

Umgebungsmodelle und Navigationsdaten für ortsbezogene Dienste in Gebäuden

Dissertation

an der
Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München



vorgelegt von

Peter Ruppel

Tag der Einreichung: 28. Juni 2011

Tag des Rigorosums: 16. Dezember 2011

1. Berichterstatter: Prof. Dr. Claudia Linnhoff-Popien, LMU München
2. Berichterstatter: Prof. Dr. Christian Becker, Universität Mannheim

Zusammenfassung

Ortsbezogene Dienste in Gebäuden (*Indoor Location-based Services, I-LBS*) sammeln, verarbeiten und stellen Informationen bereit, die in Abhängigkeit der Aufenthaltsorte ihrer Zielobjekte sowie auf der Grundlage eines Umgebungsmodells des jeweiligen Gebäudes berechnet werden. Die Anwendungsbereiche von I-LBS reichen dabei von personenbezogenen Diensten wie z.B. Navigation, ortsbezogenen Benachrichtigungen und Ressourcen-Suche über organisationsbezogene Dienste zur Steuerung und Optimierung von Arbeitsprozessen und Ressourcen bis hin zu Anwendungen, die personen- und organisationsübergreifend Informationen über Zielobjekte und deren jeweilige Umgebungen austauschen. Für die Erzeugung und Nutzung von Umgebungsmodellen ergeben sich mehrere Herausforderungen: Bei der automatischen Erzeugung eines Modells müssen die Eigenschaften eines Gebäudes so erfasst werden, dass dadurch nicht nur Orte gesucht oder kürzeste Wege ermittelt werden können, sondern auch Navigationsdaten für die Berechnung von Navigationsanweisungen zur Verfügung stehen. Dabei stellt sich die Frage, wie ein solches Umgebungsmodell aufgebaut sein muss und wie es sich berechnen lässt. Zur Bereitstellung für verschiedene Dienste wird ein geeignetes Format benötigt, in dem alle relevanten geometrischen, symbolischen und topologischen Informationen über ein Gebäude abgebildet sind. Eine wichtige Rolle spielen dabei auch unterschiedliche Positionierungsverfahren, die miteinander kombiniert werden können und deren heterogene Positionsdaten integriert werden müssen. Für die Realisierung von Indoor-Navigationssystemen besteht sowohl bei stationären als auch bei mobilen Lösungen eine Herausforderung in der Berechnung von dedizierten Navigationsanweisungen. Diese leiten einen Benutzer anhand von textbasierten oder graphischen Hinweisen zu seinem Ziel, dabei können auch Landmarken in die Navigationsanweisungen mit einbezogen werden. Im Rahmen dieser Dissertation werden neue Lösungen für die vorgenannten Probleme aufgezeigt, und die entwickelten Konzepte werden anhand von Simulationen sowie praktischen Umsetzungen diskutiert.

Location Models and Navigation Data for Indoor Location-based Services

Indoor Location-based Services (I-LBS) collect, process and provide information by taking into account the properties of a building and the locations of mobile targets inside. Application scenarios for I-LBS include personal services such as navigation, location-dependent notifications or searching for resources, as well as corporate services for managing and optimizing workflows, and services that combine location data from targets among several organizations. Creating and utilizing a location model involves several challenges: the properties of a building need to be modeled in a way which allows for computations not limited to searching for nearby places or calculating shortest paths, but also include navigation data, which is required for providing navigation instructions. Thus a central question is which information should be included in a location model and how can it be computed. In order to provide location models for different services, an appropriate format is required, which covers the relevant geometric, symbolic and topological information. Also positioning systems play an important part, given that they can be combined with each other and that their heterogeneous position data need to be integrated. Providing intelligible and dedicated navigation instructions poses a challenge for the realization of both stationary and mobile indoor navigation systems. Such instructions guide users to their destination by means of textual or graphical references, which both can be further enriched by incorporating landmarks into the navigation instructions. This dissertation develops new solutions to the above-mentioned problems and the presented concepts are evaluated on the basis of simulations and practical implementations.

