridor festgelegt werden. Beide Merkmale sollen möglichst genau erfasst werden.

Das erzeugte Gebäudemodell darf keine Überlappungen zwischen Räumen oder zwischen Räumen und Korridoren enthalten.

4.2.4 Umgang mit ungenauen Sensordaten

Generell soll das System mit ungenauen Sensordaten umgehen können. Da die Bewegungsspuren über die Zeit einer Drift ausgesetzt sind, soll eine Methode entwickelt werden, um diese zu korrigieren. Spuren, für welche eine solche Korrektur nicht möglich ist, sollen erkannt und verworfen werden. Trotzdem soll das System möglichst viele Spuren für die Erfassung des Innenraummodells verwertbar machen. Die Erfassung der Geometrie soll robust und nicht von einzelnen Spuren abhängig sein.

5 Entwurf

In diesem Kapitel wird beschrieben, wie durch Bewegungsspuren und mögliche Ausrichtungen der Korridore ein Innenraummodell erstellt werden kann.

Als Eingabe werden Bewegungsspuren erwartet. Diese bestehen aus Änderungsvektoren pro Schritt. Es wird angenommen, dass die Bewegungsspuren durch einen Inertialsensor auf dem Fuß mittels ZUPT aufgezeichnet wurden.

Eine weitere Eingabe ist die Menge der möglichen Orientierungen von Korridoren. Diese wird aus einem Außenraummodell, welches von einem Kartendienst wie OpenStreetMap bereitgestellt wird, extrahiert. Die Orientierungen werden hierbei lediglich zur Korrektur von längeren Segmenten, wie in Abschnitt 2.2.3 beschrieben, benötigt.

5.1 Übersicht über den Kartenerstellungsprozess

Der Prozess zum Erstellen einer Karte gliedert sich wie folgt (siehe auch Abbildung 5.1):

Erfassung der Daten Grundlage für das Erfassen von Innenräumen bilden per ZUPT gewonnene Schrittvektoren. Diese werden von Benutzern erfasst und übertragen, so dass sie dem Kartenerstellungsdienst zur Verfügung stehen.

Segmentierung und Außenwandkorrektur Jede Bewegungsspur wird zunächst einzeln und unabhängig von den anderen Spuren korrigiert. Dazu wird die Spur segmentiert und Korridorsegmente anhand von möglichen Orientierungen durch das Außenraummodell ausgerichtet.

Erstellen eines Basisplans Als nächstes wird ein initiales Korridorskelett erstellt, welches auch Basisplan genannt wird. Für diesen werden möglichst verlässliche Spuren gesucht, an denen später die Qualität der anderen Spuren festgestellt werden kann. Durch eine Relation können Korridorsegmente zusammengefasst und mit Korridoren identifiziert werden.

Iteratives Hinzufügen von Spuren auf das Korridorskelett Nun werden iterativ Spuren auf das Korridorskelett hinzugefügt. Hierzu werden in jedem Schritt alle noch nicht hinzugefügten Spuren bewertet. Scheinen diese auf das Korridorskelett zu passen, so werden sie hinzugefügt.

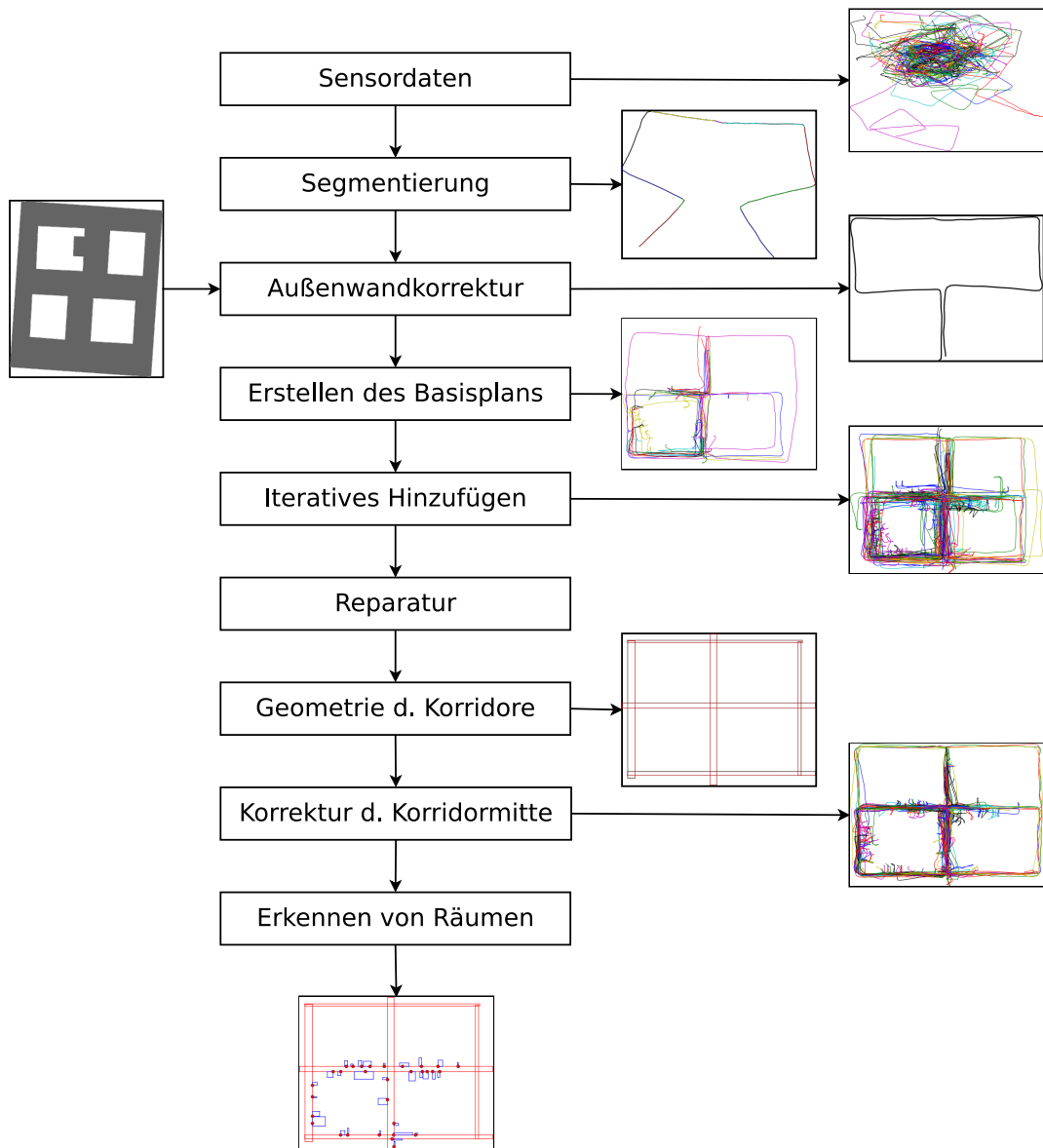


Abbildung 5.1: Überblick über die einzelnen Schritte der Innenraummodellierung.

Reparaturschritt Manche Spuren konnten nach dem vorherigen Schritt noch nicht auf das Skelett hinzugefügt werden. In diesem Verarbeitungsschritt wird versucht diese unpassenden Spuren auf das Korridorskelett anzupassen. Dazu wird eine Spur beispielsweise gedreht und mit Hilfe der Bewertung vom vorherigen Schritt entschieden, mit welcher Drehung eine Spur am besten auf das Korridorskelett passt. Dies ist der letzte Schritt, um weitere Spuren auf das Korridorskelett hinzuzufügen.

Geometrische Repräsentation von Korridoren Nachdem das Korridorskelett vollständig bekannt ist, gilt es nun eine geometrische Repräsentation von diesem zu bestimmen. Hierzu wird die Breite von Korridoren mit Quantilen und die Länge von diesen durch die empirische Verteilungsfunktion festgelegt.

Korrektur der Spuren durch die Korridormitte Durch die bekannte geometrische Ausdehnung der Korridore ergibt sich eine neue Möglichkeit, um Spuren auf dem Skelett erneut zu korrigieren. Hierzu werden Korridorsegmente auf die Mitte ihrer Korridore gelegt.

Erkennen von Räumen Mit dem vorherigen Korrekturschritt ist es nun möglich, Räume geometrisch aus den Spuren zu erkennen. Es wird eine Türposition und Ausdehnung festgelegt.

Zusammenfügen von Räumen Wurden Räume in den einzelnen Spuren gefunden, so wird versucht diese nun zusammenzufügen. Hierzu wird die Ausdehnung und die Position der Türe herangezogen.

5.2 Erfassung der Daten

Als Grundlage für das Erfassen von Innenräumen dienen Bewegungsdaten, welche mittels ZUPT von einem Inertialsensor auf dem Fuß erfasst wurden. Diese bestehen aus einem Startpunkt p_0 , sowie einer Menge von Schrittvektoren $S = \{s_1, \dots, s_n\}$. Bei jedem Schritt eines Benutzers wird ein neuer 2D Änderungsvektor berechnet.

Es wird angenommen, dass eine geeignete Technik zum Feststellen des Startpunktes p_0 und der initialen Bewegungsrichtung gegeben ist. Denkbar ist eine Bestimmung der Türposition durch GPS, wie in CrowdInside. Eine andere Möglichkeit den Startpunkt festzulegen ergibt sich durch eine bereits bestehende Karte. Der Benutzer kann auf dieser seinen Startpunkt und eine Startrichtung wählen.

5.3 Segmentierung und Außenwandkorrektur

Um mit den Spuren arbeiten zu können, wird zunächst die Drift in den Spuren durch Anwenden der Außenwandkorrektur begrenzt. Dabei wird jede Bewegungsspur einzeln und unabhängig von den anderen Spuren korrigiert. Um die Außenwandkorrektur anwenden zu können, müssen die Spuren zunächst segmentiert werden.

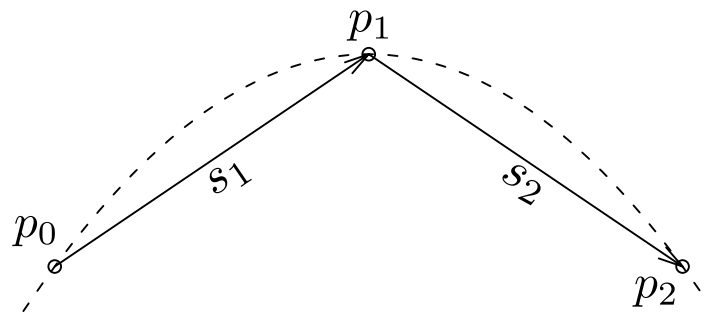


Abbildung 5.2: Segmentierung durch Polynominterpolation

5.3.1 Segmentierung der Bewegungsspuren

Als erster Bearbeitungsschritt werden die Spuren in gerade und zusammenhängende Stücke, welche durch Kurven getrennt werden, unterteilt (siehe Abbildung 5.4). In Abschnitt 2.2.1 wurde bereits eine Methode, um eine Spur in Segmente zu unterteilen, vorgestellt. Hierzu wurde die Abweichung der Richtung von zwei aufeinanderfolgenden Schritten herangezogen um zu entscheiden, ob diese Schritte zum selben Segment gehören.

Im Folgenden werden zwei weitere Segmentierungsmethoden vorgestellt. Hierbei werden jeweils die drei Punkte von zwei aufeinanderfolgenden Schritten betrachtet. Bei der ersten Methode werden diese Punkte durch ein Polynom zweiten Grades interpoliert und anhand des quadratischen Koeffizienten entschieden, ob die zwei Schritte zum selben Segment gehören. Die zweite Methode verwendet die Höhe des durch die drei Punkte entstandenen Dreiecks, um ebendieser Fragestellung nachzugehen.

Verwendung von Polynomen zur Segmentierung

Die drei Punkte zweier aufeinanderfolgender Schritte kann man nach geeigneter Drehung als Polynom zweiten Grades interpolieren. Der konstante und lineare Koeffizient gibt dann die gerade Fortbewegung wieder. Der quadratische Koeffizient hingegen gibt die Krümmung der Spur bei diesen zwei Schritten an. Dieser soll verwendet werden, um zu unterscheiden, ob zwei Schritte zum selben Segment gehören.

Formal ist zu den Schritten i und $i + 1$ einer Spur durch $p_{i-1} = (x_{i-1}, y_{i-1})$ der Startpunkt von Schritt Nummer i und durch $p_i = (x_i, y_i)$ der Endpunkt von diesem gegeben. Der Punkte p_i ist hierbei gleichzeitig der Startpunkt von Schritt $i + 1$. Der Endpunkt von Schritt $i + 1$ ist durch $p_{i+1} = (x_{i+1}, y_{i+1})$ gegeben (siehe Abbildung 5.2).

Man dreht nun die drei Punkte um p_{i-1} bis $y_{i-1} = x_{i+1}$ gilt und interpoliert diese durch ein quadratisches Polynom $a_0 + a_1x + a_2x^2$. Ist hierbei $|a_2|$ kleiner als ein festgelegter Schwellwert, so gehören s_1 und s_2 zum selben Segment. Ist $|a_2|$ größer, so weichen die drei Punkte von einer Linie zu sehr ab und gehören somit zu unterschiedlichen Segmenten.

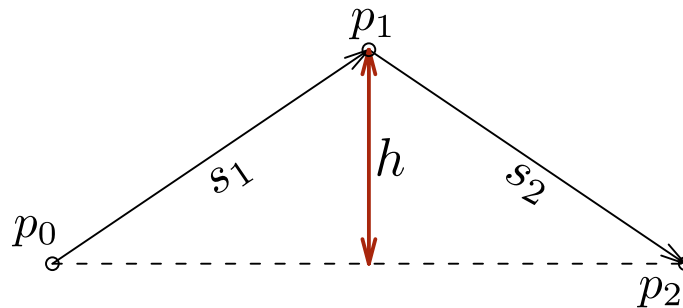


Abbildung 5.3: Segmentierung anhand der Höhe des Dreiecks

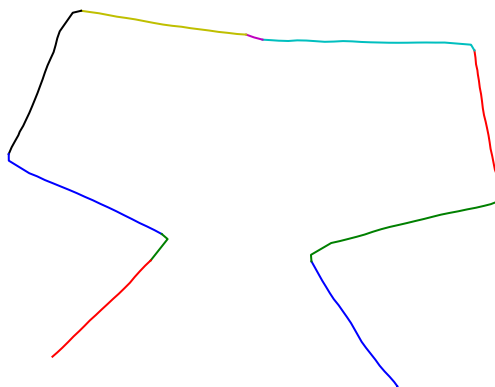


Abbildung 5.4: Reales Beispiel einer Segmentierung. Gerade Stücke der Bewegungsspur werden in ein Segment zusammengefasst. Die unterschiedlichen Farben grenzen aufeinanderfolgende Segmente ab. Die beiden Enden (rot und blau) wurden im selben Korridor aufgenommen. Dies zeigt exemplarisch die mit der Zeit größer werdende Abweichung der Daten aus der IMU.

Segmentierung durch Höhe des Dreiecks

Bei der Segmentierung durch die Höhe des Dreiecks betrachtet man wieder zwei aufeinanderfolgende Schritte s_i, s_{i+1} und das daraus entstandene Dreieck. Dieses Dreieck setzt sich zusammen aus dem Startpunkt des ersten Schrittes $p_{i-1} = (x_{i-1}, y_{i-1})$, dem Endpunkt des ersten Schrittes, welcher gleichzeitig auch Startpunkt des zweiten Schrittes ist, $p_i = (x_i, y_i)$ und dem Endpunkt des zweiten Schrittes, $p_{i+1} = (x_{i+1}, y_{i+1})$ (siehe Abbildung 5.3).

Man unterscheidet nun anhand der Höhe dieses Dreiecks h , ob sich die Schritte s_i und s_{i+1} im selben Segment befinden. Ist $h \leq h_{\max}$, so befinden sich s_i und s_{i+1} im selben Segment. Ist h größer als h_{\max} , so befinden sich die Schritte in unterschiedlichen Segmenten.

Abbildung 5.4 zeigt, wie eine Segmentierung mittels dieser Methode auf einer Bewegungsspur aussehen kann. Die maximale Höhe h_{\max} wurde dort auf 15 cm festgelegt.

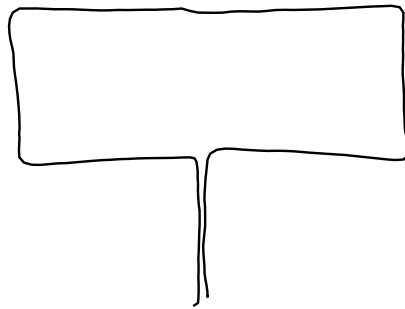


Abbildung 5.5: Beispiel einer an Außenwänden ausgerichteten Spur. Dargestellt ist die Spur aus Abbildung 5.4 nach der Außenwandkorrektur.

5.3.2 Außenwandkorrektur

Korridorsegmente Ein Ziel ist es, zu unterscheiden, ob Segmente in einem Raum oder auf einem Korridor erfasst wurden. Wie bereits in Abschnitt 2.2.2 beschrieben, werden längere Segmente auch *Korridorsegmente* genannt. Ein Korridorsegment muss wegen seiner Länge in einem Korridor gegangen worden sein. Für kleine Segmente kann man hingegen nicht unterscheiden, ob diese in einem Raum oder in einem Korridor erfasst wurden.

Typischerweise geht man in Räumen anders als auf Korridoren [AY12]. Dies liegt zum einen daran, dass in Räumen mehr Platz durch Möbel verstellt ist, so dass man gar nicht über eine längere Strecke geradeaus gehen kann. Zum anderen möchte man auf Korridoren meist möglichst schnell von Raum *a* zu Raum *b* kommen und geht daher gerade.

Außenwandkorrektur Durch die bereits in Abschnitt 2.2.3 vorgestellte Korrektur anhand von Außenräumen kann die Drift einer Spur deutlich verbessert werden (siehe Abbildung 5.5). Hierzu werden Korridorsegmenten eine Orientierung orthogonal oder parallel zu Außenwänden zugewiesen.

Entscheidet sich die Außenwandkorrektur für eine falsche Ausrichtung eines Korridorsegments, so kann dies zu großen Fehlern in der Spur führen. Gibt es zum Beispiel nur zwei Ausrichtungen und steht ein Korridorsegment zwischen diesen, so entscheidet eine leichte Beugung, ob das Segment in die eine oder in die andere Richtung ausgerichtet wird. Es kann zum Zeitpunkt der Korrektur nicht festgestellt werden, ob ein Korridorsegment falsch ausgerichtet worden ist.

5.4 Erstellen eines Basisplans

Ein generelles Problem beim Erstellen eines Innenraummodells sind falsche Daten. Die Korrektur durch Außenwände verhindert zwar das Abdriften der Spuren, jedoch fügt diese

teilweise auch weitere Fehler ein. Ein Ziel ist es also herauszufinden, welche Spuren korrekt sind und gut mit der wirklich gegangenen Spur übereinstimmen und welche nicht.

Um mit diesem Problem umzugehen, wird zunächst die Korridorrelation definiert. Mit dieser kann man feststellen, welche Teile der Spuren auf dem selben Korridor gegangen wurden. Teile der Spuren können so zueinander korreliert werden. Dadurch ist es beispielsweise möglich, die Anzahl der Korridorsegmente auf einem Korridor als Indiz für die Korrektheit der dazugehörigen Spuren zu verwenden.

5.4.1 Die Korridorrelation

Um mehrere Korridorsegmente einem Korridor zuordnen zu können, wird die Korridorrelation definiert. Diese berücksichtigt sowohl die Ausrichtung als auch den räumlichen Zusammenhang, um mehrere Korridorsegmente zusammenfassen zu können.

Zur Definition der Korridorrelation wird im Folgenden zunächst der Begriff eines umschließenden Rechtecks einer Spur definiert. Um eine Toleranz für den räumlichen Zusammenhang herzustellen, werden diese Rechtecke entsprechend erweitert. Letztendlich wird zunächst eine Relation für benachbarte Korridorsegmente definiert. Der transitive Abschluss dieser Relation ist dann die Korridorrelation.

Umschließende und x -umschließende Rechtecke

Wie bereits in Abschnitt 2.2.3 angemerkt, verlaufen Korridore meist orthogonal oder parallel zu Außenwänden. Als geometrische Darstellung von Korridoren werden Rechtecke verwendet. Diese Rechtecke werden entsprechend den Außenwänden orientiert. Sie haben also eine Seite entweder orthogonal oder parallel zu einer Außenwand.

Das Rechteck mit minimalem Flächeninhalt, welches alle absoluten Punkte eines Segments enthält und welches nach einer Außenwand orientiert ist, nennt man das *umschließende Rechteck*.

Das umschließende Rechteck eines Korridorsegments besitzt eine kurze und eine lange Ausdehnung. Das *x -umschließende Rechteck* erhält man aus dem umschließenden Rechteck, indem man auf jeder Seite der kürzeren Ausdehnung x Meter hinzufügt (siehe Abbildung 5.6). Die kürzere Ausdehnung des umschließenden Rechtecks wird somit beim x -umschließenden Rechteck um $2 \cdot x$ verlängert.

Benachbarte Korridorsegmente

Durch das x -umschließende Rechteck wird nun eine Relation auf den Korridorsegmenten definiert. Zwei Korridorsegmente sind *benachbart*, wenn diese parallel zur selben Außenwand ausgerichtet sind und die x -umschließenden Rechtecke der Segmente sich überschneiden.

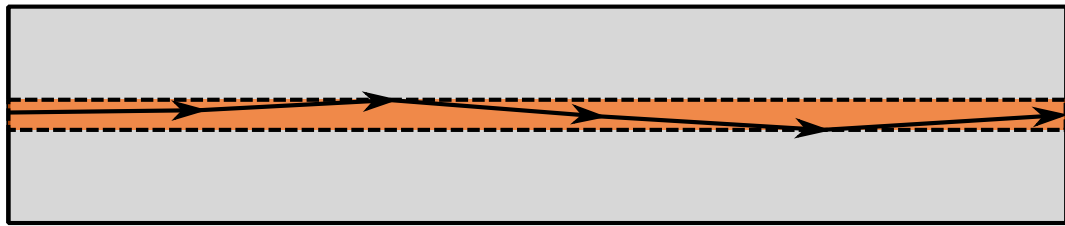


Abbildung 5.6: Beispiel eines x -umschließenden Rechtecks (grau) und eines umschließenden Rechtecks (orange) einer Spur.