Danke an alle, die mich auf dem Weg dieser Arbeit begleitet haben.

Inhaltsverzeichnis

1. Einleitung	1
2. Ortsbezogene Dienste in Gebäuden - Grundlagen und Definitionen	4
2.1. Grundlegende Begriffe und Definitionen	5
2.1.1. Ortsbezogene und ortsabhängige Dienste	5
2.1.2. Indoor Location-based Services	7
2.1.3. Positionierungsverfahren	10
2.1.4. Umgebungsmodelle	13
2.1.5. Navigation	15
2.2. Problemstellung und Ziel dieser Arbeit	20
2.3. Zusammenfassung	22
3. Bereitstellung von Umgebungsmodellen für I-LBS	23
3.1. Klassifikation von Umgebungsmodellen und verwandte Arbeiten	24
3.2. Anforderungen an ein Umgebungsmodell für I-LBS	29
3.3. Positionsdaten aus heterogenen Positionierungsverfahren	35
3.3.1. Positionsdaten und Koordinatensysteme	38
3.3.2. Kombination von Positionierungsverfahren	42
3.4. Erzeugung von Wegenetzen	55
3.4.1. Randorientierte Algorithmen zur Erzeugung von Wegenetzen	60
3.4.2. Flächenorientierte Algorithmen zur Erzeugung von Wegenetzen:	68
3.4.3. Navigation Meshes	70
3.5. Erzeugung von Umgebungsmodellen	72
3.5.1. Behandlung verschiedener Polygone	75
3.5.2. Empirischer Vergleich für die Erzeugung von Wegenetzen	76
3.6. Umgebungsmodelle für mobile Anwendungen	83
3.6.1. Bündelung von Anfragen	85
3.7. Zusammenfassung und Diskussion	90
4. Ein GML-basiertes Umgebungsmodell für I-LBS	93
4.1. Geography Markup Language	93
4.2. Verwandte Arbeiten	99
4.3. Ein Umgebungsmodell für Indoor LBS	103
4.4. Bewertung und Diskussion	112

5. Indoor Navigation	117
5.1. Verwandte Arbeiten	118
5.2. Navigationsanweisungen und Landmarken	120
5.2.1. Erzeugung von Navigationsanweisungen aus heterogenen Wegenetzen	121
5.2.2. Einbeziehung von Landmarken	126
5.3. Indoor Navigation auf verteilten Anzeigesystemen	129
5.3.1. Verteilte Anzeigesysteme	130
5.3.2. Kontextsensitive Wegewahl	135
5.3.3. Test und Evaluierung	136
5.4. Spontane und anonyme Navigation	138
5.5. Zusammenfassung und Diskussion	142
6. Zusammenfassung und Ausblick	143
6.1. Ausblick	144
6.1.1. Marktplätze für Umgebungsmodelle	144
6.1.2. Spontaner Zugriff auf Indoor-Positionierungsverfahren	144
6.1.3. Indoor-Navigation über neue Benutzerschnittstellen	145
Abbildungsverzeichnis	146
Literaturverzeichnis	148
A. Building Information Geography Markup Language	162

1. Einleitung

Informationen über den Aufenthaltsort und die Umgebung von Menschen und Objekten werden zunehmend in IT-Systemen genutzt, um Inhalte und Dienste ortsbezogen bereit zu stellen. In den vergangenen Jahren haben ortsbezogene Dienste (Location-based Services, LBS) eine rasante Entwicklung durchlebt, und mittlerweile ist die Verwendung von LBS für zahlreiche Benutzer ebenso selbstverständlich wie das Versenden einer SMS. Begünstigt durch den Ausbau von Mobilfunk- und drahtlosen lokalen Netzen ist inzwischen eine nahezu flächendeckende ständige Konnektivität zwischen dem mobilen Endgerät und beliebigen Systemen im Internet gegeben, und längst sind einige Anbieter dazu übergegangen, die Ausführung von Diensten vollständig vom Endgerät in das Netz zu verlagern. Unabhängig von der Art und Weise eines konkreten Dienstes ist jedoch allen LBS gemein, dass sie einerseits Verfahren zur Positionsbestimmung der beteiligten Zielobjekte benötigen, andererseits aber auch ein Modell der Umgebung erforderlich ist, um die Positionsdaten miteinander in Beziehung zu setzen. Beispielsweise existiert mit dem Global Positioning System (GPS) eine weltweit verfügbare Möglichkeit zur Positionsbestimmung im Außenbereich. Gleichzeitig bieten verschiedenste Behörden und Unternehmen umfangreiches Kartenmaterial und Schnittstellen an, über die sich zum Beispiel Aufenthaltsorte visualisieren oder kürzeste Wege in einem Straßennetz berechnen lassen.