Der Parameter x legt hierbei die maximale Distanz zweier benachbarter Korridorsegmente fest.

Wie man leicht sieht, ist jedes Korridorsegment zu sich selbst benachbart. Wenn ein Korridorsegment k_1 zu einem Korridorsegment k_2 benachbart ist, so ist auch k_2 zu k_1 benachbart. Die Relation der benachbarten Korridorsegmente ist also symmetrisch und reflexiv.

Korridore und Korridorrelation

Die vorhergehenden Definitionen erlauben nun die Definition der Korridorrelation. Diese soll festlegen, wann sich zwei Korridorsegmente auf dem selben Korridor befinden. Die benachbarte Korridorrelation ist hierzu noch nicht geeignet. Sind k_1 und k_2 , sowie k_2 und k_3 benachbart, so ist nicht garantiert, dass auch k_1 und k_3 benachbart sind. Die Korridorsegmente k_1 und k_3 haben allerdings die selbe Ausrichtung und da beide zu k_2 benachbart sind, sollten sie auch räumlich nicht zu weit getrennt sein. Man möchte also eine Relation, in welcher auch k_1 und k_3 in einem solchen Fall in Relation stehen.

Die *Korridorrelation* ist definiert als die transitive Fortsetzung der Relation der benachbarten Korridorsegmente.

Wie bereits festgestellt, ist die Relation der benachbarten Korridorsegmente symmetrisch und reflexiv. Durch Konstruktion erbt die Korridorrelation diese Eigenschaften. Zusätzlich ist sie nach Definition auch transitiv. Somit ist die Korridorrelation eine Äquivalenzrelation. Also besitzt jedes Korridorsegment eine Äquivalenzklasse.

Betrachtet man eine Äquivalenzklasse der Korridorrelation, so sind alle Korridorsegmente in dieser parallel zur selben Außenwand und durch die erweiterten Rechtecke räumlich sehr nah. Es ist also anzunehmen, dass die Elemente einer Äquivalenzklasse auf demselben Korridor erfasst wurden. Man identifiziert daher Äquivalenzklassen der Korridorrelation mit Korridoren und verwendet die Begriffe synonym.

Den Korridor eines Korridorsegments k bezeichnet man – angelehnt an die übliche Notation von Äquivalenzklassen – mit $[k]$. Zu einer Menge von Segmenten $K = \{k_1, \dots, k_n\}$ bezeichnet $\mathcal{F}_K = \{[k] : k \in K\}$ die Menge aller Korridore aus K . Diese Menge nennt man auch *Flurskelett*.

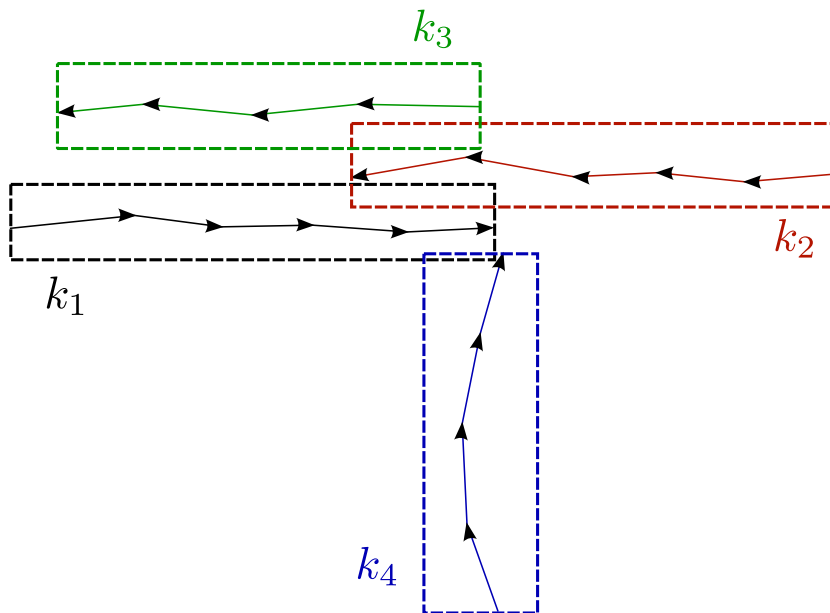


Abbildung 5.7: Beispiel zur Korridorrelation

Abbildung 5.7 zeigt ein Beispiel zu benachbarten Korridorsegmenten, x -umschließenden Rechtecken und der Korridorrelation. Dargestellt sind vier Korridorsegmente mit ihren x -umschließenden Rechtecken. Die Segmente k_1 und k_2 sind benachbart. Ebenso wie Segment k_2 und k_3 . Die Segmente k_1 und k_3 sind jedoch nicht benachbart. Trotzdem sind in der Korridorrelation k_1, k_2 und k_3 in der selben Äquivalenzrelation. Das Korridorsegment k_4 besitzt eine andere Ausrichtung, wodurch es weder zu k_1 benachbart, noch im selben Korridor wie k_1, k_2 und k_3 ist.

5.4.2 Finden geeigneter Spuren für den Basisplan

Ein Problem sind durch die Außenwandkorrektur falsch ausgerichtete Korridorsegmente. Durch deren Länge macht eine Falschausrichtung die komplette Spur ab Beginn des Korridorsegments unbrauchbar. Erschwerend kommt hinzu, dass es bisher keine Möglichkeit gibt überhaupt festzustellen, wann eine Spur mit der real gegangenen Spur größtenteils übereinstimmt.

Gelöst wird dieses Problem dadurch, dass durch einige Klassifikatoren sehr wenige Spuren ausgesucht werden, welche von hoher Qualität zu sein scheinen. Dies kann zum Beispiel durch den Winkel, um welchen ein Korridorsegment bei der Außenwandkorrektur gedreht werden musste, sein. Das Flurskelett dieser ausgewählten Spuren nennt man den *Basisplan*. Ist ein Basisplan erstellt, so kann später für weitere Spuren entschieden werden, ob diese hinzugefügt werden sollen oder nicht. Dies wird anhand der Korridorsegmente und mit Hilfe der Korridorrelation entschieden.

Im Folgenden werden einige Klassifikationen vorgestellt, mit denen man möglichst gute Bewegungsspuren aus allen Spuren herausfiltern kann.

Klassifikation durch die Außenwandkorrektur

Eine Möglichkeit der Klassifikation ist durch den Winkel, um welchen das erste Korridorsegment bei der Außenwandkorrektur gedreht werden muss, gegeben. Der Winkel $\alpha \in (-\pi, \pi]$, um welchen das erste Korridorsegment gedreht werden muss, um einer durch die Außenwände vorgegebenen Orientierung zu entsprechen, sollte hierbei im Betrag möglichst klein sein. Hierzu setzt man einen maximalen Winkel α_{\max} fest. Ist $|\alpha| \leq \alpha_{\max}$, so wird die Spur als gut betrachtet. Gilt hingegen $|\alpha| > \alpha_{\max}$, so sollte die Spur nicht für die Erstellung des Basisplans verwendet werden.

Durch die Überprüfung des ersten Korridorsegments wird besonders bei falscher Startrichtung die Spur als unbrauchbar für den Basisplan klassifiziert. Tests ergaben, dass wenn das erste Korridorsegment bereits richtig gedreht wurde, auch die weiteren Korridorsegmente richtig orientiert werden konnten. Wurde hingegen bereits das erste Korridorsegment falsch ausgerichtet, so ist die Spur komplett unbrauchbar für einen Basisplan.

Korridore sind gute Startpunkte

Die Außenwandkorrektur verringert ein Abdriften nur auf Korridorsegmenten. Die Drift vom Anfang einer Spur bis zum ersten Korridorsegment bleibt zunächst erhalten. Diese *Anfangsdrift* addiert sich mit einer abweichenden Startrichtung zu einem Richtungsfehler des ersten Korridorsegments. Ist dieser zu groß, so wird zum einen das erste Korridorsegment in die falsche Richtung ausgerichtet. Zum anderen hält bei sehr großen Abweichungen die Klassifikation anhand der Außenwandkorrektur die Spur fälschlicherweise als für den Basisplan geeignet.

Bewegungsspuren, welche in Räumen starten, haben eine besonders hohe Anfangsdrift. Diese kommt durch das Umgehen von Hindernissen bis zur Türe des Raumes zu Stande.

Bewegungsspuren, welche mit einem Korridorsegment starten, haben eine Anfangsdrift von 0. Wie Voruntersuchungen im Erdgeschoss des Informatikgebäudes gezeigt haben, werden solche Spuren durch die Außenwandkorrektur meist in die richtige Richtung gedreht. Je länger der Startpunkt von dem ersten Korridorsegment entfernt ist, desto größer kann die Anfangsdrift sein. Der Abstand bis zum ersten Korridorsegment kann über mehrere Metriken gemessen werden. Im Folgenden wird die Anzahl der Schritte und die Zeit bis zum ersten Korridorsegment als Klassifikation verwendet.

Für die Klassifikation nach der Zeit wird eine maximale Zeit t_{\max} festgelegt. Ist die Zeit bis zum ersten Korridorsegment t kleiner als t_{\max} , so erfüllt die Spur die Klassifikation. Bei zu langer Zeit bis zum ersten Korridorsegment kann die Anfangsdrift durch fehlerhafte Messungen zu groß werden, so dass die Außenwandkorrektur das erste Korridorsegment eventuell falsch ausrichtet.

Die Klassifikation mit der Anzahl der Schritte funktioniert analog. Hier wird eine maximale Anzahl an Schritten s_{\max} vorgegeben. Ist die Anzahl der Schritte bis zum ersten Korridorsegment s kleiner als s_{\max} , so erfüllt die Spur diese Klassifikation. Ist sie größer, so wird die Bewegungsspur ausgefiltert.

5.4.3 Zusammenstellen des Basisplans

Ziel ist es nun, aus der Menge aller Spuren eines Gebäudes \mathcal{S} einen Basisplan zusammenzustellen. Hierzu wird eine Teilmenge $\mathcal{S}_{\text{geeignet}} \subseteq \mathcal{S}$ aller Spuren, welche von allen Klassifikationen als geeignet bewertet wurde, verwendet.

Trotz Klassifikation ist es nicht ausgeschlossen, dass einzelne Spuren aus $\mathcal{S}_{\text{geeignet}}$ durch die Außenwandkorrektur falsch ausgerichtet wurden. Daher wird nur eine Teilmenge aus $\mathcal{F}_{K(\mathcal{S}_{\text{geeignet}})}$ als Basisplan gewählt. Ein Korridor wird nur dann in den Basisplan aufgenommen, wenn seine Äquivalenzklasse genügend viele Elemente besitzt. Hierfür wird ein Parameter k_{\min} festgelegt, welcher die minimale Anzahl der Elemente eines Korridors des Basisplans beschreibt. Wurden viele Korridorsegmente nahe beieinander mit derselben Ausrichtung erfasst, so ist anzunehmen, dass sich an dieser Position ein Korridor im Gebäude befindet. Der Basisplan stellt sich somit zusammen aus

$$\mathcal{F}_{\text{Basisplan}} = \{k \in \mathcal{F}_{K(\mathcal{S}_{\text{geeignet}})} : |k| \geq k_{\min}\}.$$

Es gibt mehrere Möglichkeiten um mit Spuren umzugehen, welche zwar eine gute Klassifikation erhalten hatten, allerdings ein Korridorsegment besitzen, welches nur von sehr wenigen Spuren bestätigt wird. Eine Möglichkeit ist, solche Spuren aus dem Basisplan zunächst wieder zu entfernen. Dies hat jedoch weitere Auswirkungen, da es dadurch eventuell weitere Äquivalenzklassen der Korridorrelation geben kann, welche nur sehr wenige Korridorsegmente beinhalten. Des Weiteren kann dadurch auch eine Äquivalenzklasse in mehrere kleinere zerfallen.

Als andere Möglichkeit kann man die verbleibenden Korridorsegmente der Spur weiterhin als Bestandteil von Äquivalenzrelationen auf dem Basisplan belassen. Besonders wenn die Korridorsegmente einer Spur nur bis zu einem gewissen Zeitpunkt in Korridoren mit vielen Elementen sind, macht diese Strategie Sinn.

5.5 Iteratives Hinzufügen von Spuren auf das Korridorskelett

Auf den Basisplan sollen nun iterativ weitere Spuren hinzugefügt werden. Dabei soll verhindert werden, dass durch die Außenwandkorrektur falsch gedrehte Spuren aufgenommen werden.

Passende Spuren

Durch das Hinzufügen von Korridorsegmenten einer Bewegungsspur auf das Flurskelett ändern sich die Äquivalenzklassen. Es gibt folgende Fälle, wie sich das Hinzufügen eines Korridorsegments k auf ein Flurskelett \mathcal{F} auswirken kann:

1. Das Korridorsegment k kann keinem bestehenden Korridor zugeordnet werden. In diesem Fall gilt $||[k]| = 1$.
2. Durch die Transitivität können durch k mehrere vorher bestehende Korridore in einen Korridor zusammengefasst werden.
3. Das Korridorsegment k kann einem bestehenden Korridor zugeordnet werden und durch k werden keine zwei Korridore zusammengefasst. In diesem Fall vergrößert sich die Anzahl der Korridorsegmente in dieser Äquivalenzrelation durch das Hinzufügen von k um eins.

Ein Korridorsegment k heißt auf \mathcal{F} passend, wenn durch das Hinzufügen keine weitere Äquivalenzklasse entsteht. Fall 1 ist für passende Korridorsegmente also ausgeschlossen. Eine Spur s heißt passend auf das Flurskelett \mathcal{F} , wenn all ihre Korridorsegmente k_1, \dots, k_n auf \mathcal{F} passen.

Iteratives Hinzufügen

Das Hinzufügen der Spuren geschieht iterativ. Hierzu werden in jedem Schritt einige auf das aktuelle Flurskelett möglichst passende Spuren ausgesucht. Durch Hinzufügen der Korridorsegmente der Spuren entsteht ein neues Flurskelett. Dieses wird im nächsten iterativen Schritt verwendet. Als erstes Flurskelett wird der Basisplan verwendet. Im Folgenden bezeichnet \mathcal{F} das jeweils aktuelle Flurskelett.

Das Verfahren wird so lange fortgesetzt, bis keine verbleibende Spur den Anforderungen für das Hinzufügen genügt.

Abwägen zwischen Erkennungsrate und Risiko

Fügt man nur Spuren hinzu, für welche alle Korridorsegmente auf das Flurskelett passen, so findet man keine neuen Korridore. Andererseits könnten Korridorsegmente, welche nicht auf das Flurskelett passen, durch die Außenwandkorrektur falsch ausgerichtet sein. Man muss also zwischen der Erkennungsrate von Korridoren und dem Risiko, falsch gedrehte Korridorsegmente aufzunehmen, abwägen. Im Folgenden sollen daher Bewertungsfunktionen verwendet werden, welche angeben sollen, wie gut eine Spur auf das Flurskelett passt.

5.5.1 Bewertungsfunktionen

Bewertungsfunktionen geben an, wie gut eine Spur auf ein Flurskelett passt. Für eine Spur werden hierzu die Korridorsegmente herangezogen. Bewertungsfunktionen lassen sich also als Funktionen der Form

$$B : \{\text{Flurskelette}\} \times \{\text{Korridorsegmente}\} \rightarrow [0, 1]$$

schreiben.

Der Funktionswert einer Bewertungsfunktionen ist als eine Art Wahrscheinlichkeit zu verstehen. Dieser soll 1 sein, wenn eine Spur sehr gut auf das Flurskelett passt. Je kleiner der von der Bewertungsfunktion zurückgegebene Wert ist, desto schlechter passt die Spur. Gibt eine Bewertungsfunktion 0 zurück, so passt die Spur nicht. Wählt man also Spuren aus, welche eine niedrige Bewertung erhalten haben, so kann man eventuell mehr Korridore finden. Wählt man nur Spuren mit sehr hoher Bewertung aus, so verringert sich das Risiko, falsch ausgerichtete Spuren auf das Korridorskelett aufzunehmen.

Im Spezialfall der Bewertung einer Spur ohne Korridorsegmente sollten die Bewertungsfunktionen 0 zurückgeben, da solche Spuren im hier vorgestellten Entwurf nicht anhand der Außenwände korrigiert werden konnten.

5.5.2 Bewertung anhand der Länge auf bestehenden Korridoren

Eine Möglichkeit zur Bewertung der Spuren ergibt sich durch die Summe der Länge von passenden Korridorsegmenten. Um diesen Wert zu skalieren, wird er durch die Summe der Länge aller Korridorsegmente geteilt. Die Bewertung einer Spur s mit Korridorsegmenten k_1, \dots, k_n , welche jeweils die Länge $L(k_i)$ haben, auf dem aktuellen Flurskelett \mathcal{F} lässt sich also formal wie folgt beschreiben:

$$B_1(\mathcal{F}, \{k_1, \dots, k_n\}) = \frac{\sum_{k_i: [k_i] \in \mathcal{F}} L(k_i)}{\sum_{k_i} L(k_i)}$$

Diese Bewertung ist unabhängig von der Reihenfolge der Korridorsegmente. Die folgende Bewertungsfunktion soll zwei besondere Korridorsegmente aus der Spur hervorheben.

5.5.3 Bewertung anhand des ersten und letzten Korridorsegments

Besonders in den ersten Iterationen ist die Topologie des Flurskeletts möglicherweise noch nicht vollständig bekannt. Die vorherig beschriebene Bewertungsfunktion kann das Erkennen von neuen Korridoren, besonders bei kurzen Spuren, mitunter sehr schwer machen. Um neue Korridore bei geringem Risiko trotzdem erkennen zu können, wird eine weitere Bewertungsfunktion eingeführt.

Bei dieser wird nur das erste Korridorsegment k_1 und letzte, k_n , betrachtet. Passen beide Segmente auf das aktuelle Flurskelett, so wird einer Spur die bestmögliche Bewertung vergeben. Anderenfalls erhält die Spur die schlechteste.

Formal ist diese Bewertungsfunktion einer Bewegungsspur s mit Korridorsegmenten $\{k_1, \dots, k_n\}$ auf einem Flurskelett \mathcal{F} also wie folgt gegeben:

$$B_2(\mathcal{F}, \{k_1, \dots, k_n\}) = \begin{cases} 1, & \text{falls } [k_1] \in \mathcal{F} \text{ und } [k_n] \in \mathcal{F} \\ 0, & \text{sonst} \end{cases}$$

Wurde ein Korridorsegment k bei der Außenwandkorrektur einer Spur s falsch ausgerichtet, so gibt es meist kein auf k folgendes passendes Korridorsegment. Ist dies der Fall, so passt auch das letzte Korridorsegment nicht und B_2 gibt entsprechend 0 zurück.

5.5.4 Kombination von Bewertungsfunktionen

Es wurden zwei Bewertungsfunktionen B_1 und B_2 vorgestellt. Während B_2 nur das erste und letzte Korridorsegment für die Bewertung heranzieht, verwendet B_1 alle Korridorsegmente. Wie bereits erwähnt, eignet sich B_2 besser, um neue Korridore zu finden. Durch B_1 hingegen kann das Risiko falsch ausgerichtete Korridorsegmente aufzunehmen, besser kontrolliert werden. Ideal wäre also eine Kombination der beiden Funktionen. Hierfür wird letztendlich die Bewertungsfunktion $B(\mathcal{F}, K) = 0.75 \cdot B_1(\mathcal{F}, K) + 0.25 \cdot B_2(\mathcal{F}, K)$ verwendet.

Das iterative Hinzufügen von Bewegungsspuren wird nun wie in Algorithmus 5.1 beschreiben durchgeführt. Hierbei werden alle noch nicht hinzugefügten Spuren mit dem aktuellen Flurskelett \mathcal{F} durch B bewertet. Ist die Bewertung hinreichend gut, so wird die Spur zum Skelett hinzugefügt.

Wie nach der Erstellung des Basisplans, ist es sinnvoll Äquivalenzklassen der Korridorrelation erst ab einer gewissen Anzahl an involvierten Spuren als Korridore zu betrachten. Hierdurch wird vermieden, dass einzelne fehlerhafte Spuren anderen falschen Spuren zu guten Bewertungen verhelfen.

5.6 Reparaturschritt

Eventuell konnten einige Spuren im letzten Schritt nicht auf das Flurskelett hinzugefügt werden. Dies kann unterschiedliche Gründe haben. Im Folgenden werden einige solche Gründe mit Möglichkeiten zur Lösung beschrieben.

Algorithmus 5.1 Iteratives Hinzufügen von Bewegungsspuren auf den Basisplan

```

 $\mathcal{F} \leftarrow$  Grundriss aus Basisplan
spuren  $\leftarrow$  {Spuren, welche nicht für den Basisplan benutzt wurden}
while im letzten Schritt eine Spur hinzugefügt wurde  $\wedge$  spuren  $\neq \emptyset$  do
  hinzu  $\leftarrow \emptyset$ 
  for all  $s \in$  spuren do
    if  $0.75 \cdot B_1(\mathcal{F}, K(s)) + 0.25 \cdot B_2(\mathcal{F}, K(s)) \geq 0.8$  then
      hinzu  $\leftarrow$  hinzu  $\cup \{s\}$ 
    end if
  end for
  for all  $s \in$  hinzu do
    Füge jedes Korridorsegment von  $s$  zu  $\mathcal{F}$  hinzu
    spuren  $\leftarrow$  spuren  $\setminus \{s\}$ 
  end for
end while

```

5.6.1 Fehlertypen und Reparatur**Spuren ohne Korridorsegmente**

Spuren ohne Korridorsegmente können nach dem hier beschriebenen Entwurf nicht korrigiert werden. Alle Bewertungsfunktionen geben in diesem Spezialfall 0 zurück. Da eine Korrektur nicht möglich ist, muss auf die Verwendung solcher Spuren verzichtet werden.

Entweder hielt sich der Benutzer, welche die Spur aufgezeichnet hatte, nie für eine längere Strecke auf einem Korridor auf oder er lief dort nicht gerade. Als dritte Möglichkeit könnte auch der Sensor defekt gewesen sein. In jedem Fall ist fraglich, ob die Spuren überhaupt Zusatzinformationen über das Gebäude liefern können.

Falsche Drehung und Position durch Fehler in einem Raum

Wie Voruntersuchungen zeigten, kann bei der Erfassung von vielen kurzen, aufeinanderfolgenden Segmenten, wie es etwa bei der Bewegung in einem Raum geschieht, ein großer Fehler in der Richtung der Schrittvektoren entstehen. Dieser Umstand liegt vor, wenn eine Spur nur bis zu einem gewissen Zeitpunkt auf das konstruierte Flurskelett passt.

Man kann in diesem Fall wenigstens den ersten, passenden Teil der Spur verwenden. Hierzu schneidet man die Bewegungsspur nach dem letzten passenden Korridorsegment ab und verwirft das Ende.

Falsche Startrichtung

Besitzt eine Spur Korridorsegmente, die jedoch von Anfang an nicht mit einem konstruierten Flurskelett übereinstimmen, so kann die Startrichtung falsch bestimmt worden sein.

Als Lösung kann versucht werden, die komplette Spur in alle mögliche Richtungen, welche sich durch die Außenwandkorrektur ergeben, zu drehen. Dadurch wird jeweils das erste Korridorsegment anders ausgerichtet. Entsprechend wirkt sich die Drehung der Spur auf die anderen Korridorsegmente aus.

Für jede gedrehte Spur werden die Bewertungsfunktionen auf dem iterativ konstruierten Flurskelett ausgewertet. Es wird dann bestimmt, ob es eine signifikant bessere Drehung um den Startpunkt gibt. Eine Rotation s_r ist hierbei signifikant besser als alle anderen, wenn deren Bewertung $B(\mathcal{F}, K(s_r))$ um beispielsweise mindestens 0,2 besser als die Bewertung aller anderen Drehungen ist.

Gibt es eine signifikant bessere Ausrichtung und ist deren Bewertung hinreichend gut, so wird die Spur mit dieser Drehung zum Flurskelett hinzugefügt.

Fehler in der initialen Position

Konnte keiner der obigen Fehler detektiert werden oder konnte keine passende Startrichtung gefunden werden, um die Spur zum Flurskelett hinzuzufügen, so kann angenommen werden, dass die Startposition falsch ist.

In diesem Fall kann es helfen, die Spur analog zu [FS11] als eine Bewegungsvorschrift zu betrachten und ein Map-Matching auf das bestehende Flurskelett auszuführen. Eine solches Verfahren kann neben der Startposition auch eine richtige initiale Drehung liefern.

5.6.2 Detektierung des Fehlertyps einer Spur

Zur Behebung einer der obigen Fehler muss zunächst festgestellt werden, welcher Fehlertyp vorliegt. Dies geschieht in folgender Reihenfolge:

1. Zunächst wird überprüft, ob die Spur Korridorsegmente besitzt. Enthält sie keine, so wird sie gemäß der Lösung für den ersten Fehlertyp verworfen.
2. Besitzt die Spur Korridorsegmente und sind die ersten m Korridorsegmente passend, so hat die Spur eine falsche Drehung oder Position durch einen Fehler im Raum.
3. Passt selbst das erste Korridorsegment nicht, so wird zunächst getestet, ob es eine Ausrichtung anhand der Außenwände gibt, für welche die Spur signifikant besser ist als für alle anderen Ausrichtungen. Trifft dies zu, so kann die Spur durch diese Ausrichtung repariert werden.

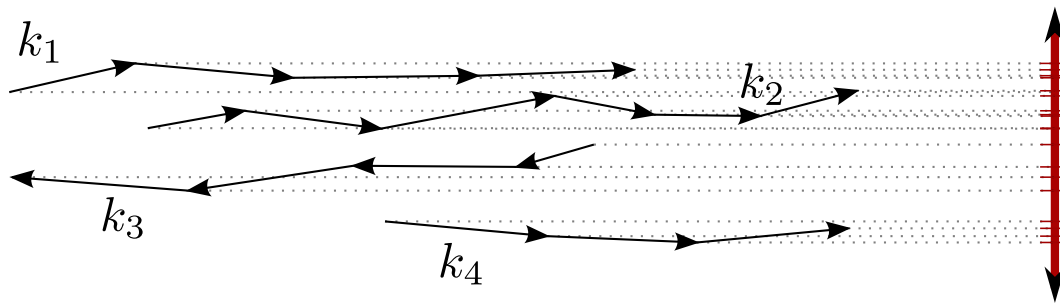


Abbildung 5.8: Projektion der absoluten Punkte p_i der Korridorsegmente zur Bestimmung der Breite eines Korridors. Alle Punkte werden auf die rote Achse parallel zur Korridorausrichtung projiziert.

4. Konnte die Spur im letzten Schritt nicht repariert werden, so kann die Startposition falsch gewählt sein. Es wird also versucht eine bessere Startposition für die Spur zu finden.

Mit dem Hinzufügen von reparierten Spuren ist das Flurskelett vollständig erstellt. Es werden keine weiteren Spuren verwendet. Als nächster Schritt wird die Geometrie der Korridore konstruiert.

5.7 Geometrische Repräsentation von Korridoren

Die geometrische Ausdehnung eines Korridors wird als Rechteck modelliert. Nach der Annahme über die Orientierung der Korridore anhand der Außenwände muss dieses Rechteck auch entsprechend entweder parallel oder orthogonal zu diesen orientiert sein.

Um die Geometrie des Rechtecks zu bestimmen werden drei Einzelwerte berechnet. Dies sind die Breite, die Mitte und die Länge des Korridors. Die Breite ist hierbei die kürzere und die Länge die längere Ausdehnung des Rechtecks.

5.7.1 Bestimmung der Breite eines Korridors

Für die Bestimmung der Breite eines Korridors werden die absoluten Punkte p_i aller Korridorsegmente der Äquivalenzrelation herangezogen. Alle absoluten Punkte werden auf eine Achse parallel zur Korridorausrichtung projiziert (siehe Abbildung 5.8). Aus den daraus gewonnenen 1D Punkten wird schließlich die Breite des Korridors berechnet.

Betrachtet man die Position der 1D Punkte relativ zueinander, so sind diese zur Korridorbreite in etwa normalverteilt. Diese Beobachtung motiviert die Verwendung des Interquartilsabstands zur Bestimmung der Breite. Der Interquartilsabstand ist der Abstand zwischen einem 0.25- und einem 0.75-Quantil. Zur Bestimmung von diesem werden also 25% der kleinsten

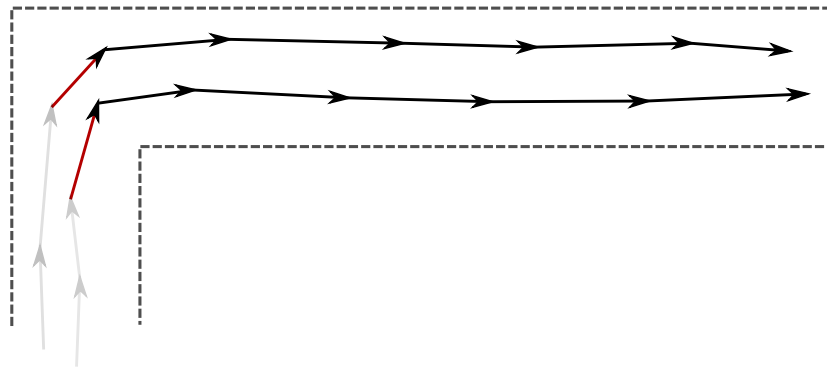


Abbildung 5.9: Beispiel zur Bestimmung der Korridormitte. Die direkt vorhergehende Schritte der zwei abgebildeten Korridorsegmente sind rot markiert.

und 25% der größten Werte abgeschnitten und dann der maximale Abstand zweier Punkte als Breite verwendet.

5.7.2 Bestimmung der Korridormitte

Eine naheliegende Strategie wäre es nun die 0.25- und 0.75-Quantile für die tatsächliche Ausdehnung des Korridors zu verwenden. Dementsprechend könnte man die Mitte als den Mittelpunkt zwischen diesen Quantilen festlegen. Bei Voruntersuchungen konnte allerdings festgestellt werden, dass die absoluten Punkte bezüglich der Korridormitte nicht normalverteilt sind. Je nach Topologie ergeben sich bei jeder Kurve gewisse Messfehler. Bei jeder Abbiegung in einen Korridor tendieren die Spuren nach der Kurve weiter in die Richtung aus der die Spur kam. Dies macht sich besonders bei Korridoren, in die nur aus einer Richtung eingebogen werden kann, bemerkbar.

Die verschobenen Spuren nach einer Abbiegung motivieren eine Ausnutzung der Topologie. Hierfür wird für jedes Korridorsegment k eines Korridors K der direkt vorhergehende Schritt s betrachtet (siehe Abbildung 5.9). Ist der Anfangspunkt des Schrittes auf der projizierten Achse kleiner als sein Ende, so wird ein Zähler $K.kleiner$ hochgezählt. Ist der Anfang des Schrittes s auf der projizierten Achse größer, so wird entsprechend der Zähler $K.größer$ erhöht. Es wird also gezählt von welcher Richtung aus in das Korridorsegment eingebogen wurde.

Durch die Anzahl der eingebogenen Spuren aus jeder Richtung kann nun ein normierter Korrekturfaktor des Korridors $K.verschiebung$ bestimmt werden:

$$K.verschiebung = \frac{K.größer - K.kleiner}{K.größer + K.kleiner}$$

Der Wert von $K.verschiebung$ liegt zwischen -1 und 1 . Ist er positiv, so soll die Mitte auf der Achse zur Beschreibung der Korridorbreite weiter nach rechts verschoben werden. Ist der Wert negativ, so entsprechend nach links.

Entsprechend zu K .verschiebung wird nun wieder ein Quantil gewählt, um die Mitte des Korridors festzulegen. Sind gleich viele Schritte von beiden Seiten in die Korridorsegmente eingegangen, wird der Median als Korridormitte verwendet. Andernfalls wird das p -Quantil mit $p = 0.5 - 0.5 \cdot K$.verschiebung gewählt. Wurde in allen Korridorsegmenten von derselben Seite aus eingetreten, so wird also entweder der kleinste oder der größte Wert gewählt.

5.7.3 Bestimmung der Länge eines Korridors

Eine einfache Möglichkeit, um die Länge eines Korridors zu bestimmen, ist es, dessen maximale Ausdehnung entlang der Längsseite des Korridors zu verwenden. Eine Schwäche der Korridorrelation ist es, dass die Länge der Korridore nicht beschränkt wird. Es kann also sehr wohl sein, dass sich falsch gedrehte Korridorsegmente in diesen befinden, welche die Korridore verlängern. Um dies zu beheben, kann man die Korridore mit Hilfe des Außenraummodells abschneiden. Dabei sollte darauf geachtet werden, dass die topologische Struktur erhalten bleibt.

Ermitteln der Länge mittels empirischer Verteilungsfunktion

Eine weitere Methode zur Bestimmung der Länge von Korridoren ergibt sich durch die empirische Verteilungsfunktion. Hierzu werden nur die Punkte der Enden der Korridorsegmente in einem Korridor betrachtet. Diese werden wieder auf eine Achse, dieses Mal parallel zur Längsausdehnung des Korridors, projiziert.

Intervalle von Korridorsegmenten Um ein Korridorsegment k eines Korridors K als Intervall betrachten zu können, wird der absolute Anfangs- und Endpunkt dieses Segments auf eine Achse parallel zur Längsausdehnung von K projiziert. Das Intervall I_k erstreckt sich nun zwischen diesen Punkten (siehe Abbildung 5.10).

Bei jedem so erhaltenen Intervall $I_k = [a, b]$ wird a der Startpunkt und b der Endpunkt des Intervalls genannt. Für die Bestimmung der Korridorlänge wird die Menge aller Startpunkte mit \mathcal{S} und die Menge aller Endpunkte mit \mathcal{E} bezeichnet.

Bestehender Anteil der Segmente eines Korridors Interessant ist nun die Betrachtung der Anzahl der bestehenden Korridorsegmente über der Längsausdehnung eines Korridors. Dazu wird für jede Raumausdehnung x die Anzahl der Intervalle, welche x enthalten, gezählt. Die relative Anzahl dieser Intervalle kann man auch als Funktion mit Hilfe der empirischen Verteilungsfunktionen der Start- und Endpunkte interpretieren.

Für einen Punkt x auf der Längsachse eines Korridors ist der Anteil der bestehenden Korridorsegmente durch $A(x) = F_{\mathcal{S}}(x) - F_{\mathcal{E}}(x)$ gegeben. Hierbei ist $F_{\mathcal{S}}(x)$ die relative Anzahl von Korridorsegmenten mit Startpunkt kleiner als x . Durch $F_{\mathcal{E}}(x)$ lässt sich der Anteil der an Position x abgeschlossenen Korridorsegmente beschreiben.

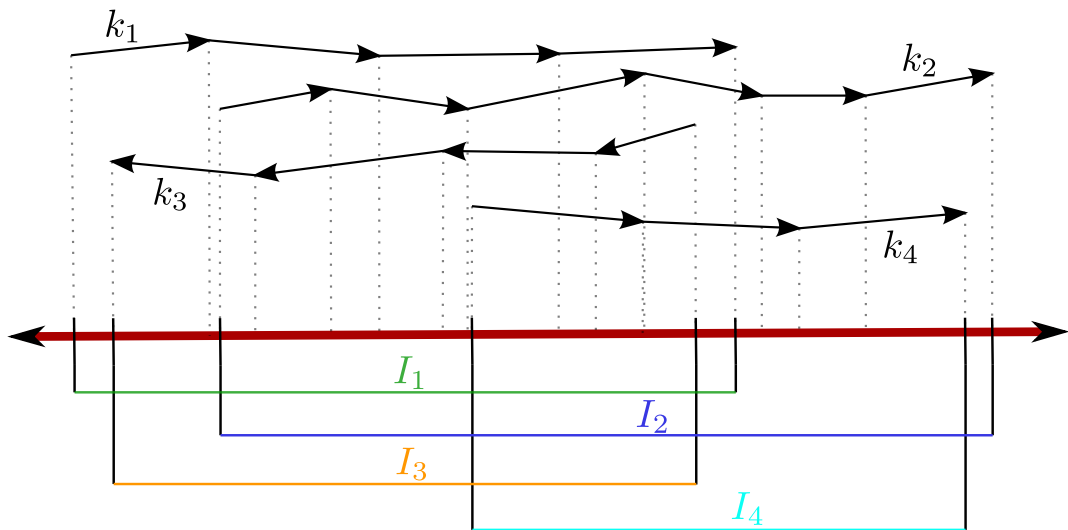


Abbildung 5.10: Projektion auf Intervalle

Durch die Konstruktion der Start- und Endpunkte muss stets $F_S(x) \geq F_E(x)$ gelten. Somit gilt auch immer $A(x) \geq 0$. Die empirische Verteilungsfunktionen $F_S(x)$ und $F_E(x)$ nehmen nur Werte zwischen 0 und 1 an. Daher gilt $A(x) \leq 1$. Dabei erreicht $A(x)$ nur dann den Wert 1, wenn zu einem Punkt alle Korridorsegmente bestehen.

Durch die Korridorrelation ergeben sich einige speziellen Eigenschaften von $A(x)$. Zwischen dem ersten Anfang x^- und dem letzten Ende x^+ eines Korridorsegments gilt für $x \in (x^-, x^+)$ stets $A(x) > 0$. Zwei Korridorsegmente müssen sich in der Längsausdehnung jeweils überschneiden um benachbart zu sein. Nach Konstruktion muss ein Korridorsegment jeweils mit einem anderen benachbart sein. Einzige Ausnahme ist wenn der Korridor nur aus einem Segment besteht. Aber auch in diesem Fall gilt für $x \in (x^-, x^+)$ stets $A(x) > 0$.

Bestimmen der Längsausdehnung mittels empirischer Verteilungsfunktion Es soll nun die Länge eines Korridors festgelegt werden. Hierzu wird der Anstieg von 0 beziehungsweise der Abstieg auf 0 der Funktion $A(x)$ betrachtet. Dieser verläuft meist monoton, so dass dafür auch die Funktionen $F_S(x)$ und $1 - F_E(x)$ verwendet werden können.

In Abbildung 5.11 sind Beispiele für diese beide Funktionen auf realen Daten gegeben. Wie man dort sieht ist ein Großteil aller Korridorsegmente nur auf einer Hälfte des Korridors erfasst worden. Tatsächlich gibt es in der Mitte des Korridors eine Abzweigung, welche viele Spuren genommen hatten. Diese Situation macht es schwierig einen Grenzwert festzulegen, ab wann die Länge eines Korridors abgeschnitten wird. Legt man diesen zum Beispiel auf 0,2 fest, so würde der Gang nur bis zur Abweichung in der Mitte erkannt werden.

Zur Ermittlung der Länge eines Korridors benötigt man also eine andere Methode. Man möchte die Information von einzelnen, nur kurz über die anderen Korridorsegmente überstehenden Segmenten nicht so sehr gewichten. Trotzdem soll diese Information in die Länge

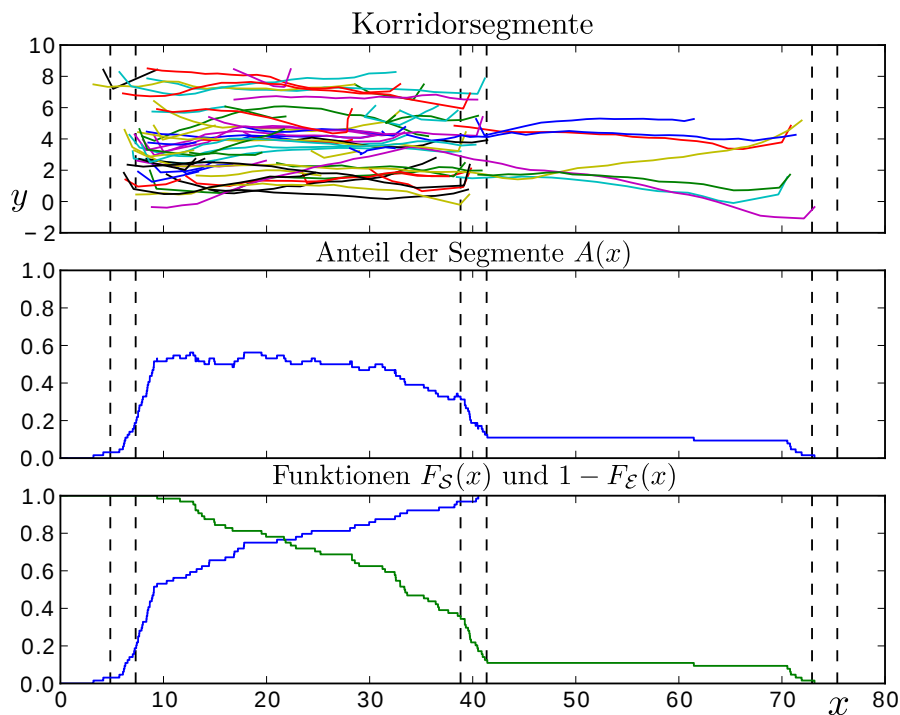


Abbildung 5.11: Darstellung von 64 Korridorsegmente des Korridors entlang der westlichen Außenwand des Informatikgebäudes. In gestrichelten Linien sind reale Abzweigungen in den Korridor eingezeichnet. Die Funktion $F_S(x)$ ist in blau und $1 - F_E(x)$ in grün dargestellt.

eingehen. Stehen mehrere Spuren über, so soll dies ein stärkeres Indiz für eine längere Ausdehnung sein.

Die Ausdehnung sollte also zum einen Abhängig von einzelnen überstehenden Segmenten, als auch der relativen Anzahl von diesen sein. Daher empfiehlt es sich den Flächeninhalt unterhalb des Anteils der bestätigenden Segmente zu betrachten. Es werden beide Enden separat betrachtet. Für den Anfang wird ein Wert x gesucht, so dass der Flächeninhalt unter $A(x)$ im Bereich $(-\infty, x]$ gleich einem vorgegebenen Wert a ist. Entsprechend wird die Ausdehnung auf der anderen Seite des Korridors verfahren. Stehen mehr Segmente über, so wird auf diese Weise weniger von deren Länge abgeschnitten.

Bestimmung des Start- und Endpunkts eines Korridors mittels Verteilungsfunktionen

Wie bereits erwähnt, sind die Enden, an denen der Anteil der bestehenden Korridorsegmente nicht 0 ist, meist monoton. Daher werden hier wieder für den Start die Verteilungsfunktion von S beziehungsweise für das Ende die Funktion $1 - F_E(x)$ verwendet.

Algorithmus 5.2 Finden des Startpunkts eines Korridors K mittels Verteilungsfunktion

```
 $a \leftarrow$  angestrebter Flächeninhalt  
 $\mathcal{S} \leftarrow$  sortiere( $\{\text{Startpunkte der Korridorsegmente von } K\}$ )  
 $x \leftarrow$  INTEGER_MIN  
integral,  $k \leftarrow 0$   
while integral  $\leq a$  do  
    anteil  $\leftarrow k/|\mathcal{S}|$   
    integral  $\leftarrow$  integral + anteil  $\cdot |\mathcal{S}[k] - x|$   
     $x \leftarrow \mathcal{S}[k]$   
     $k \leftarrow k + 1$   
end while  
 $K.start \leftarrow x - (\text{integral} - a)/\text{anteil}$ 
```

Um den Start- und Endpunkt eines Korridors K zu wählen, wird der Flächeninhalt bis zum Start- beziehungsweise Endpunkt des Korridors unter der jeweiligen Funktion mit a festgelegt. Für den Startpunkt $K.start$ ergibt sich damit folgende Gleichung

$$\int_{-\infty}^{K.start} F_{\mathcal{S}}(x) dx = a.$$

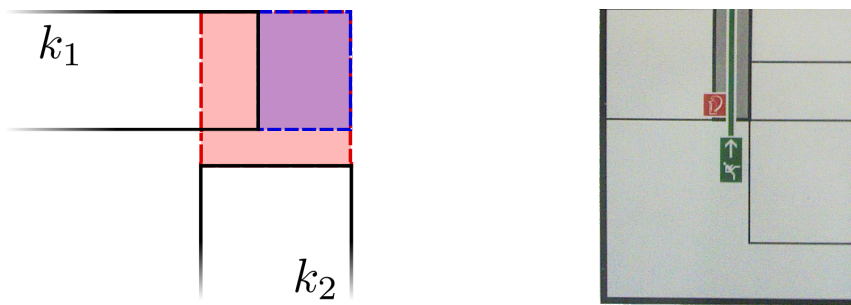
Eine ähnliche Gleichung ergibt sich für den Endpunkt des Korridors $K.ende$:

$$\int_{K.ende}^{\infty} 1 - F_{\mathcal{E}}(x) dx = a$$

Da es sich bei beiden Funktionen jeweils um eine Stufenfunktion handelt, ist es sehr leicht möglich durch gegebene Daten $K.start$ und $K.ende$ zu bestimmen. Algorithmus 5.2 zeigt wie man den Startpunkt $K.start$ eines Korridors K nach dieser Methode bestimmen kann. Das Bestimmen des Endpunkts funktioniert analog.