Parallel wurden unterschiedliche Positionierungsverfahren für den Einsatz innerhalb von Gebäuden entwickelt und dadurch die Chance für eine neue Klasse von ortsbezogenen Diensten geschaffen: die so genannten *Indoor Location-based Services (I-LBS)*. Diese sammeln, verarbeiten und stellen Informationen bereit, die in Abhängigkeit der Aufenthaltsorte ihrer Zielobjekte sowie auf der Grundlage eines Umgebungsmodells des jeweiligen Gebäudes berechnet werden. Die Anwendungsbereiche von I-LBS reichen dabei von personenbezogenen Diensten wie z.B. Navigation, automatisierten ortsbezogenen Benachrichtigungen und Ressourcen-Suche über organisationsbezogene Dienste wie beispielsweise Steuerung und Optimierung von Arbeitsprozessen und Ressourcen bis hin zu Anwendungen, die personen- und organisationsübergreifend Informationen über Zielobjekte und deren jeweilige Umgebungen austauschen.

Im Gegensatz zu existierenden Location-based Services im Outdoor-Bereich erfordern I-LBS neue Ansätze und Umgebungsmodelle für die Beschreibung und Verarbeitung von Positionsangaben. Im Outdoor-Bereich sind Positionsangaben beispielsweise in Form von Koordinaten innerhalb eines geographischen Koordinatensystems oder einer Post-Adresse möglich. Die darunter liegenden Umgebungsmodelle setzen sich dabei aus einer Menge von geometrischen Objekten und beispielsweise einer hierarchischen Struktur der Orts- und Straßennamen zusammen. Gängige Navigationssysteme im Auto und persönliche Navigationsassistenten verwenden für die Routenplanung und Erzeugung von Navigations-

1. Einleitung

anweisungen zusätzlich umfangreiche Informationen über die Topologie der Straßennetze. Manche Systeme berücksichtigen zusätzlich dynamisch Änderungen im jeweiligen Umgebungsmodell und können so auf temporär nicht verfügbare Wege oder neue Möglichkeiten in der Wegewahl reagieren. Realisiert werden die entsprechenden Dienste mit Hilfe von Geographischen Informationssystemen (GIS), die Umgebungsmodelle und Kartendaten in entsprechenden Datenbanken vorhalten. Solche und weitere Funktionalitäten sind auch für I-LBS relevant, jedoch erfordern sie andere Arten von Umgebungsmodellen, um den speziellen Gegebenheiten innerhalb von Gebäuden gerecht zu werden. Die Berechnung kürzester Wege, das Bereitstellen von Karten und Navigationsanweisungen, die Berücksichtigung unterschiedlicher Maßstäbe und Koordinatensysteme, sowie die Integration von Positionsdaten aus verschiedenen Positionierungsverfahren bringen verschiedene Herausforderungen an die Umgebungsmodelle mit sich.