5.7.4 Verlängern der Korridore

Gibt es in einer Bewegungspur zwei aufeinanderfolgende Korridorsegmente, welche in unterschiedlichen Korridoren liegen, so nennt man diese Korridore *aneinanderliegend*. Befinden sich zwischen aneinanderliegenden Korridoren auf einer Spur nur wenige Schritte oder kann man auf anderer Weise sicher stellen, dass der Benutzer, welcher die Spur aufgenommen hatte, in diesem Abschnitt nur durch eine Kurve gegangen ist, so sollten sich die jeweiligen Rechtecke der Korridore überlappen. Ist dies nicht der Fall, so können diese Rechtecke entsprechend erweitert werden.



- (a) Beispiel der topologischen Erweiterung. Die Enden der Korridore k_1 und k_2 werden entsprechend erweitert. (b) Beispiel für einen Korridor, der in einer Sackgasse endet und für den keine topologische Längenkorrektur möglich ist.

Abbildung 5.12: Beispiele zur topologischen Erweiterung

Erweitert werden sollte dabei jeweils nur die Länge eines Korridors. Man setzt diese von beiden Rechtecken so lange fort, bis sich diese überschneiden. Hierbei sollte kein Korridor über die maximale Ausdehnung des anderen erweitert werden. Abbildung 5.12a zeigt ein Beispiel der topologischen Erweiterung.

Man kann diese topologische Erweiterung auch dazu verwenden, die Korridore zu verkürzen und so die komplette Länge von Korridoren zu bestimmen. Wie in Abbildung 5.12b gezeigt, ist dies allerdings nicht immer möglich. Im dort gegebenen Beispiel ist der Korridor eine Sackgasse. Es gibt am Ende des Korridors keinen weiteren angrenzenden Korridor. Damit ist auch keine topologische Erweiterung zur Korrektur der Länge möglich.

5.8 Korrektur der Spuren durch die Korridormitte

Durch die Außenwandkorrektur kann ein Abdriften über die Zeit größtenteils verhindert werden. Trotzdem wird dadurch nur die Ausrichtung von Korridorsegmenten korrigiert, nicht jedoch deren Position. Die nun bekannte Geometrie von Korridoren erlaubt auch eine Korrektur der Position von Korridorsegmenten.

Da die Länge der einzelnen Schritten bisher nicht korrigiert wurde, weichen auch die an Außenwänden korrigierten Spuren über die Zeit ab. Es ist daher nicht möglich zu entscheiden, auf welcher Seite eines Korridors ein Benutzer gegangen ist. Daher kann man annehmen, dass er den Korridor mittig durchlaufen hat. Der dadurch entstandene Fehler lässt sich auf die geschätzte Korridorbreite beschränken. Wird die Mitte des Korridors gut geschätzt, so ist der maximale Fehler der Spur auf der Achse orthogonal zur Längsausdehnung des Korridors konstant.

Durch das Gehen einer 90° Kurve kann mit dieser Methode die Position einer 2D Spur auf beiden Achsen korrigiert werden. Der entstandene Fehler ist auf beiden Achsen durch die erkannten Korridormitten beschränkt. Die sonst mit der Zeit immer weiter von der Realität

abweichenden Spuren haben somit nach dieser Korrektur einen beschränkten Fehler. Dies macht die Erkennung von Räumen möglich.

5.9 Erkennen von Räumen

Für die Erfassung von Räumen soll neben der geometrischen Ausdehnung auch die Position der Türe erkannt werden. Die geometrische Form von Räumen wird wieder wie bei Korridoren als Rechteck angenommen. Dieses Rechteck wird entsprechend dem angrenzenden Korridor ausgerichtet.

Zunächst werden Räume für jede einzelne Spur bestimmt. In einem weiteren Schritt werden die Räume aus unterschiedlichen Spuren zusammengefasst.

5.9.1 Erkennen der Räume aus der Geometrie des Flurskeletts

Für das Erkennen von Räumen in einer Spur werden Schritte, welche vermutlich nicht in Räumen gegangen wurden, nach und nach entfernt. Die Spur zerfällt dadurch in einzelne zusammenhängende Teile. Diese Spurteile werden *Abschnitte* genannt. Ziel ist es, die Abschnitte einer Spur, welche in einem Raum erfasst wurden, zu erhalten und den Rest der Spur zu entfernen.

Entfernen von Korridorsegmenten

Als erster Schritt werden lange Segmente aus den Spuren entfernt. Um sicherzustellen, dass nur Segmente aus Korridoren entfernt werden, kann eine größere Mindestlänge als bei Korridorsegmenten festgesetzt werden.

Die Außenwandkorrektur funktioniert auch mit kleineren Korridorsegmenten sehr gut, da man sich, um lange Segmente erfassen zu können, entlang von Wänden bewegen muss. Sind diese Wände parallel oder orthogonal zu Außenwänden, so funktioniert die Außenwandkorrektur auch mit kleinerer Mindestlänge. Beim Finden von Räumen muss sichergestellt werden, dass ein langes Segment, welches entfernt werden soll, tatsächlich auf einem Korridor aufgenommen wurde. Ist dies nicht der Fall, so werden möglicherweise Räume nicht erkannt. Die Mindestlänge muss also größer gewählt werden. Denkbar ist auch diese anhand der bekannten Geometrie der Korridore festzulegen.

Nicht immer wurden Korridorsegmente auch einem Korridor zugeordnet. Beinhaltet die Äquivalenzklasse des Korridorsegments zu wenig Elemente, so wurde diese nicht als Korridor betrachtet. Dieser Schritt entfernt auch lange Segmente, welche nicht Teil eines Korridors sind.

Entfernen von Schritten auf Korridoren

Sind die Segmente, welche aufgrund ihrer Länge definitiv auf einem Korridor erfasst wurden, ausgeschlossen, so gibt es noch weitere, kleinere Segmente. Diese wurden entweder in Korridoren, Kurven oder Räumen aufgenommen.

Um die Geometrie der Korridore auszunutzen, werden nun alle Schritte, welche innerhalb von Korridoren liegen, entfernt. Damit werden viele Kurven und Abschnitte, in welchen der Benutzer einem Hindernis ausweichen mussten und dabei aufgezeichnete Korridorsegmente unterbrochen wurden, ausgefiltert.

Entfernen von Kurven und geraden Abschnitten

Nach dem vorherigen Schritt können immer noch Teile von Kurven übrig geblieben sein. Um zu überprüfen, ob ein Abschnitt wirklich in einem Raum aufgenommen wurde, wird die Länge der Spur und das umschließende Rechteck des Abschnitts verwendet. Das umschließende Rechteck eines Abschnitts ist dabei analog zum umschließenden Rechteck eines Segments definiert. Man versteht darunter das Rechteck mit minimalem Flächeninhalt, welches alle absoluten Punkte des Abschnitts beinhaltet und in welchem zwei Seiten parallel zu einer Außenwand stehen.

Es wird angenommen, dass es nur eine Türe pro Raum gibt. Betritt ein Benutzer einen Raum und verlässt ihn wieder, so muss er eine Schleife in diesem gehen. Es ist also zu erwarten, dass die Länge L des Abschnitts in etwa dem Umfang U des umschließenden Rechtecks des Abschnitts entspricht.

Auf diese Weise können nur Räume, welche auch vom Benutzer betreten und verlassen wurden, erkannt werden. Räume, in welchen der Benutzer gestartet ist oder in welchen die Aufzeichnung endet, können nicht erkannt werden.

Entfernen von zu kurzen Abschnitten

Um Fehler bei der Raumerkennung auszuschließen, werden als letzter Schritt Abschnitte mit einer Länge kleiner als einem Meter entfernt. Zur Erkennung eines Raumes muss der Benutzer also ein Stück in diesen gegangen sein.

5.9.2 Bestimmung von Geometrie und Türposition

Für die Bestimmung der Geometrie von Räumen werden die absoluten Punkte der Abschnitte herangezogen. Als geometrische Form werden Rechtecke verwendet, welche zwei parallele Seiten zu dem angrenzenden Korridor haben. Dieses Rechteck wird so gewählt, dass es alle absoluten Punkte enthält, aber minimalen Flächeninhalt besitzt.

Um die Türposition zu bestimmen, wird der Punkt, an dem der Korridor in der Spur vor dem Abschnitt zuletzt überschritten wurde, gewählt.

5.10 Zusammenfügen von Räumen

Nachdem nun Räume in einzelnen Spuren erfasst wurden, werden nun wieder – ähnlich wie bei Korridoren durch die Korridorrelation – mehrere Räume aus verschiedenen Spuren zu einem zusammengefasst.

Kriterien für das Zusammenfügen

Es gibt mehrere Kriterien, um festzustellen, wann zwei Räume a, b aus verschiedenen Spuren s_a, s_b zusammengefasst werden sollen. Als notwendiges Kriterium um Räume zusammen zu fügen müssen diese an den selben Korridor angrenzen. Im Folgenden werden einige weitere Kriterien vorgestellt.

Überlappung der Räume Ein Indiz ist die Überlappung der geometrischen Ausdehnungen von a und b . Dazu werden die Rechtecke von a und b geschnitten. Ist die entstandene Fläche größer als ein Parameter A_{\min} , so werden die Räume zusammengefügt.

Abstand der Mittelpunkte der Räume Der Abstand der Mittelpunkte der Rechtecke der Räume eignet sich als weiteres Indiz. Ist dieser kleiner als d_{\max} , so werden die Räume zusammengefasst.

Abstand der Türpositionen Ein weiteres Indiz ist der Abstand der Türpositionen. Da Räume nur eine Türe besitzen, sollte der Abstand zwischen den unterschiedlich erkannten Türpositionen kleiner als τ_{\max} sein.

Das zweite Kriterium wird meist auch bei überlappenden Räumen erfüllt. Überlappen sich Räume, so ist auch der Abstand der Mittelpunkte nahe beieinander. Durch das zweite Kriterium können auch sehr nahe beieinanderliegende, aber sich nicht überlappende Räume als Kandidaten für das Zusammenfügen erkannt werden. Dies ist insbesondere bei sehr schmalen Räumen zu erwarten.

Um festzustellen, ob Räume zusammengefügt werden, sollte entweder das erste oder zweite Kriterium, in jedem Fall aber das Kriterium über den Abstand der Türposition, erfüllt sein. Auf diese Weise können nahe beieinanderliegende Räume, wie auch große sich überlappende Räume, zusammengefügt werden, sofern die Position der Türen nicht zu weit abweicht.

Algorithmus 5.3 beschreibt, wie Räume ρ_i aus verschiedenen Spuren zusammengefügt werden. Die Räume werden hierzu in Mengen r_i unterteilt. Jede Menge entspricht einem zusammengefügtem Raum. Erfüllt der Raum einer Spur ρ_1 aus einem zusammengefügtem Raum r_1 die Kriterien zum Zusammenfügen mit irgendeinem Raum aus einer Spur ρ_2 eines

Algorithmus 5.3 Zusammenfügen von Räumen aus verschiedenen Bewegungsspuren

```

 $R_{start} \leftarrow$  Menge aus den pro Spur gefundenen Räumen
 $R \leftarrow \{\{\rho\} : \rho \in R_{start}\}$ 
zusammengefügt  $\leftarrow$  True
while zusammengefügt do
  zusammengefügt  $\leftarrow$  False
  for  $r_1 \in R$  do
    for  $r_2 \in R \setminus \{r_1\}$  do
      if  $\exists(\rho_1, \rho_2) \in r_1 \times r_2 : \rho_2$  erfüllt die Kriterien zum Zusammenfügen mit  $\rho_1$  then
         $r_1 \leftarrow r_1 \cup r_2$ 
         $R \leftarrow R \setminus \{r_2\}$ 
        zusammengefügt  $\leftarrow$  True
      end if
    end for
  end for
end while
return  $R$ 

```

zusammengefassten Raumes r_2 , so werden die Mengen r_1 und r_2 als ein Raum zusammengefasst. Der Algorithmus endet, wenn keine Räume mehr zusammengefasst werden konnten.

Geometrie und Türposition von zusammengefassten Räumen

Für die Repräsentation der Geometrie der zusammengefassten Räume wird nun das umschließende Rechteck, welches entsprechend des anliegenden Korridors ausgerichtet ist und welches alle absolute Punkte der Abschnitte enthält, verwendet. Dieses kann auch aus den vorher bestimmten Ausdehnungen der Räume pro Spur bestimmt werden.

Zur Bestimmung der Türposition wird der Mittelwert der einzelnen Türen verwendet. Da diese auf einer Linie liegen, muss auch die resultierende Türposition auf der Wand zwischen Raum und Korridor liegen. Die Mittelung wirkt sich daher nur auf die Position in dieser Wand aus.

Widerspruchsfreie Räume

Eine Anforderung an das System ist, dass sich Räume nicht überlappen. Dies soll nun sichergestellt werden. Zwei Räume, welche eine überlappende geometrische Repräsentation haben, aber nicht zusammengefügt werden konnten, werden *sich gegenseitig ausschließend* genannt.

Als einfache Lösung von sich gegenseitig ausschließenden Räumen kann man den Raum, welcher aus den wenigsten Räumen aus unterschiedlichen Spuren zusammengefasst wurde,

entfernen. Man kann erwarten, dass die Wände von Räumen beispielsweise durch Möbel verstellt sind. Entsprechend gibt es eine höhere Toleranz durch die sich angrenzende Räume nicht überlappen sollten. Der mit weniger Spuren erkannte Raum ist als falsch anzunehmen.

Ausschließen von zu kleinen Räumen

Es ist unklar, ob Räume mit kleinem Flächeninhalt nicht doch falsch erkannt wurden. Bisher wurden nur in ihrer Länge sehr kurze Abschnitte aus den Spuren ausgefiltert. Dies sollte nur als erstes Kriterium für falsch erkannte Räume dienen.

Die wichtige Information, die auch aus kleineren Abschnitten gewonnen werden konnte, ist die Position der Türe. Auch durch kurze Abbiegungen in einen Raum ergibt sich so eine weitere Position, wodurch es einen weiteren Messpunkt für den Mittelpunkt gibt. Eventuell können durch eine weitere Türposition auch zwei Räume zusammengefasst werden, deren Türen sonst zu weit auseinander gewesen wären. In diesem Fall hätte der kurze Abschnitt ein weiteres Indiz für einen Raum an dieser Position verstärkt.

Konnte ein solcher kleiner Abschnitt allerdings nicht mit weiteren, größeren zusammengefasst werden, so bleibt unklar, ob er nicht fälschlicherweise als Raum angenommen wurde. Daher sollten Räume mit einem Flächeninhalt unter einem Quadratmeter verworfen werden.

6 Implementierung

Zur Evaluierung des Systems wurden alle Komponenten, welche zur Erfassung von Bewegungsspuren notwendig sind, implementiert. Dies beinhaltet neben einem Programm zur Kartenerstellung auch eine App für Android, ein Datenbankschema, dessen Schnittstellen, sowie die Implementierung der Sensoreinheit, welche auf dem Fuß getragen wird. Hierbei konnte teilweise auf bestehenden Teilen aus dem Forschungsprojekt „ComNSense“ aufgebaut werden. In diesem Kapitel wird die Implementierung all dieser Komponenten im Detail beschrieben.

6.1 Architektur

Abbildung 6.1 gibt eine Übersicht über die verschiedenen Komponenten und Technologien der Implementierung. Da es noch keine fertige Sensoreinheit für die Aufzeichnung von Bewegungsspuren gibt, wurde diese mit Hilfe eines Raspberry Pi und einer IMU zusammengestellt.

Inertialsensor Als Inertialsensor kommt eine IMU von Xsens Technologies zum Einsatz. Dieser wird mit einem Band auf dem Schuh eines Benutzers befestigt und ist per USB mit dem Raspberry Pi verbunden.

Raspberry Pi Um die Kommunikation des Sensors mit dem Smartphone zu realisieren und die Sensordaten auszuwerten, kommt ein Kleinrechner zum Einsatz. Auf diesem läuft Raspbian, eine debianartige Linux Distribution. Neben dem Sensor besitzt der Rechner ein Bluetooth-Interface, über das er mit dem Smartphone kommuniziert.

Smartphone Als Plattform für die Implementierung der App zur Aufzeichnung von Bewegungsspuren wurde Android gewählt. Als Minimalversion wird das Android SDK in Version 14 vorausgesetzt.

Datenbank-Server Die Spuren werden in einer PostgreSQL-Datenbank gespeichert.

Spurserver Der Spurserver dient als Schnittstelle zwischen dem Datenbank-Server und den mobilen Endgeräten. Er nimmt Daten von den Smartphones entgegen und stellt diese für die Kartenerstellung bereit.

Kartenerstellungsdienst Der Dienst für die Erstellung von Innenraummodellen fragt sämtliche Daten eines Gebäudes vom Spurserver ab und erstellt eine Karte nach dem vorgestellten Verfahren.

Im Folgenden wird die Implementierung der einzelnen Komponenten beschrieben.

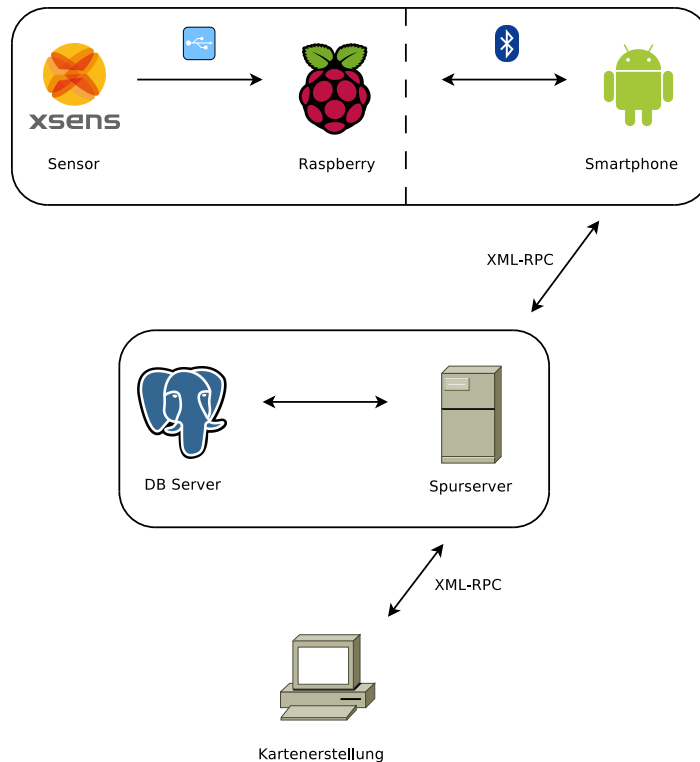


Abbildung 6.1: Übersicht über die verschiedenen Komponenten und deren Kommunikationsschnittstellen, welche für die Implementierung verwendet wurden.

6.2 Mobile Sensoreinheiten

Das Smartphone, der Inertialsensor und der Raspberry Pi bilden zusammen die mobile Sensoreinheit.

6.2.1 Die Sensoreinheit

Da noch keine geeignete Sensoreinheit kommerziell erhältlich ist, wurde als Prototyp ein Inertialsensor von Xsens Technologies, sowie ein Raspberry Pi, verwendet (siehe Abbildung 6.2). Beide Komponenten sind per USB verbunden und werden durch einen Akku mit Energie versorgt. Zur Kommunikation der Einheit mit dem Smartphone ist an den Raspberry Pi ein Bluetooth-Modul angeschlossen.

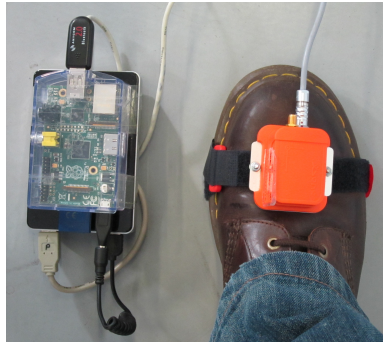


Abbildung 6.2: Die prototypische Sensoreinheit bestehend aus Inertialsensor, welcher auf dem Schuh getragen wird, und Raspberry Pi.

Auslesen der Sensordaten

Um die Daten aus dem Sensor auszulesen, wurde eine leicht modifizierte Version von `xsPi`, einem Beispielpogramm des Inertialsensors, verwendet. Dieses wurde um ZUPT erweitert. Dementsprechend gibt es bei jedem Schritt einen 2D Änderungsvektor aus.

Damit man neben den 2D Änderungsvektoren pro Schritt auch die Geschwindigkeit des Benutzer erfassen kann, wurde die Ausgabe von `xsPi` modifiziert, so dass diese nun auch die Aktuelle Systemzeit als UNIX Zeitstempel mit Genauigkeit in Millisekunden beinhaltet. Mit der Differenz der Zeitstempel und der zurückgelegten Strecke ist es so möglich, die Geschwindigkeit des Benutzers auf einem Abschnitt der Spur zu erhalten.

Kommunikation der Sensoreinheit mit dem Smartphone

Damit sich das Smartphone mit der Sensoreinheit verbinden kann, läuft auf dem Raspberry Pi das Programm `xsBTPi`. Dieses wird beim Systemstart gestartet und wartet auf eingehende Verbindungen via Bluetooth. Die Bluetooth-Verbindung ermöglicht es dem Smartphone über ein proprietäres Protokoll durch `xsBTPi` das `xsPi` Programm auf der Sensoreinheit zu starten. Die Textausgabe von `xsPi`, welche sämtliche Sensordaten beinhaltet, wird daraufhin in eine temporäre Datei gespeichert.

Tabelle 6.1 listet sämtliche Befehle auf, welche von `xsBTPi` über Bluetooth entgegen genommen werden. Neben dem Starten und Stoppen von `xsPi` gibt es Befehle, um auf den Inhalt der temporären Datei, welche die Sensordaten beinhaltet, auszulesen.

Damit der Sensor richtig funktioniert, darf nur ein Prozess von `xsPi` auf diesen zugreifen. Dies wird von `xsBTPi` sichergestellt. Des Weiteren stoppt das Script automatisch den `xsPi` Prozess, wenn das Smartphone die Verbindung trennt. Es wartet dann auf eine neue Verbindung und ermöglicht somit die Aufzeichnung von beliebig vielen Spuren.

Befehl	Beschreibung
Start xsPi	Startet xsPi
Stop xsPi	Stoppt xsPi und setzt alle Daten zurück
Read Line	Sendet die nächste ungelesene Zeile der Ausgabe von xsPi
Get File	Sendet die komplette Ausgabe vom laufenden xsPi Prozess
Exit	Schließen der Verbindung und Stoppen von xsPi

Tabelle 6.1: Die von der Sensoreinheit via Bluetooth bereitgestellten Befehle.



Abbildung 6.3: Smartphone mit der App.

6.2.2 Die ComNSenseClient App

Als Benutzerschnittstelle zur Erfassung von Bewegungsspuren wurde eine im Rahmen des Forschungsprojekts „ComNSense“ entwickelte Android App erweitert. Hierbei wurde zum einen eine Komponente, zur Zwischenspeicherung der Sensordaten auf dem Smartphone, als auch eine weitere Komponente zur Kommunikation mit dem Spurserver implementiert.

Die App ist speziell für die Evaluation im zweiten Stock des Informatikgebäudes ausgelegt. Man kann das Gebäude über dieses nicht betreten. Daher ist keine Methode, um etwa eine Eingangstüre via GPS als Startposition zu finden, implementiert. Stattdessen kann der Benutzer auf einer Karte eine Startrichtung und Position angeben.

Ruft ein Benutzer die App auf, so sieht er eine Karte des Informatikgebäudes, sowie zwei Textfelder (siehe Abbildung 6.3). Durch diese Textfelder kann er einen Benutzernamen und die Nummer des Szenarios, welches er für die Evaluation erfassen möchte, angeben. In den Einstellungen der App lassen sich verschiedene Sensoreinheiten auswählen, zu welchen sich das Smartphone verbinden kann. Damit ist jedes Smartphone in der Lage, mit jeder der zur Evaluation verwendeten Sensoreinheiten Bewegungsspuren aufzuzeichnen.

Um bei der Evaluation weitere absolute Punkte mit aufzunehmen, wurden Korrekturpunkte implementiert. Diese können vom Benutzer gesetzt werden und sollen dessen aktuelle Position wiedergeben. Korrekturpunkte sind durch die manuelle Eingabe auf dem Smartphone,

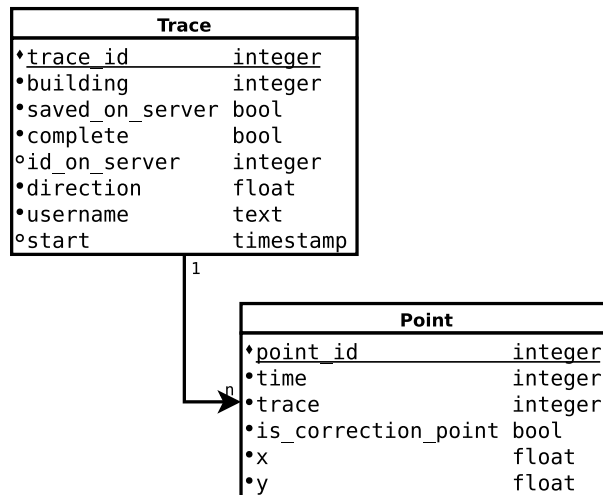


Abbildung 6.4: Datenbankschema der SQLite Datenbank in der Android App

wie alle anderen eingegebenen Punkte auch, mit einem Fehler versehen. Sie können allerdings als Imitation weiterer Positionsangaben, welche von nahen W-LAN oder Bluetooth Signalen oder einzeln verfügbaren GPS Positionen erfasst wurden, verwendet werden.

Abspeichern der Spuren auf dem Smartphone

Um eine sichere Übertragung der Bewegungsspuren auf den Spurserver zu ermöglichen, müssen diese zunächst auf dem Smartphone zwischengespeichert werden. Hierfür kommt eine von Android bereitgestellte SQLite Datenbank zum Einsatz.

Datenbankschema Auf dem Smartphone werden zwei SQLite Tabellen erstellt (siehe Abbildung 6.4). In der Tabelle Trace sind alle Informationen über eine Spur gespeichert. Die Tabelle Point hält Informationen über einen Schrittvektor oder einen Korrekturpunkt.

Für eine Bewegungsspur werden neben der Nummer des Gebäudes, dem Benutzernamen, der Startrichtung und dem Startzeitpunkt auch Informationen über den Zustand von Aufzeichnung und Übertragung zum Server gespeichert. Das Feld complete gibt an, ob die Erfassung einer Spur abgeschlossen wurde. Konnte die Spur erfolgreich auf den Spurserver übertragen werden, so wird saved_on_server auf True gesetzt. In diesem Fall können alle Daten über diese Spur auf dem Smartphone gelöscht werden. Das Feld id_on_server wird verwendet, um eventuell unterbrochene Übertragungen zum Spurserver unter der selben Identifikationsnummer wieder aufnehmen zu können.

Implementierung mittels SQLiteOpenHelper Für die Realisierung des Datenbankzugriffs wurde SQLiteOpenHelper verwendet. Hierzu erbt die für die App entwickelte Klasse

DatabaseOpenHelper von SQLiteOpenHelper. Somit stellt DatabaseOpenHelper dem Rest der App lesbare oder schreibbare Datenbankinstanzen zur Verfügung.

Um von der SQLiteOpenHelper Klasse zu erben, müssen die Methoden onCreate und onUpdate überschrieben werden. Die Methode onCreate wird aufgerufen, wenn die Datenbank noch nicht existiert. Entsprechend führt diese einen SQL Befehl aus, um die Datenbank zu erstellen.

Beim Aufruf der Elternklasse im Konstruktor muss eine Versionsnummer angegeben werden. Stimmt diese nicht mit der Versionsnummer, welche bei der Erstellung der Datenbank verwendet wurde, überein, so wird die onUpdate Methode aufgerufen, um das Datenbankschema anzupassen.

Um den Code kompakt zu halten, wurden bei Änderungen an der Datenbank jeweils der Speicher für die App und damit auch der Inhalt und die Struktur der Datenbank gelöscht. Dementsprechend steht die Version auf 1.

Übertragung der Spurdaten auf den Server

Um die Daten auf den Spurserver zu übertragen, kommt das XML-RPC Protokoll zum Einsatz. Für die Android Clients wird hierzu auf die bestehende XML-RPC Bibliothek android-xmlrpc [and12] zurückgegriffen.

Modifikation an android-xmlrpc zum Verwenden von präemptiver Authentifizierung XML-RPC baut auf HTTP auf. Das Paket android-xmlrpc verwendet für HTTP eine externe Bibliothek. Es kommt hierfür die org.apache.http Bibliothek zum Einsatz.

Um die Datensätze zu unterscheiden, wird von dem Spurserver der Benutzername der HTTP Anfrage verwendet. Für einen Webbrowser benötigt eine Authentifizierung in HTTP normalerweise zwei Anfragen. Eine erste Anfrage ohne Benutzername und Passwort wird zunächst vom Server mit einem 401 Unauthorized Fehler beantwortet. Daraufhin fragt der Browser den Benutzer nach einem Benutzernamen und Passwort, um mit diesen Daten eine weitere Anfrage an den Server zu stellen.

Die erste Anfrage wird nicht zwingend notwendig, wenn man schon im voraus davon ausgehen kann, dass eine Authentifizierung erwartet wird. Die Daten für die Authentifizierung können dann bereits *präemptiv* mit einer ersten Anfrage gesendet werden. Gerade im hier vorliegenden Fall von sehr vielen mobilen Nutzern sollte jede Möglichkeit der Reduktion der Anfragen auch ausgenutzt werden.

Die Bibliothek org.apache.http geht jedoch von dem Modell mit zwei Anfragen aus. Der präemptive Modus, also die Daten zur Authentifikation direkt zu senden, ist in dieser Bibliothek nur über Umwege möglich. Da der Spurserver allerdings nur den präemptiven Modus unterstützt und keinen Fehler bei fehlenden Authentifizierungsdaten zurücksendet, musste das android-xmlrpc Paket entsprechend angepasst werden. Hierzu wurde die XMLRPCClient

Klasse so angepasst, dass sie bei jeder Anfrage an den Server die Authentifizierungsdaten in den Kopfdaten mitsendet.

Zuverlässige Übertragung von Bewegungsspuren Bei drahtloser Kommunikation mit mobilen Endgeräten sollte man von einem möglichen Ausfall der Verbindung ausgehen. Im Fall eines Ausfalls kann man jedoch erwarten, dass die Verbindung in endlicher Zeit wieder hergestellt wird.

XML-RPC ist weder gegen Verbindungsausfälle abgesichert, noch hält es einen Status über die Verbindung. Um die Übertragung robust gegen Verbindungsausfälle zu machen, muss ein entsprechendes Protokoll durchgeführt werden.

Wird vom Benutzer die Aufzeichnung einer Spur beendet, so wird in der Datenbank im entsprechenden Eintrag der Spur das `complete` Bit gesetzt. Die Spur ist somit bereit, um auf den Spurserver hochgeladen zu werden.

Zu einigen bestimmten Zeitpunkten, wie etwa wenn die Aufzeichnung einer Spur beendet wurde, beim Start der App oder wenn der Benutzer einen „Hochladen“ Knopf betätigt, wird das Hochladen aller verbleibenden fertigen Spuren angestoßen. Hierzu wird ein neuer Prozess gestartet. Dieser läuft im Hintergrund und fragt zunächst alle Spuren ab, welche zwar vollständig (`complete`) aber noch nicht auf den Server hochgeladen sind (`saved_on_server`).

Es wird nun versucht alle solche Spuren nacheinander hochzuladen. Hierzu wird zunächst überprüft, ob für eine Spur bereits eine Identifikationsnummer des Servers bekannt ist. Diese wurde dann entsprechend vorher auf dem Smartphone im `id_on_server` Feld gespeichert. Ist dieses `NULL`, so wird zunächst eine `getNewTraceId` Anfrage an den Server gesendet, um eine neue Identifikationsnummer zu bekommen. Diese wird entsprechend in dem dafür vorgesehen Feld in der SQLite Datenbank auf dem Smartphone gespeichert.

Nun wird für jeden Schritt eine `addPoint` Anfrage an den Server gesendet. Dieser beantwortet die Anfrage mit `True`, wenn der Schritt erfolgreich in die Datenbank eingetragen werden konnte oder bereits vorher in der Datenbank existierte. `False` wird nur zurückgegeben, falls es ein Problem mit dem Datenbankserver gab.

Empfängt das Smartphone ein `False` vom Server oder wird eine `Exception` durch den XML-RPC Aufruf ausgelöst, so wird der Hintergrundprozess beendet. Erst bei erneutem Anstoßen von diesem kann die abgebrochene Übertragung fortgesetzt werden. Es wird davon ausgegangen, dass eventuell aufgetretene Probleme auf dem Server (wenn von diesem mit `False` geantwortet wurde) oder mit der Verbindung (wenn eine `Exception` von der XML-RPC Komponente ausgelöst wurde) nach einiger Zeit gelöst werden konnten und durch eine erneute Übertragung die Daten auf dem Server gespeichert werden können.

Sind alle Schritte erfolgreich hochgeladen, so wird `saved_on_server` für die entsprechende Spur gesetzt und dem Benutzer ein Hinweis (`Toast`) angezeigt.

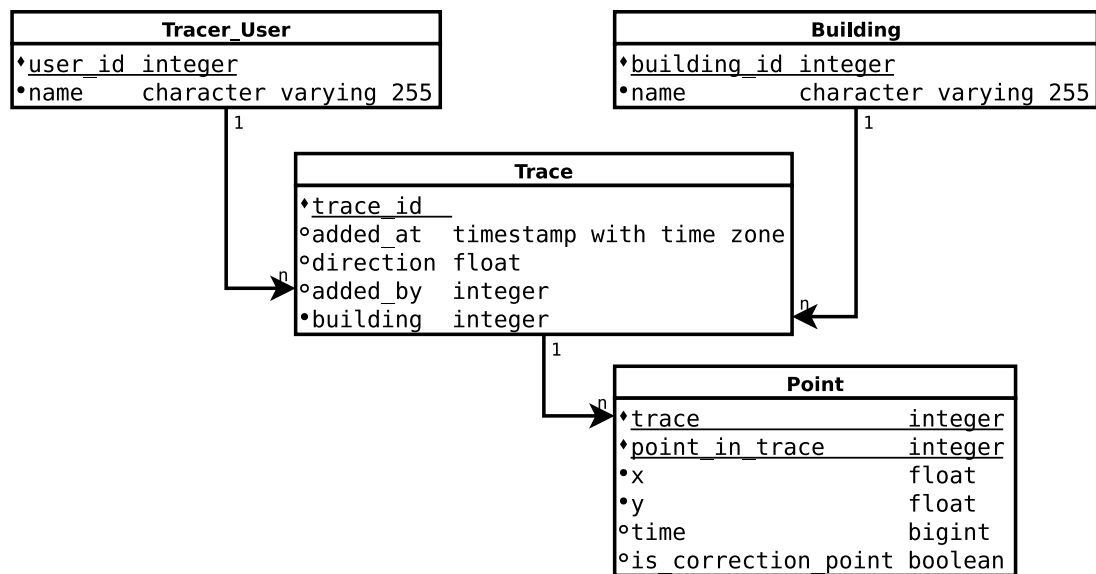


Abbildung 6.5: Datenbankmodell für die Abspeicherung der Bewegungsspuren

6.3 Datenbankschema für Bewegungsspuren

Der Datenbank-Server hält die Daten sämtlicher Bewegungsspuren. Verwendet wird dafür PostgreSQL in Version 9.2.4. Es wurde darauf geachtet keine speziellen Datentypen, wie etwa den in PostgreSQL eingebaute Typ für Punkte, zu verwenden, um das Schema portabel zu halten. Daher ist es möglich das Datenbankschema auf andere SQL basierte Datenbanken zu portieren. Im Folgenden werden die Entitäten des Datenbankmodells zum Speichern der Bewegungsspuren beschrieben (siehe Abbildung 6.5).

6.3.1 Beschreibung des Datenbankschemas

Tracer_User

Die Tabelle `Tracer_User` dient zur Unterscheidung von verschiedenen Benutzern im System. Dadurch ist es möglich die Bewegungsspuren weiter zu benennen. Durch die Unterscheidung der Benutzer können aufgezeichnete Spuren später einfacher wiedergefunden und aus den anderen Spuren herausgefiltert werden. Auch können Testdaten mit einem entsprechenden Kürzel versehen und so vom Kartenerstellungsprozess ausgeschlossen werden. Ein Benutzer wird durch einen maximal 255 Zeichen langen Namen und eine eindeutige Identifikationsnummer repräsentiert.

Building

Durch die Tabelle `Building` wird ein Unterscheiden der Gebäude möglich. Jedem Gebäude wird ein Name und eine Identifikationsnummer zugewiesen. Dadurch ist es möglich die Spuren nach Gebäude zu gruppieren und für jedes Gebäude ein Innenraummodell zu erstellen. Da die Implementierung nur 2D Daten verarbeitet, muss für die Erfassung eines realen Gebäudes pro Stockwerk eine Zeile in der `Building` Tabelle gespeichert werden.

Trace

Die Tabelle `Trace` dient zum Speichern von Bewegungsspuren. Sie verknüpft hierfür ein Gebäude, einen Benutzer und einige Punkte.

Ein `Trace` hat mehrere Punkte. Der Startpunkt ist der `Point` mit der kleinsten `point_in_trace` Nummer. Dieser ist ein Korrekturpunkt. Des Weiteren ist für eine Bewegungsspur auch die Uhrzeit sowie eine Startrichtung gespeichert. Die Startrichtung ist im Bogenmaß gegeben. Eine Startrichtung von 0 bedeutet, dass der Benutzer Richtung Norden gestartet ist.

Point

In der Tabelle `Point` werden Informationen über absolute Korrekturpunkte und relative Schrittvektoren gespeichert. Durch das Feld `is_correction_point` wird zwischen diesen unterschieden.

Schrittvektoren Für Schrittvektoren wird die Positions- und Zeitänderung zum vorherigen Schrittvektor gespeichert. Die Zeit wird als Anzahl der Millisekunden seit dem Start der Spur repräsentiert.

Korrekturpunkte Neben den Schrittvektoren werden in der Datenbank auch Korrekturpunkte erfasst. Korrekturpunkte sind absolute Punkte, für welche die Position bekannt ist. Die App ermöglicht es dem Benutzer seine aktuelle Position auf einer Karte auszuwählen. Alle diese gewählten Punkte werden als Korrekturpunkte abgespeichert. Ist der erste Punkt ein Korrekturpunkt, so gibt dieser die Startposition der Spur an.

Korrekturpunkte können auch für das Abspeichern von absoluten Positionen aus anderen Quellen verwendet werden. Dies können zum Beispiel vereinzelt GPS Positionen, die geschätzte Position durch W-LAN Signale oder die Position anhand markanter Orte sein.

Funktionsname	Beschreibung
<code>getBuilding()</code>	Gibt ein Array von IDs und Namen aller Gebäude zurück.
<code>addBuilding()</code>	Fügt ein Gebäude zur Datenbank hinzu.
<code>getNewTraceId()</code>	Erzeugt eine neue Spur und gibt dessen ID zurück.
<code>addPoint()</code>	Fügt einen 2D Schrittvektor inklusive der Zeit in Millisekunden seit dem Start der Spur hinzu.
<code>getTraces()</code>	Gibt alle Spuren in einem Gebäude zurück.

Tabelle 6.2: Die vom Spurserver per XML-RPC bereitgestellten Funktionen mit Beschreibung.

6.4 Bereitstellen und Abspeichern von Bewegungsspuren

Um die Daten einer Spur von den Smartphones entgegenzunehmen und die Daten für den Dienst zur Erstellung der Karte bereitzustellen, wurde ein separater Dienst entwickelt. Dieser ist in PHP geschrieben und stellt per XML-RPC eine entsprechende Schnittstelle für die Abfrage und Abspeicherung der Daten zur Verfügung. Pro Anfrage kommuniziert dieser Dienst mit dem Datenbank-Server, welcher die Daten entsprechend speichert. Tabelle 6.2 listet alle per XML-RPC bereitgestellten Funktionen des Spurservers mit einer kurzen Beschreibung auf.

Um unterscheiden zu können, von welchem Smartphone welche Daten aufgenommen wurden, können verschiedene Benutzernamen gewählt werden. Hierzu wird die Authentifizierung von HTTP verwendet. Diese erlaubt es den Benutzernamen direkt in der Form von `http://benutzername@server/` anzugeben.

Damit die Anfragen an den Datenbankserver effizient und sicher realisiert sind, wurden vorbereitete SQL Anfragen („prepared statements“) verwendet. Diese werden in PHP durch die eingebaute PDO Klasse bereitgestellt. Diese stellt auch die Verbindung mit der PostgreSQL Datenbank her.

6.5 Implementierung des Dienstes zur Kartenerstellung

Um aus den Bewegungsspuren eine Karte zu erstellen, wurde ein Dienst in Python implementiert. Dieser holt per XML-RPC sämtliche Daten eines Gebäudes vom Spurserver ab. Daraus errechnet er, wie in Kapitel 5 beschrieben, das Innenraummodell. Die Implementierung wurde im Rahmen des ComNSense Projekts durchgeführt. Entsprechend gibt es Schnittstellen für andere Softwarekomponenten.

6.5.1 Architektur

Das Geometrie Modul

Für die Handhabung von geometrischen Objekten wurde ein eigenes Modul entworfen. Im Folgenden werden dessen, für die Implementierung des Dienstes, wichtigsten Klassen kurz beschrieben.

Vector Ein *Vector* kann sowohl eine Position als auch eine Positionsänderung beschreiben. Die Klasse ist hierbei allgemein für den n -dimensionalen Raum implementiert, wird im Modul für die Bearbeitung von Bewegungsspuren allerdings nur im zweidimensionalen verwendet. Instanzen der Klasse *Vector* können wie gewohnt addiert und subtrahiert werden. Des Weiteren gibt es Methoden, um den eingeschlossenen Winkel zweier *Vector* Instanzen zu bestimmen.

Edge Die Klasse *Edge* beschreibt eine Strecke. Sie besitzt jeweils eine Instanz von *Vector* als Start- und Endpunkt. Es gibt unter anderem Methoden zum Rotieren (*rotate*), Verschieben (*translate*) und um die euklidische Länge zu bestimmen (*getLength*).

Polygon Die Klasse *Polygon* repräsentiert einen geschlossenen Polygonzug. Sie besteht aus mehreren Instanzen von *Vector*. Zu einem *Polygon* kann unter anderem der Flächeninhalt (*getArea*) und der Umfang (*getPerimeter*) berechnet werden. Des Weiteren lassen sich die Kanten eines Polygons bestimmen (*getEdges*).

Rectangle Zur Repräsentation von Rechtecken wurde die Klasse *Rectangle* implementiert. Diese erbt von der Klasse *Polygon*. Eine wichtige Methode, welche auf Rechtecke angewandt werden kann, ist die Bestimmung des Mittelpunktes (*getCenter*).

AxisAlignedRectangle Die Klasse *AxisAlignedRectangle* erbt von *Rectangle* und repräsentiert ein an den Hauptachsen ausgerichtetes Rechteck. Diese Klasse eignet sich somit für die spätere Repräsentation von Räumen und Korridoren, da es im Informatikgebäude nur zwei Ausrichtungen orthogonal und vertikal zu Außenwänden gibt und das Koordinatensystem auf diese ausgerichtet wird.

Weitere Funktionen auf *AxisAlignedRectangle* Instanzen ist unter anderem das für die Korridorrelation nötige Erweitern der Breite (*extendWidth*) und Höhe (*extendHeight*). Außerdem kann überprüft werden, ob sich zwei Rechtecke überlappen (*overlaps*).

Modul zur Bearbeitung der Spuren

Das Modul `TraceDerivation` wurde für die Bearbeitung der Spuren nach dem in dieser Arbeit vorgestellten Entwurf implementiert. Es besteht aus folgenden Klassen (siehe Abbildung 6.6):

Step Die `Step` Klasse hält Informationen über einen Schrittvektor. Sie erbt von der Klasse `Edge` des Geometrie Moduls. Zusätzlich zum Anfangs- und Endpunkt beinhaltet sie Informationen über die Zeit (`time`). Um bei der Korrektur von einzelnen Segmenten die Schrittlänge zu erhalten, werden Korrekturschritte eingefügt. Bei diesen ist `is_correction_step` gesetzt.