Relevante Informationen für ein Umgebungsmodell können z.B. die Lage und Form einzelner Räume, die Verbindungen zwischen den Räumen oder beliebige Orte von Interesse sein. Eine weitere wichtige Eigenschaft ist die *Begehbarkeit*, und Informationen darüber werden besonders für Navigationsanwendungen und -dienste benötigt. Die Erzeugung von Umgebungsmodellen gestaltet sich in der Praxis jedoch zumeist schwierig und aufwändig, da 1.) die entsprechenden Informationen aus Rohdaten wie z.B. Bauplänen oder CAD-Dateien extrahiert werden müssen und 2.) keine einheitlichen Standards für die Speicherung der Umgebungsmodelle existieren. Ersteres resultiert daher in einem umfangreichen manuellen Arbeitsaufwand und führt insbesondere für große Anlagen und Gebäude zu hohen Kosten für die Erzeugung des Modells. Fehlende Standards hingegen erschweren den Austausch von Daten zwischen verschiedenen Akteuren und verlangsamen Entwicklungs- und Integrationsprozesse bei neuen Diensten. Zwar existieren unterschiedliche Ansätze zur Speicherung von Gebäudedaten, jedoch sind diese entweder vorrangig für die Visualisierung konzipiert und unterstützen z.B. keine topologischen Beziehungen zwischen Räumen, oder ihre Komplexität ist auf die Nutzung durch Architekten ausgelegt und nicht für die Verwendung durch LBS-Entwickler geeignet.

I-LBS werden bereits jetzt z.B. in Krankenhäusern oder an Flughäfen eingesetzt, um schnell und effektiv Geräte und Personen zu lokalisieren. Gerade beim Einsatz teurer mobiler Geräte rentiert sich die Anschaffung eines solchen Dienstes bereits nach kurzer Zeit, wenn dadurch Redundanzen bei der Anschaffung des Gerätes vermieden werden können. Die existierenden Ansätze und Lösungen für I-LBS zielen jedoch entweder vornehmlich auf die Visualisierung von Positionen auf einer Karte ab oder sie ermöglichen lediglich die rudimentäre Verarbeitung der Koordinaten eines Zielobjekts, wodurch z.B. das Betreten oder Verlassen eines Raumes angezeigt werden kann. Viele existierende Systeme sind an ein spezifisches Positionierungsverfahren gebunden und daher nur schwer mit anderen Positionierungsverfahren kombinierbar. Die Kombination von Positionierungsverfahren würde nicht nur eine bessere Genauigkeit oder Abdeckung ermöglichen, sondern darüber hinaus auch Kosten für die Installation oder den Energieverbrauch auf einem Endgerät positiv beeinflussen. Neue Dienste erfordern in Zukunft nicht nur, dass auch topologische Informationen für die Wegewahl in einem Gebäude berücksichtigt werden können, sondern auch, dass die zugrunde liegenden Daten und Modelle diese Informationen möglichst explizit enthalten.

1. Einleitung

Gleichzeitig gilt es zu beachten, mit welchen Komponenten eine Berechnung durchgeführt wird. Für viele Szenarien ist es wichtig, zwischen der Rolle des Nutzers eines I-LBS und der Rolle des Zielobjekts zu unterscheiden. Diese Rollen können von zwei völlig disjunkten Akteuren übernommen werden, wenn z.B. ein Koordinator die Aufenthaltsorte und Entfernungen von Mitarbeitern und Geräten verwendet, um einen zeitlich und räumlich optimalen Einsatzplan zu erstellen.

Unter den vielen verschiedenen möglichen Szenarien für I-LBS ist die *Indoor Navigation* ein besonderer Fall, denn sie steht in enger Verbindung zu der Art des verfügbaren Umgebungsmodells. Um Navigationsanweisungen in Form von Text oder graphischen Hinweisen erzeugen zu können, müssen genaue Informationen darüber vorliegen, wie ein Weg durch ein Gebäude verläuft. Die Begehrbarkeit kann durch ein Wegenetz ausgedrückt werden, und eine Herausforderung für die Indoor Navigation besteht darin, die Anweisungen unabhängig von der Art des verwendeten Wegenetzes zu berechnen. Zusätzlich können Navigationsanweisungen verbessert werden, wenn sie Hinweise auf in der Umgebung befindliche Landmarken enthalten. Auf diese Art und Weise kann ein Benutzer einfacher der geplanten