Trace Bewegungsspuren werden durch die Klasse `Trace` repräsentiert. Es stehen sämtliche Daten, welche bei der Erfassung einer Bewegungsspur angefallen sind, zur Verfügung. Ein `Trace` ist der Zusammenschluss von mehreren `Step` Instanzen. Ein `Step` fängt dabei dort an, wo der vorherige aufgehört hatte. Um dies sicherzustellen, gibt es die Methode `updatePositions`, welche die Schritte der Reihenfolge nach aneinander anpasst.

Weitere wichtige Methoden von `Trace` sind die Segmentierung (`getSegments`) und das Erhalten von absoluten Schrittunkten (`getAbsolutePoints`). Letztere Methode wird besonders zur Bestimmung der Geometrie häufig genutzt.

Segment Die Klasse `Segment` repräsentiert einen zusammenhängenden Abschnitt einer Spur. Sie erbt dazu von `Trace`. Die dem Abschnitt zugrundeliegende Bewegungsspur wird im Attribut `trace` gespeichert. Somit kennt jeder Abschnitt seine Bewegungsspur. Der häufigste Einsatz von `Segment` Instanzen ist die Segmentierung. Die Klasse wird jedoch auch zur Implementierung von Raumabschnitten herangezogen.

Zusammenhängende Abschnitte einer Spur können auch durch eine Startzeit und eine Endzeit in einer Spur gesehen werden. Entsprechend existieren Methoden um Schrittvektoren aus einem `Trace`, welche zwischen der Start- und Endzeit aufgenommen wurden, zu übernehmen (`pullSteps`) oder in diesem zu aktualisieren (`pushSteps`).

Durch die Methode `isInRelationTo` kann überprüft werden, ob zwei Korridorsegmente benachbart sind. Diese überprüft mittels `isAlignedHorizontally`, ob diese gleich ausgerichtet sind. Da es im Informatikgebäude nur zwei Ausrichtungen anhand von Außenwänden gibt, kann diese Ausrichtung durch einen booleschen Wert repräsentiert werden.

TraceBasedCorridor Die `TraceBasedCorridor` Klasse repräsentiert eine Äquivalenzklasse der Korridorrelation. Sie hält daher ein Array aus mehreren `Segment` Instanzen. Der Konstruktor von `TraceBasedCorridor` lässt sich nur mit einem `Segment` aufrufen. Dieses `Segment` wird auch das Basissegment (`baseSeg`) genannt und definiert die Ausrichtung des

6.5 Implementierung des Dienstes zur Kartenerstellung

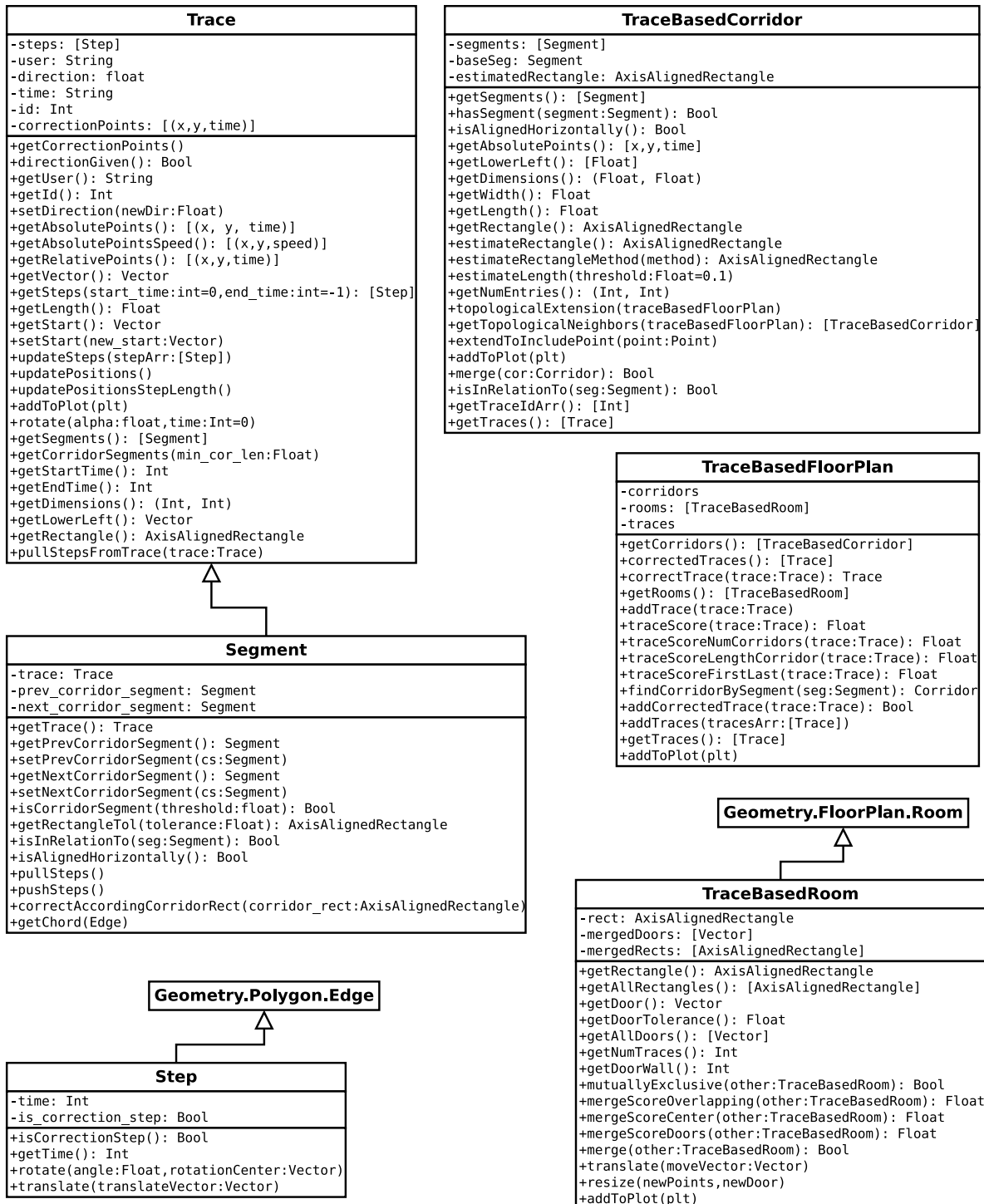


Abbildung 6.6: UML Diagramm des Kartenerstellungsdienstes

Korridors. Durch Aufruf der Klassenmethode `mergeCorridorArray` werden die Äquivalenzklassen der Korridorrelation gebildet. Das Array, welches die Segment Instanzen hält, wird entsprechend erweitert.

Weitere wichtige Methoden der Klasse `TraceBasedCorridor` sind die Bestimmung der Geometrie (`estimateRectangle`), sowie die Überprüfung, ob ein Segment zu dieser Äquivalenzklasse gehört (`isInRelationTo`).

TraceBasedRoom Um Räume zu repräsentieren wird die `TraceBasedRoom` Klasse verwendet. Dort ist zum einen die Zusammenführung von Räumen als auch das geometrische Abschneiden von Spuren mit Hilfe der Ausdehnungen von Korridoren implementiert. `TraceBasedRoom` erbt von `Room`, welche Räume in dem verwendeten Geometrie Paket repräsentiert. Implementiert sind Kriterien, welche die Nähe von zwei Räumen durch Überlappung und den Mittelpunkt der Räume bestimmen, sowie die Nähe der Türpositionen zueinander.

TraceBasedFloorPlan In `TraceBasedFloorPlan` werden Korridore und Räume gehalten. Außerdem ist dort die Bewertung, wie gut einzelne Spuren auf den Korridor passen, implementiert (`traceScoreLengthCorridor`, `traceScoreFirstLast`). Des Weiteren finden sich dort Methoden, um ein Array von `Trace` auf das Flurskelett hinzuzufügen, wie auch eine Methode um bei Spuren, von denen die initiale Drehung als falsch angenommen wird, mit Hilfe der Bewertungsfunktionen eine korrekte Ausrichtung zu finden.

Die Raumerkennung wird automatisch angestoßen, sobald `getRooms` aufgerufen wird. Die Räume werden in einem Array gespeichert, damit nicht bei jedem Aufruf von `getRooms` der komplette Prozess zur Raumfindung durchlaufen werden muss.

Eine weitere wichtige Methode dieser Klasse ist das Korrigieren mit Hilfe der geometrischen Repräsentation von Korridoren (`correctTraces`).

TracerDatabase Die Klasse `TracerDatabase` ist für die Kommunikation mit dem Spurserver verantwortlich. Per XML-RPC werden von dort alle Spuren abgefragt. Sie stellt außerdem eine Methode zur Verfügung, in der auf den letzten Cache zugegriffen werden kann. Damit ist ein Arbeiten auf dem zuletzt verwendeten Datensatz auch ohne den Spurserver möglich.

Durch die Methode `getTracesByUserRegexp` kann zur Auswahl bestimmter Szenarien oder Benutzer ein regulärer Ausdruck übergeben werden. Es werden nur Bewegungsspuren, in welchen der Benutzername auf diesen regulären Ausdruck passt, zurückgegeben.

Parameters Damit die verschiedenen Parameter in den einzelnen Methoden zentral verwaltet werden können, wurde das `Parameters` Modul entwickelt. Dieses kann sowohl fest eingestellte Parameter als auch automatisch gewählte verwenden. Die Parameter unterscheiden sich hierbei in ihrem Typ (`float` oder `int`) und der Verteilung, wie diese automatisch

gewählt werden können. Mit Ausnahme des `FixedParameter` lassen sich alle Parameter aus einer Konfigurationsdatei laden.

FixedParameter Ein `FixedParameter` ist ein fest eingestellter Parameter. Dieser ist durch die Implementierung fest vorgegeben und nur im Quelltext änderbar.

UniformIntParameter Ein `UniformIntParameter` lässt sich randomisiert aus einem Bereich wählen. Auch kann ein solcher Parameter aus einer Konfigurationsdatei ausgelesen werden.

UniformFloatParameter `UniformFloatParameter` verhalten sich wie `UniformIntParameter`, nur für Parameter, welche Fließkommazahlen verwenden.

BiasedFloatParameter Ein `BiasedFloatParameter` verwendet für die zufällige Wahl mit Wahrscheinlichkeit p einen fix angegebenen Wert. Mit Wahrscheinlichkeit $1 - p$ wird der Parameter aus einem Bereich gewählt.

6.5.2 Caching von Spuren und konstruierten Korridorskeletten

Außer zum Evaluieren des vorgestellten Verfahrens, soll das Modul `TraceDerivation` auch ein Innenraummodell für externe Anwendungen zur Verfügung stellen. Damit das Flurskelett für eine solche externe Anwendung nicht bei jedem Aufruf neu generiert werden muss, wurde ein Mechanismus zum Caching entwickelt.

Das Flurskelett wird dazu mit Hilfe von `pickle` abgespeichert. Die Speicherung erfolgt unter dem Dateinamen eines MD5 Hashes über die geordneten Parameter, welche zur Erstellung des Flurskettts verwendet wurden. Soll ein neues Innenraummodell erstellt werden, so wird zunächst überprüft, ob eine Datei mit dem per Hash gebildeten Dateinamen existiert. Ist dies der Fall, so wird das bereits erzeugte Innenraummodell zurückgegeben. Der Vorgang des Zwischenspeicherns ist nur dann möglich, wenn in der Zwischenzeit keine neuen Bewegungsspuren aufgenommen wurden.

6.5.3 Mapper

`Mapper` ist das Hauptprogramm zum Durchführen der Erstellung eines Innenraummodells. Es nimmt zahlreiche Parameter über die Kommandozeile entgegen und generiert ein Innenraummodell, welches als Bild ausgegeben wird. Wurde ein Parameter nicht übergeben, so wird dieser aus der Konfigurationsdatei `mapper.conf` ausgelesen. Tabelle 6.3 listet einige Optionen von `Mapper` auf. Daneben können auch alle Systemparameter beim Aufruf geändert werden.

Optionen	Beschreibung
-h, --help	Gibt einen Hilfetext aus.
--outfile	Angabe der Ausgabedatei. Jedes Ausgabeformat der matplotlib wird unterstützt.
--recompute	Cache ignorieren.
--trace-filter	Nur Bewegungsspuren verwenden, in welchen der Benutzername auf den regulären Ausdruck passt.
--plot-traces	Zusätzlich zu den geometrischen Ausdehnungen der Korridore verwendete Spuren einzeichnen.
--plot-trace	Wie vorheriger Parameter, aber nur Spuren, in welchen der Benutzername auf einen regulären Ausdruck passt, einzeichnen.
--plot-corrected-traces	An den Korridormitten korrigierte, verwendete Bewegungsspuren einzeichnen.
--plot-rooms	Räume in die Ausgabe einzeichnen.
--load-gt	Das reale Gebäudemodell einzeichnen.

Tabelle 6.3: Einige Optionen von Mapper

7 Evaluation

Um das System zu evaluieren wurden, mit der prototypischen Implementierung Daten aus dem Informatikgebäude erfasst. Aus diesen Daten wurden nach dem vorgestellten Verfahren Innenraummodelle erstellt. Diese wurden durch verschiedene Metriken mit dem realen Gebäudemodell verglichen. Dieses Kapitel dokumentiert zum einen das Vorgehen der Datenerfassung, zum anderen stellt es die verwendeten Metriken und deren Ergebnisse vor.

7.1 Methodologie

Grundlage für die Evaluation sollen einige erfasste Bewegungsspuren sein. Um aus diesen Innenraummodelle, welche möglichst unabhängig von einzelnen Spuren sind, zu konstruieren, wurden aus allen erfassten Bewegungsspuren einige zufällig ausgewählt. Dieses Vorgehen erlaubt die Evaluation mit einer bestimmten Anzahl von Bewegungsspuren. Es ist anzunehmen, dass mit der Zeit immer mehr Bewegungsspuren in einem Gebäude aufgenommen werden. Es soll festgestellt werden, ab wann die verschiedenen Merkmale des Innenraummodells verwendbar sind und wie gut diese auf das reale Gebäudemodell passen.

Um möglichst realistische Spuren aufzuzeichnen, wurden einige verschiedene Szenarien gewählt. Diese sollen Tätigkeiten, welche von Personen alltäglich im Gebäude durchgeführt werden, imitieren.

7.1.1 Szenarien der Datenerfassung

Alle Bewegungsspuren wurden im zweiten Stock des Informatikgebäudes, V38 der Universität Stuttgart, von vier Testpersonen gleichzeitig aufgenommen. Die Testpersonen hatten dabei Zugang zu fast allen Räumen der Abteilung Verteilte Systeme sowie zu einigen Büros der Abteilung Anwendersoftware. Abbildung 7.1 zeigt eine Übersicht über den Raumplan und die Räume, welche für diese Evaluation zur Verfügung standen. Man sieht dort auch, dass es nur zwei Ausrichtungen, orthogonal und parallel zu Außenwänden, gibt, an denen Korridorsegmente korrigiert werden können.

Als Startpunkte wurden zwei Treppen und zwei Aufzüge gewählt. Es wird angenommen, dass es eine zuverlässige Technik zum Erkennen von Treppen und Aufzügen wie in [PHF₁₁, AY₁₂] gibt und deren Position innerhalb des Gebäudes durch andere Spuren bereits bekannt ist.

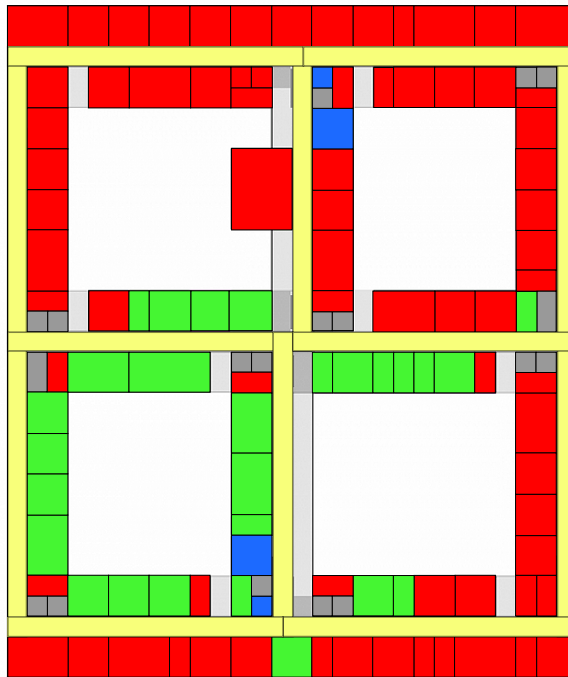


Abbildung 7.1: Übersicht über das zweite Stockwerk des Informatikgebäudes der Universität Stuttgart. Räume, zu denen bei der Evaluation Zugang bestand, sind grün eingezeichnet. Räume, zu welchen kein Zugang bestand, dagegen rot. Die blau markierten Räume sind entweder Treppenhäuser oder Aufzüge, welche als Startpunkte dienten. Korridore sind in gelb eingezeichnet und durch Linien voneinander getrennt.

Für die Evaluation sollten möglichst alle Tätigkeiten, welche im Gebäude durchgeführt werden, imitiert werden. Hierzu wurden einige verschiedene Szenarien entwickelt, welche im Folgenden vorgestellt werden.

Besucher auf Raumsuche

Im ersten Szenario sind die Testpersonen Besucher des Informatikgebäudes und suchen den Raum eines Mitarbeiters. Hierzu war lediglich ein Startpunkt und eine Raumnummer gegeben. Das System, mit welchem die Räume des Informatikgebäudes nummeriert sind, ist nicht intuitiv. Allerdings lässt sich durch dieses Nummerierungsschema feststellen, ob man sich im selben Flur wie der gesuchte Raum befindet.

Findet ein Besucher einen Raum, so klopft er an der Türe, wartet kurz und betritt schließlich den Raum. Im Raum wartet er kurz oder setzt sich hin, um dann wieder zu einem für den Benutzer beliebigen Startpunkt zurückzukehren.

Es wurden für jede Testperson zufällig 13 Startpunkte und zu besuchende Raumnummern vergeben.

Betreten des Büros

In einem weiteren Szenario befanden sich die Testpersonen auf dem Weg zur Arbeit. Damit sie auf dem schnellsten Weg in ihr Büro finden, bekamen diese neben dem Startpunkt und der Raumnummer weitere Wegpunkte. Die Start- und Zielpunkte wurden wieder zufällig gewählt.

Um möglichst viele Bewegungsspuren aufzunehmen, sollten sich die Testpersonen in ihren jeweiligen Büros kurz setzen und anschließend zu einem beliebigen Ausgangspunkt zurückkehren.

Arbeiten im Büro

Die bisher aufgenommenen Spuren waren kurz und umfassten das Betreten von jeweils nur einem Raum. Um auch längere Spuren zu erhalten, wurde das Arbeiten im Büro nachgestellt. Bei einem jeden solchen Szenario hatte jede Testperson ein festes Büro, in welches sie immer wieder zurückgekehrt ist. Es gab verschiedene Tätigkeiten, welche zwischendurch erfüllt werden sollten. Diese beinhalteten zum Beispiel den Gang zum Druckerraum, der Kaffeemaschine oder das Besuchen eines anderen Mitarbeiters in einem anderen Büro. Die Aufgaben wurden so verteilt, dass bei jeder Spur das feste Büro vier Mal verlassen werden musste.

Teilnehmen an einer Besprechung

Um eine Korrelation zwischen den Bewegungsspuren zu erfassen, wurde die Teilnahme von mehreren Benutzern an einer Besprechung imitiert. Hierzu starteten die vier Testpersonen aus nebeneinanderliegenden Büros und gingen gemeinsam in den Besprechungsraum. Dort setzten sich die Benutzer kurz und kehrten schließlich wieder in ihre Startbüros zurück.

Spuren einer Reinigungskraft und eines Wachmanns

Es wurden auch einige Spezialfälle von Spuren aufgezeichnet. Dies sind zum einen Spuren einer Reinigungskraft, welche einen Raum nach dem anderen besucht und zum anderen Spuren eines Wachmanns, welcher nie einen Raum betritt.

Eine Reinigungskraft betritt ein Büro nach dem anderen. Die Testpersonen wurden angewiesen diesen Raum komplett zu durchlaufen, um die Ausmaße des Raumes in der Spur

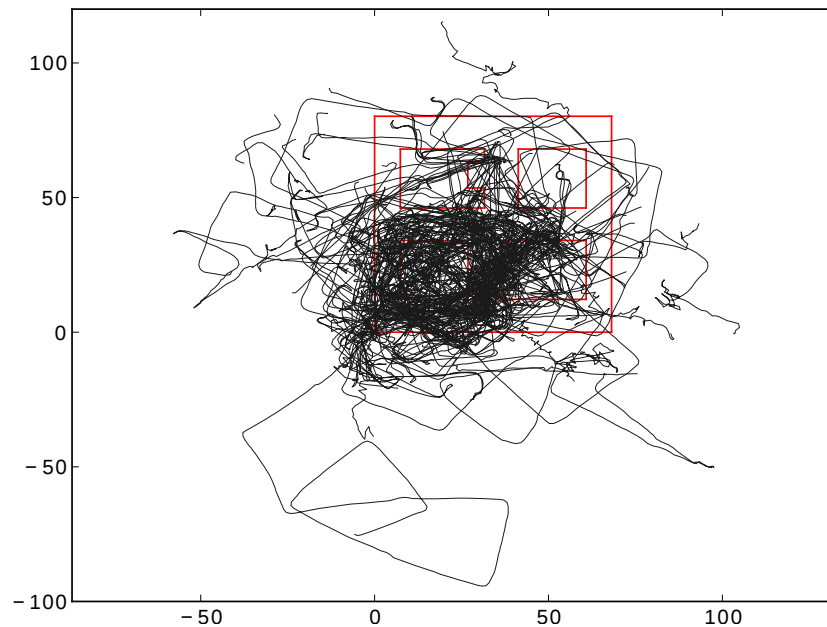


Abbildung 7.2: Die unkorrigierten, per ZUPT erfassten Bewegungsspuren. Im Hintergrund sind in rot die Umrisse des Informatikgebäudes gezeichnet.

deutlich zu machen. Durch das Besuchen eines Büros nach dem anderen kann durch Verfahren, welche den topologischen Zusammenhang von Räumen besser erkennen, eventuell eine deutliche Verbesserung der Raumerkennung erzielt werden.

Einen weiteren Spezialfall stellen die Spuren eines Wachmanns dar. Dieser bewegt sich nur auf Fluren und betritt niemals Räume. Durch solche Spuren soll die Erkennung von Fluren und deren Zusammenhang besser erfasst werden.

7.1.2 Die erfassten Bewegungsspuren

Die Bewegungsspuren wurden nicht gleichmäßig über das komplette Gebäude aufgenommen. Es bestand nur zu Räumen aus dem südlichen Teil des Gebäudes Zugang. Durch die Szenarien wurde der Nordteil des Gebäudes nicht oft besucht. Die meisten Spuren wurden im südwestlichen Quadranten aufgenommen, in dem Zugang zu den meisten Räumen bestand.

Abbildung 7.2 zeigt die per ZUPT gewonnenen Daten vor der Verarbeitung des Kartendienstes. Im Hintergrund sind zum Vergleich die Umrisse des Informatikgebäudes eingezeichnet. Es ragen sehr viele Spuren über das Gebäude hinaus.

Insgesamt wurden 227 Spuren aufgenommen. Deren Gesamtlänge beträgt 22.294 m. Damit beträgt die durchschnittliche Länge pro Spur 98 m. In allen Spuren wurden 17.659

Schrittvektoren aufgezeichnet. Die durchschnittliche Länge der Schrittvektoren beträgt somit 1,26 m.

Während der Tests traten vermehrt Probleme mit einer der vier Sensoreinheiten auf. Wie sich später herausstellte lieferte diese bei einigen Aufzeichnungen auch falsche Daten. Die Spuren aus dieser Einheit wurden trotzdem zur Evaluation herangezogen, da der Umgang mit solchen Daten eine Anforderung an das System ist.

7.1.3 Vorgehen bei der Evaluation

Um zu testen, wie sich das System mit einer variablen Anzahl an Spuren verhält, wurde bei der Evaluation aus allen erfassten Spuren eine gewisse Anzahl zufällig gewählt. Aus diesen Spuren wurde dann ein Innenraummodell erstellt. Dieses Innenraummodell wurde mit dem realen Gebäudemodell durch die Metriken verglichen. Der komplette Vorgang wurde einige Male wiederholt. Auf diese Weise ist die Evaluation nicht von einzelnen Spuren abhängig.

Für das weitere Vorgehen der Evaluation sollen zunächst die Methoden für die Segmentierung und die Korridormitte bestimmt werden. Auch soll zunächst gezeigt werden, dass sich aus allen Spuren ein Innenraummodell ableiten lässt.

7.2 Voruntersuchungen

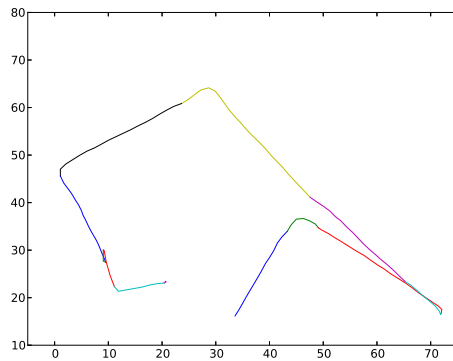
Um eine Methode zum Finden der Mitte von Korridoren sowie eine zum Segmentieren der Bewegungsspuren festzulegen, wurde eine Voruntersuchung der Daten durchgeführt. Hierbei wurden die Verfahren exemplarisch getestet. Die jeweils besseren Methoden wurden für die Evaluation verwendet. Des Weiteren wird in diesem Abschnitt als Demonstration der Machbarkeit ein Innenraummodell aus allen erfassten Spuren gezeigt.

7.2.1 Vergleich der Segmentierungsmethoden

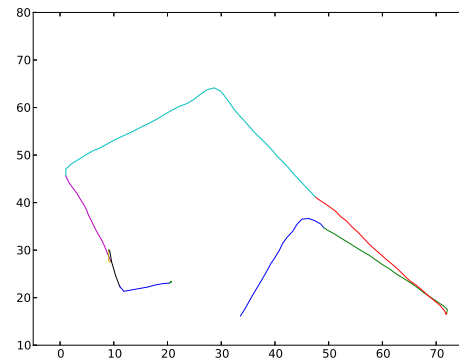
Um aus den in Abschnitt 5.3.1 vorgestellten Segmentierungsmethoden eine auszuwählen, wurden diese exemplarisch verglichen. Die Segmentierungsmethode durch die Höhe des Dreiecks hat sich hierbei als die bessere Methode herausgestellt.

Beide Methoden verwenden drei Punkte zweier aufeinanderfolgender Schritte. Eine Methode führt nach geeigneter Transformation der Punkte eine Interpolation durch und betrachtet das dadurch entstandene Polynom zweiten Grades. Je nach absolutem Wert des quadratischen Koeffizienten wird entschieden, ob die Schritte zum selben Segment gehören oder nicht. Die zweite Methode betrachtet das durch die drei Punkte entstandene Dreieck und entschied anhand dessen Höhe h , ob sich die zwei Schritte im selben Segment befinden.

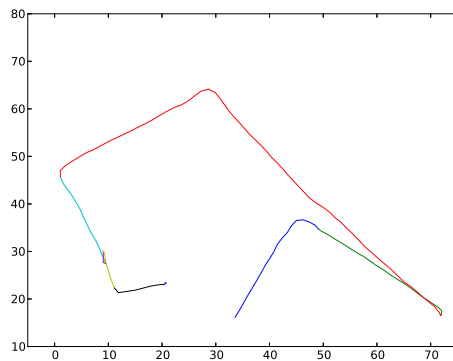
In Abbildung 7.3 ist ein exemplarischer Vergleich der beiden Verfahren an einer Bewegungspur dargestellt. Die Abbildungen 7.3a bis 7.3c zeigen die Segmentierungsmethode mittels



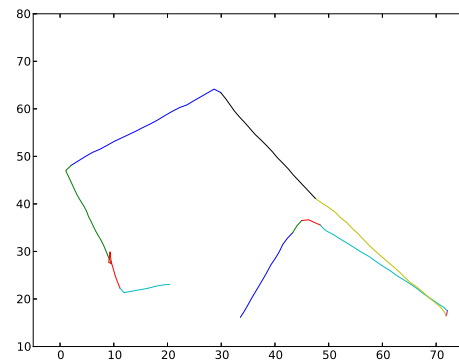
(a) Segmentierung durch Interpolation mit Schwellwert 0,06



(b) Segmentierung durch Interpolation mit Schwellwert 0,066



(c) Segmentierung durch Interpolation mit Schwellwert 0,07



(d) Segmentierung durch Höhe des Dreiecks mit Parameter $h_{\max} = 15$ cm

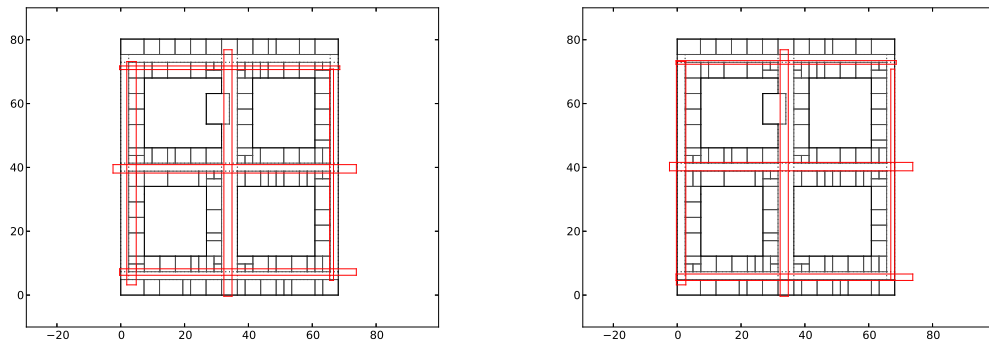
Abbildung 7.3: Exemplarischer Vergleich der Segmentierungsmethoden. Die Farben dienen zur Unterscheidung von aufeinanderfolgenden Segmenten.

Interpolation für verschiedene Schwellwerte. Bei dieser Methode wird die obere Kurve bei keinem der gewählten Schwellwerte durch ein Segment unterteilt. Lediglich im auf die Kurve folgenden Bereich kommt es zu einer Unterteilung zwischen dem Wert 0,066 und 0,06.

Die Segmentierung durch die Höhe des Dreiecks, dargestellt in Abbildung 7.3d, ist deutlich besser. Sie unterteilt alle Kurven in verschiedene Segmente. Für alle folgenden Evaluationen wurden daher die Segmente durch die Höhe des Dreiecks zweier aufeinanderfolgender Schritte bestimmt.

7.2.2 Vergleich der Verfahren zur Bestimmung der Korridormitte

Auch um eine Methode zum Bestimmen der Korridormitte festzulegen, wurde ein exemplarischer Vergleich durchgeführt. In Abschnitt 5.7.2 wurden hierfür zum einen ein Verfahren,



(a) Korridormitte bestimmt durch die Mitte des Interquartilbereichs (b) Korridormitte bestimmt durch den Korrekturfaktor

Abbildung 7.4: Exemplarischer Vergleich der Verfahren zum Bestimmen der Korridormitte

welches den Mittelpunkt des Interquartilbereichs wählt, vorgestellt. Zum anderen wurde eine Methode, welche durch die Richtung des vorherigen Schritts einen Korrekturfaktor findet und somit Abbiegungen berücksichtigt, präsentiert.

Das in Abbildung 7.4a dargestellte Korridorskelett, in welchem die Korridormitten durch die Mitte des Interquartilbereichs bestimmt wurden, weist eine deutliche Verzerrung hin zur Mitte des Gebäudes auf. Korridore, welche im Inneren des Gebäudes verlaufen, scheinen richtig erkannt worden zu sein. Jedoch sind Korridore an Außenwänden in das Gebäude hinein verschoben.

Das Korridorskelett, welches in Abbildung 7.4b, in welchem die Mitten durch den gewichteten Korrekturfaktor berechnet wurden, passt deutlich besser mit dem realen Korridorskelett überein. Lediglich die Korridore entlang der nördlichen und östlichen Außenwand stimmen in ihrer Mitte nicht mit dem realen Modell überein. Auf diesen Korridoren wurden allerdings auch nur sehr wenige Spuren aufgezeichnet.

Da sich das Verfahren mit Berücksichtigung des vorherigen Schrittes auf allen Daten besser zu eignen scheint, wurde dieses bei der Evaluation verwendet.

7.2.3 Proof of Concept

Um die Machbarkeit der Innenraummodellierung zu demonstrieren, wurde zunächst ein Innenraummodell aus allen Bewegungsspuren erzeugt (siehe Abbildungen 7.5 und 7.6). Für die Erstellung des Innenraummodells konnten 111 aus den 227 erfassten Spuren, also fast die Hälfte aller Spuren, verwendet werden. In Abbildung 7.5 ist die hohe Anzahl der verwertbaren Spuren im Südwest-Quadranten des Gebäudes zu erkennen. Tabelle 7.1 listet die bei der Evaluation verwendeten Parameter auf.

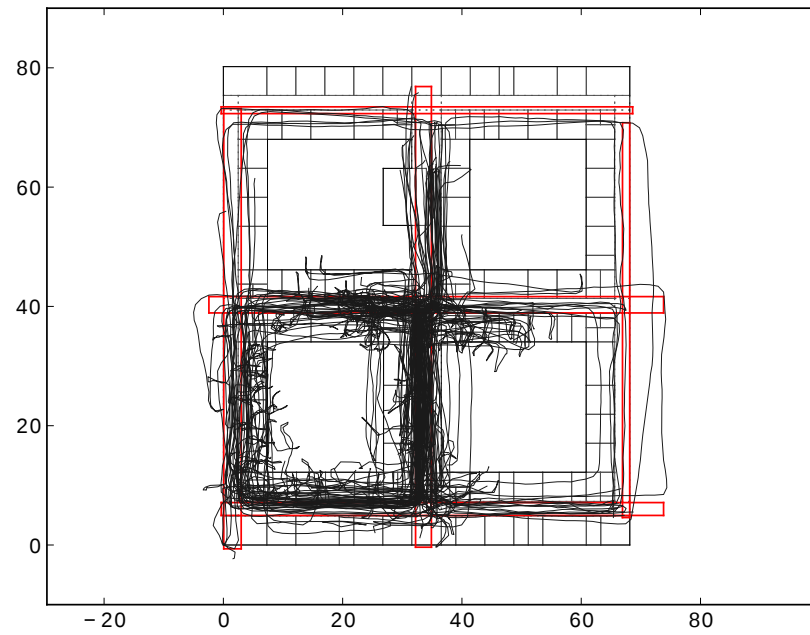


Abbildung 7.5: Aus allen Spuren abgeleitetes Flurskelett. Dargestellt sind neben den erfassten Korridoren (rot) auch die zugrunde liegenden Bewegungsspuren sowie im Hintergrund das reale Gebäudemodell.

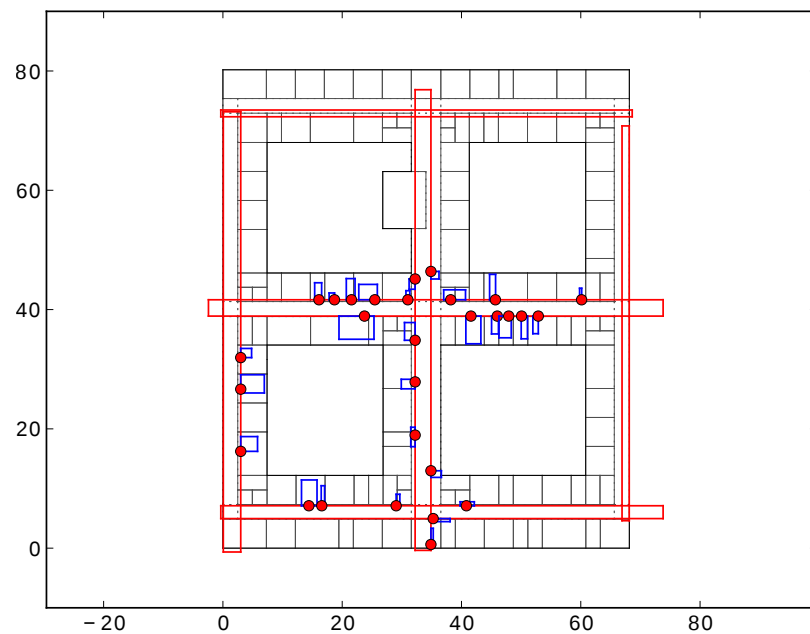


Abbildung 7.6: Aus allen Spuren abgeleitetes Innenraummodell mit Korridoren (rot), Räumen (blau) und Türen (rote Punkte). Im Hintergrund ist das reale Gebäudemodell dargestellt.

Beschreibung	Wert
Segmentierung durch Höhe, h_{\max}	0,15 m
Minimale Länge von Korridorsegmenten, L_{\min}	4 m
Maximaler Abstand in der Korridorrelation, x	0,5 m
Klassifikation für den Basisplan, t_{\max}	1 s
Klassifikation für den Basisplan, s_{\max}	1
Klassifikation für den Basisplan, α_{\max}	$\pi/8$
Klassifikation für den Basisplan, k_{\min}	4
Minimale Bewertung für das Hinzufügen auf das Skelett	0,8
Abstand der Bewertung um eine Drehung hinzuzufügen	0,2
Minimale Anzahl von Segmenten für einen Korridor	2
Länge von Korridorsegmenten für die Raumerkennung	6 m
Maximaler Abstand von Türpositionen, τ_{\max}	1 m
Maximaler Abstand der Mittelpunkte von Räumen, d_{\max}	2 m
Minimaler überlappender Flächeninhalt von Räumen, A_{\min}	0 m ²
Minimaler Flächeninhalt von Räumen	0,5 m ²

Tabelle 7.1: Werte der Parameter, welche bei der Evaluation verwendet wurden

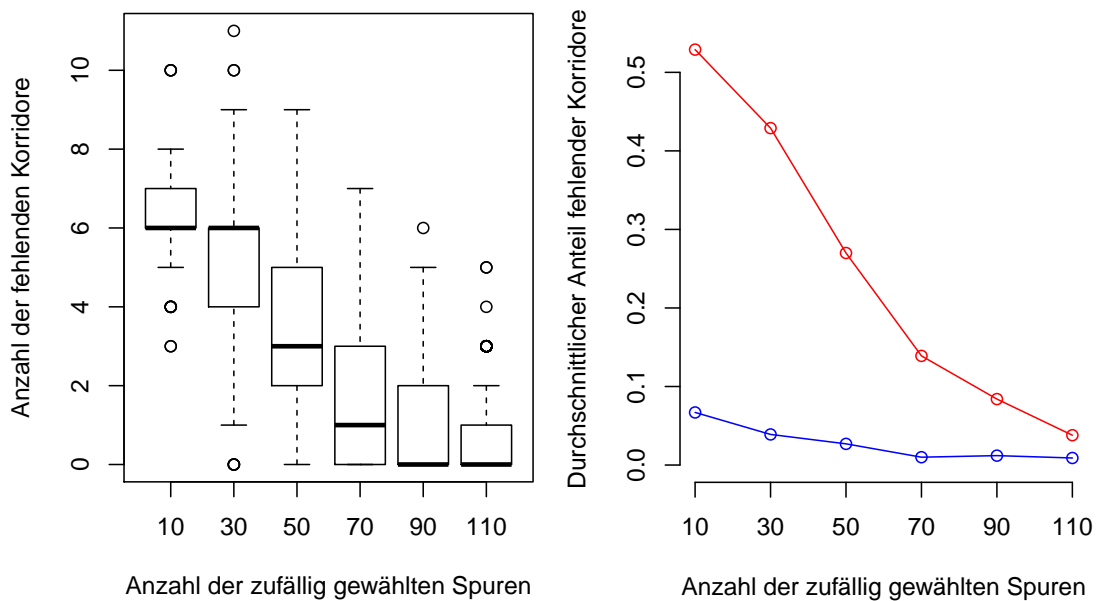
In Abbildung 7.6 ist ein Innenraummodell bestehend aus Korridoren und Räumen dargestellt. Es wurden 29 Räume detektiert. Die maximale Anzahl von Raumabschnitten, welche in einen Raum zusammengefügt wurden, ist 9. Die Anzahl der Räume, welche nur aus einem Spurabschnitt erkannt wurden, ist 17. Einige detektierte Räume gibt es im realen Gebäude nicht. So wurde beispielsweise eine Sitzecke in einem Korridor als Raum detektiert, da hier wohl ein Benutzer um diese herumgelaufen ist.

7.3 Anzahl der gefundenen Korridore

Besonders für die Korrektur und Zuordnung der Spuren ist das Erkennen der Korridore wichtig. Als erste Metrik sollen daher fehlende Korridore gezählt werden.

Um konstruierte zu realen Korridoren zuordnen zu können, wird deren Ausrichtung und Nähe verwendet. Die Nähe wird dabei über die Entfernung der Mitten der Rechtecke überprüft. Konnte einem realen Korridor auf diese Weise kein konstruierter Korridor zugeordnet werden, so fehlt dieser Korridor im erzeugten Innenraummodell.

Abbildung 7.7a zeigt die Anzahl der nicht gefundenen Korridore über der Anzahl der zufällig gewählten Spuren als Boxplots. Das Informatikgebäude besitzt insgesamt 12 Korridore (siehe Abbildung 7.1). Es zeigt sich, dass bereits bei nur 10 zufällig ausgewählten Spuren in über 64% der Fälle im erzeugten Innenraummodell maximal die Hälfte aller Korridore fehlen. Ab 90 Spuren konnten in über der Hälfte der Fälle (53%) alle Korridore erfasst werden.



(a) Anzahl der fehlenden Korridore auf dem kompletten Gebäude unter Verwendung von unterschiedlich vielen Spuren.

(b) Durchschnittlicher Anteil fehlender Korridore im süd-west Quadranten (blau) und auf dem kompletten Gebäude (rot).

Abbildung 7.7: Anzahl der gefundenen Korridore mit unterschiedlicher Anzahl von Bewegungsspuren

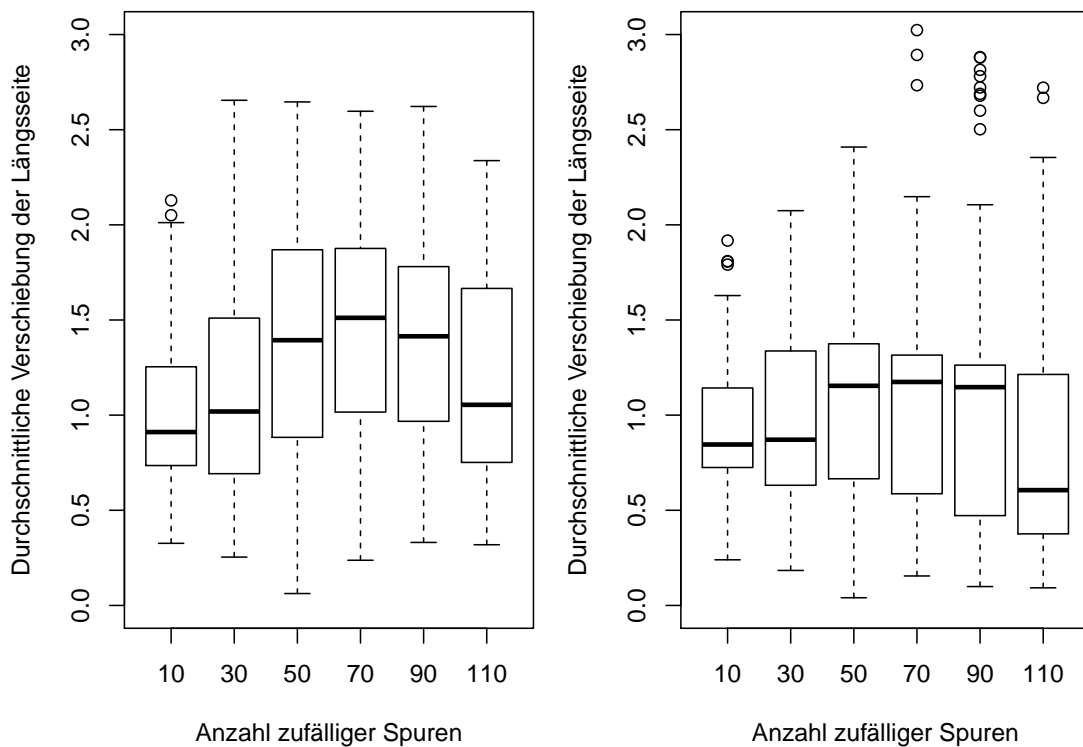
Unter der Verwendung von 110 Bewegungsspuren konnten in 74% der Fälle alle Korridore zugeordnet werden.

Abbildung 7.7b zeigt die im Durchschnitt fehlenden Korridore auf dem kompletten Gebäude und in den vier Korridoren des südwestlichen Quadranten. Im Südwest-Quadranten konnten im Durchschnitt bereits mit 70 Spuren in 99% der Durchläufe alle Korridore erkannt werden. Auf dem kompletten Gebäude fehlen hingegen selbst mit 110 Spuren 3,8% der Korridore.

Die hohe Erkennungsrate im Südwest-Quadranten ist vermutlich auf die Topologie der Korridore zurückzuführen. Es gibt nur vier Abbiegungen aus dem Quadranten heraus. Da für die Szenarien meist Aufgaben in diesem Quadranten erledigt werden mussten, können die zugehörigen Korridore sehr schnell erkannt werden.

7.4 Geometrischer Fehler von Korridoren

Einen Korridor zu finden ist leichter als dessen korrekte geometrische Ausdehnung festzustellen. Für Ersteres sind nur wenige Spuren notwendig. Für die Bestimmung der richtigen Geometrie hingegen benötigt es deutlich mehr Spuren pro Korridor, um die Messungenauigkeit auszugleichen. Die Korrelation von aus den Spuren abgeleiteten und realen Korridoren



(a) Durchschnittliche Verschiebung von allen Korridoren des Gebäudes. (b) Durchschnittliche Verschiebung von Korridoren im süd-west Quadranten.

Abbildung 7.8: Durchschnittlicher Fehler der Bestimmung der Längsseite der Korridore über der Anzahl der zufällig gewählten Bewegungsspuren.

erfolgte hierbei, wie bereits im vorherigen Abschnitt beschrieben, über die Orientierung und die Nähe der Korridore.

Abbildung 7.8 stellt die durchschnittliche Abweichung der Längsausdehnung aller Korridore pro Anzahl der zur Ableitung verwendeten Spuren dar. In Abbildung 7.8a ist der durchschnittliche Fehler aller Korridore des Gebäudes, wohingegen in Abbildung 7.8b nur der Fehler der Korridore aus dem Südwest-Quadranten dargestellt ist. Korridore im Südwest Quadranten werden dabei bis auf einige Ausreißer besser erkannt.

In beiden Abbildungen sieht man einen Anstieg der durchschnittlichen Abweichung bei einer Verwendung von 10 bis 70 Spuren. Dabei ist der Anstieg der Ungenauigkeit auf dem kompletten Gebäude deutlich größer als auf dem südwestlichen Quadranten. Dies kann durch neu erkannte Korridore, welche aus nur wenigen Spuren zusammengesetzt wurden ausgelöst werden.

Ab der Verwendung von mehr als 70 Bewegungsspuren nimmt der durchschnittliche Fehler auf allen Korridoren deutlich ab. Der Anteil der abgeleiteten Gebäudemodelle, welche einen durchschnittlichen Fehler der Bestimmung der Längsseite von unter 1,5 m haben, ist

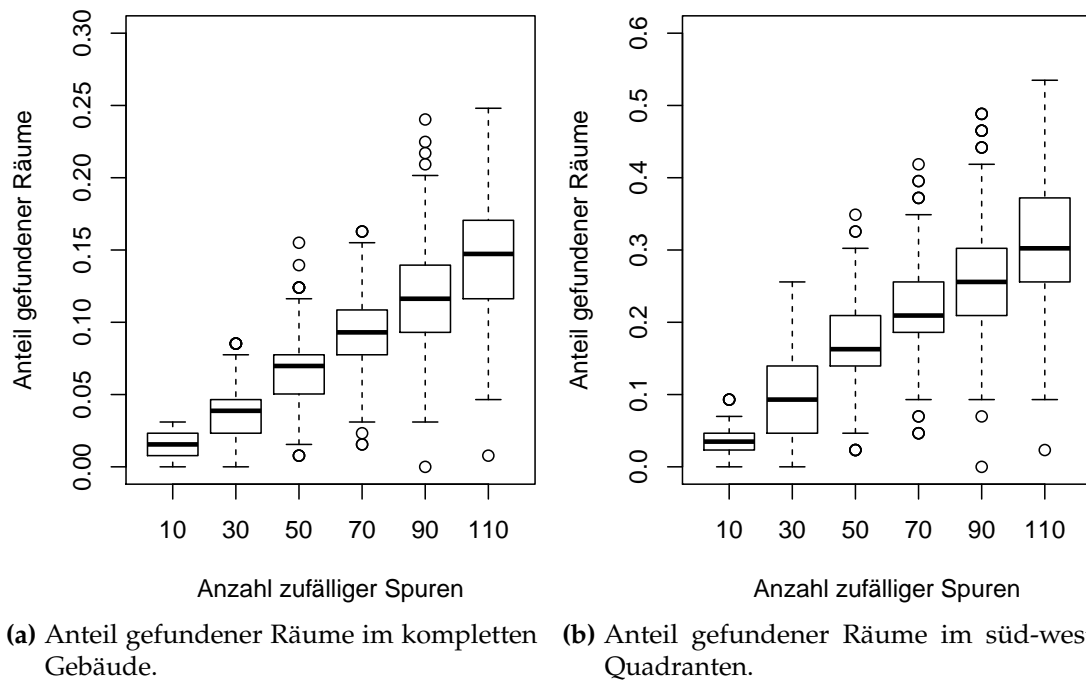


Abbildung 7.9: Anteil der gefundenen Räume über der Anzahl der zufällig gewählten Spuren auf dem Gebäude und dem Süd-West Quadranten.

bei 70 Spuren noch bei 49% und steigt über 90 Spuren mit 56% auf bis zu 67% unter der Verwendung von 110 Bewegungsspuren.

Wird die Betrachtung auf den Südwest-Quadranten eingeschränkt, so fällt dieser Abfall des durchschnittlichen Fehlers noch viel deutlicher aus. Verwendet man dort den Anteil der Innenraummodelle, welche einen kleineren durchschnittlichen Fehler als 1 m haben, so steigt dieser von 39% unter Verwendung von 70 Spuren über 41% bei 90 bis auf 55% unter Verwendung von 110 Bewegungsspuren. Bei 110 Bewegungsspuren liegt der Anteil der Innenraummodellen, welche einen durchschnittlichen Fehler auf der Längsseite von Korridoren kleiner als 1,5 m haben bei 99%. Dieser Anteil liegt unter Verwendung von 70 Spuren bereits bei 90%.

7.5 Anzahl der gefundenen Räume

Eine weitere Möglichkeit das konstruierte Gebäudemodell mit der Realität zu vergleichen ergibt sich durch die Anzahl der gefundenen Räume. Das reale Gebäudemodell weist verschiedene Raumklassen auf. Diese Raumklassen unterschieden sich in ihrer Größe. Alle Räume des Gebäudes können durch ihre Größe einer von vier Raumklassen zugeordnet werden. Um reale und konstruierte Räume einander zuordnen zu können, werden die Raumklassen an den selben Seiten von zusammengehörenden Korridoren, verglichen. Stimmen

diese überein oder konnten diese ersetzt werden, da kein anderer Raum mit der richtigen Größe verfügbar war, so gelten sie als gefunden.

Abbildung 7.9 zeigt einen Boxplot der gefundenen Räume über der Anzahl der zufällig gewählten Spuren sowohl auf dem kompletten Gebäude als auch im häufiger besuchten Südwest-Quadranten.

Auffällig ist der im Mittel lineare Zuwachs der erkannten Räume mit steigender Anzahl an verwendeten Bewegungsspuren. Der Zuwachs der gefundenen Räume stimmt mit den Resultaten von [AY12] überein. Hierfür spielt allerdings auch das Bewegungsmuster der Testpersonen eine große Rolle. In den meisten aufgenommenen Szenarien wurde nur ein Raum besucht. Sind aus mehreren zufälligen Bewegungsspuren auch mehr verwertbare dabei, so werden auch proportional mehr Räume gefunden.

Von den 43 Räumen im südwestlichen Teil des Gebäudes konnten im Durchschnitt mit 110 Spuren 13 gefunden werden, wobei jedoch nur zu 18 Räumen Zugang bestand. Um alle Räume zu finden benötigt es also deutlich mehr Spuren, als um eine genaue Geometrie der Korridore zu rekonstruieren.

8 Zusammenfassung und Ausblick

Zusammenfassung

Diese Arbeit beschäftigt sich mit dem Erfassen von Innenräumen durch Bewegungsspuren, welche von Smartphones aufgezeichnet werden. Hierzu wurde ein Inertialsensor verwendet, mit welchem durch Messen der Beschleunigung eine Bewegungspur rekonstruiert werden kann.

Die Erfassung von Innenräumen ist notwendig, da durch Innenraummodelle erst die Lokalisierung von Benutzern in Gebäuden ermöglicht wird. Steht ein solches Innenraummodell zur Verfügung, so ist auch eine Lokalisierung anhand der internen Sensoren eines Smartphones möglich. Die Kenntnis der genauen Position von Benutzern ermöglichen Location Based Services (LBS) innerhalb von Gebäuden, zum Beispiel, um einen Dienst zur Navigation in großen Gebäuden zu realisieren. Auch im Sinne des Internet der Dinge (IoT) spielt die Position des Benutzers und die ihn umgebenden Gegenstände eine Rolle. Eine Karte kann hierbei zum einen zur Visualisierung und zum anderen zur Lokalisation von diesen genutzt werden.

Die verwandten Arbeiten auf diesem Gebiet lösen das Problem entweder sehr abstrakt durch probabilistische Hilfsmittel aus der KI, so dass eine nachträgliche Erweiterung des Modells nur schwer möglich ist oder aber, sie verwenden nur die internen Sensoren eines Smartphones und fokussieren dabei sehr stark auf die Korrektur der Bewegungsspuren und nicht so sehr auf die Geometrie der abgeleiteten Räume und Korridore.

Die vorliegende Arbeit geht von der Erfassung der Bewegungsspuren mittels Inertialsensoren, welche auf dem Fuß befestigt werden aus. Die Daten werden durch Smartphones und Sensoreinheiten, welche von Benutzern in alltäglichen Situationen durch ein Gebäude getragen werden, opportunistisch erfasst und an einen zentralen Dienst übermittelt. Dieser speichert die Daten und führt von Zeit zu Zeit einen Prozess zum Erstellen des Innenraummodells durch.

Diese Arbeit konzentriert sich hauptsächlich auf die Korrektur der Spuren und das Finden einer geometrischen Repräsentation von Korridoren und Räumen. Die Korrektur und Korrelation der Bewegungsspuren findet hierbei anhand von längeren Abschnitten, in denen ein Benutzer ein gerades Stück gegangen ist, statt.

Es wurden Möglichkeiten beschrieben, wie die geometrische Breite und Länge von Korridoren aus den Daten geschätzt werden kann. Hierzu kamen zum einen der Interquartilsabstand für die Breite und eine Methode mittels empirischer Verteilungsfunktion zur Bestimmung der Länge zum Einsatz.

Um Räume in den Spuren zu identifizieren, werden die Spuren durch die bekannte Ausdehnung der Korridore nochmals korrigiert. Danach werden Teile der Spur, welche auf Korridoren oder in Kurven gegangen wurden, nacheinander entfernt.

Von Räumen soll neben der groben geometrischen Ausdehnung auch die Position der Türe geschätzt werden. Hierzu werden die Spuren wieder zueinander korreliert. Für die Position der Türe wird der Mittelwert verwendet.

Zur Untersuchung der Eigenschaften des Systems wurden alle Komponenten praktisch umgesetzt. Dies beinhaltete eine Android App, einen Datenbankserver, einen Dienst zum Abspeichern und Bereitstellen von Bewegungsspuren, sowie ein Programm für die Sensoreinheit.

Mit der Implementierung wurden über 22 Kilometer an Bewegungsspuren von vier Testpersonen an einem Tag aufgezeichnet. Dabei wurden einige Szenarien gewählt, welche typisch für die sich im Büroalltag aufhaltenden Personen sind.

Die Evaluation zeigte, dass durch den Entwurf bereits bei der Verwendung von 10 Bewegungsspuren fast die Hälfte aller Korridore zumindest stückweise erfasst werden konnten. Ab 90 Spuren wird in mehr als der Hälfte der Fälle das komplette Korridorskelett erkannt. Unter Verwendung von 110 Spuren ist die durchschnittliche Abweichung der Längsseite von Korridoren in über der Hälfte aller erzeugten Innenraummodelle kleiner als ein Meter.

Offene Fragen

Der Fokus dieser Arbeit lag auf dem grundlegenden Entwurf einer Methode, um Innenraummodelle durch Bewegungsspuren zu erfassen. Aufgrund des beschränkten Umfanges konnten nicht alle aufkommenden weitergehenden Fragestellungen behandelt werden. Abschließend sollen hier einige weitere Ideen aufgeführt werden, mit welchen durch weitere Untersuchungen, das Erstellen von Innenraummodellen durch Bewegungsspuren verbessert werden kann.

Erweiterung in 3D

Mit Inertialsensoren kann man erkennen, wann ein Benutzer eine Treppe gestiegen, eine Rolltreppe verwendet oder einen Aufzug benutzt hat [PHF11, AY12]. Damit ist eine Erweiterung in 3D denkbar, in der Spurdaten über verschiedene Stockwerke hinweg analysiert werden.

In einem solchen Modell lassen sich verschiedene architektonische Eigenschaften ausnutzen. Typischerweise sind Stockwerke ähnlich aufgebaut. Dies lässt sich zumindest bei der Bestimmung des Flurskeletts leicht ausnutzen.

Verbesserte Korrektur der Spuren durch Türpositionen

Ähnlich zu Ankerpunkten in CrowdInside könnte auch die Position der Türen zur Korrektur der Spuren verwendet werden. Diese können insbesondere für die bisher noch nicht verwendbaren Spuren herangezogen werden. Dazu muss eine Methode entwickelt werden, um zu erkennen, wann ein Benutzer sich vermutlich innerhalb eines Raumes aufgehalten hatte, ohne das geometrische Flurskelett zu verwenden. Dies wurde zum Beispiels mittels der Geschwindigkeit, mit der sich der Benutzer fortbewegt, gelöst [AY12]. Sind die Türpositionen bekannt, so kann die Spur auf bekannte Türpositionen angepasst werden.

Sammeln weiterer Merkmale, um eine genauere Positionierung zu ermöglichen

Die Korrektur anhand der Außenwände kann unter Umständen einen großen Fehler verursachen, wenn diese ein Korridorsegment falsch orientiert. Um solche Fehler zu vermeiden, wäre es hilfreich, eine weitere Methode zur groben Positionierung zur Verfügung zu haben. Auch wenn diese die Position nur auf etwa 10 m Genauigkeit schätzen kann, so könnte eine fehlerhafte Drehung eines Korridorsegments in vielen Fällen verhindert werden.

Es gibt dabei viele Möglichkeiten für das Sammeln weiterer Merkmale. Denkbar wären zum einen die Daten eines Kompasses, um die gelaufene Richtung besser abschätzen zu können. Die Verwendung von verfügbaren Bluetooth oder W-LAN Stationen für eine solche Aufgabe könnte auch zur weiteren Korrelation von Teilen der Bewegungsspuren untereinander herangezogen werden.

Literaturverzeichnis

- [and12] android-xmlrpc – very thin xmlrpc client library for Android platform, 2012. URL <https://code.google.com/p/android-xmlrpc/>. (Zitiert auf Seite 70)
- [AR12] M. Angermann, P. Robertson. Footslam: Pedestrian simultaneous localization and mapping without exteroceptive sensors—hitchhiking on human perception and cognition. *Proceedings of the IEEE*, 100(13):1840–1848, 2012. (Zitiert auf Seite 28)
- [AY12] M. Alzantot, M. Youssef. CrowdInside: Automatic Construction of Indoor Floorplans. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, S. 99–108. ACM, 2012. (Zitiert auf den Seiten 18, 25, 42, 81, 93, 96 und 97)
- [BK⁺08] S. Beauregard, M. Klepal, et al. Indoor PDR performance enhancement using minimal map information and particle filters. In *Position, Location and Navigation Symposium, 2008 IEEE/ION*, S. 141–147. IEEE, 2008. (Zitiert auf Seite 12)
- [BR11] L. Bruno, P. Robertson. Wislam: Improving footslam with wifi. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, S. 1–10. IEEE, 2011. (Zitiert auf Seite 29)
- [CEL⁺08] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, G.-S. Ahn. The rise of people-centric sensing. *Internet Computing, IEEE*, 12(4):12–21, 2008. (Zitiert auf Seite 23)
- [Ede10] H. Edelsbrunner. Alpha shapes — a survey. *Tessellations in the Sciences*, 2010. (Zitiert auf Seite 27)
- [Fox05] E. Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *Computer Graphics and Applications, IEEE*, 25(6):38–46, 2005. (Zitiert auf den Seiten 12, 16 und 17)
- [FS11] S. Funke, S. Storandt. Path shapes: an alternative method for map matching and fully autonomous self-localization. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, S. 319–328. ACM, 2011. (Zitiert auf Seite 52)
- [FTSH13] C. Fischer, P. Talkad Sukumar, M. Hazas. Tutorial: Implementing a Pedestrian Tracker Using Inertial Sensors. *Pervasive Computing, IEEE*, 12(2):17–27, 2013. (Zitiert auf Seite 17)

- [GPRH12] M. Garcia Puyol, P. Robertson, O. Heirich. Complexity-reduced FootSLAM for indoor pedestrian navigation. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, S. 1–10. IEEE, 2012. (Zitiert auf Seite 28)
- [Har13] M. Harris. How new indoor navigation systems will protect emergency responders, 2013. URL <http://spectrum.ieee.org/static/how-new-indoor-navigation-systems-will-protect-emergency-responders>. (Zitiert auf Seite 11)
- [JSZ⁺12] A. R. Jiménez, F. Seco, F. Zampella, J. C. Prieto, J. Guevara. Improved heuristic drift elimination with magnetically-aided dominant directions (MiHDE) for pedestrian navigation in complex buildings. *Journal of Location Based Services*, 6(3):186–210, 2012. (Zitiert auf den Seiten 18 und 19)
- [LBD⁺05] B. Liu, P. Brass, O. Dousse, P. Nain, D. Towsley. Mobility improves coverage of sensor networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, S. 300–308. ACM, 2005. (Zitiert auf Seite 22)
- [LDBL07] H. Liu, H. Darabi, P. Banerjee, J. Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(6):1067–1080, 2007. (Zitiert auf den Seiten 11 und 15)
- [LEM⁺08] N. D. Lane, S. B. Eisenman, M. Musolesi, E. Miluzzo, A. T. Campbell. Urban sensing systems: opportunistic or participatory? In *Proceedings of the 9th workshop on Mobile computing systems and applications*, S. 11–16. 2008. (Zitiert auf Seite 23)
- [MSN⁺09] N. Maisonneuve, M. Stevens, M. E. Niessen, P. Hanappe, L. Steels. Citizen noise pollution monitoring. In *Proceedings of the 10th Annual International Conference on Digital Government Research: Social Networks: Making Connections between Citizens, Data and Government*, S. 96–103. Digital Government Society of North America, 2009. (Zitiert auf Seite 23)
- [NSHH12] J.-O. Nilsson, I. Skog, P. Handel, K. Hari. Foot-mounted INS for everybody—an open-source embedded implementation. In *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, S. 140–145. IEEE, 2012. (Zitiert auf den Seiten 12, 16 und 17)
- [PDR11] D. Philipp, F. Durr, K. Rothermel. A sensor network abstraction for flexible public sensing systems. In *Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on*, S. 460–469. IEEE, 2011. (Zitiert auf Seite 23)
- [PHF11] M. Peter, N. Haala, D. Fritsch. Using photographed evacuation plans to support MEMS IMU navigation. In *Proceedings of the 2011 International Conference on Indoor Positioning and Indoor Navigation (IPIN2011), Guimaraes, Portugal*. 2011. (Zitiert auf den Seiten 30, 81 und 96)
- [PRA13] M. G. Puyol, P. Robertson, M. Angermann. Managing Large-Scale Mapping and Localization for Pedestrians Using Inertial Sensors. In *International Workshop on Managing Ubiquitous Communications and Services*. IEEE, 2013. (Zitiert auf Seite 28)

- [PSA⁺13] D. Philipp, J. Stachowiak, P. Alt, F. Dürr, K. Rothermel. DrOPS: Model-Driven Optimization for Public Sensing Systems. In *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Band 18, S. 22. 2013. (Zitiert auf Seite 23)
- [PSDR13] D. Philipp, J. Stachowiak, F. Dürr, K. Rothermel. Model-Driven Public Sensing in Sparse Networks. In *Proceedings of the 10th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, LNCS*, S. 1–12. Springer, 2013. (Zitiert auf Seite 23)
- [RAK10] P. Robertson, M. Angermann, M. Khider. Improving Simultaneous Localization and Mapping for pedestrian navigation and automatic mapping of buildings by using online human-based feature labeling. In *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION*, S. 365–374. 2010. (Zitiert auf Seite 30)
- [RPA11] P. Robertson, M. G. Puyol, M. Angermann. Collaborative Pedestrian Mapping of Buildings Using Inertial Sensors and FootSLAM. In *ION GNSS*. 2011. (Zitiert auf Seite 30)
- [SMo8] A. Steed, R. Milton. Using tracked mobile sensors to make maps of environmental effects. *Personal and ubiquitous computing*, 12(4):331–342, 2008. (Zitiert auf Seite 23)
- [WHo8] O. Woodman, R. Harle. Pedestrian localisation for indoor environments. In *Proceedings of the 10th international conference on Ubiquitous computing*, S. 114–123. 2008. (Zitiert auf Seite 12)
- [Wik13a] Wikipedia. Kalman filter — Wikipedia, The Free Encyclopedia, 2013. URL http://en.wikipedia.org/w/index.php?title=Kalman_filter&oldid=578612274. [Online; Stand 28. Oktober 2013]. (Zitiert auf Seite 17)
- [Wik13b] Wikipedia. Map — Wikipedia, The Free Encyclopedia, 2013. URL <http://en.wikipedia.org/w/index.php?title=Map&oldid=578924436>. [Online; Stand 28. Oktober 2013]. (Zitiert auf Seite 11)

Alle URLs wurden zuletzt am 10. 11. 2013 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift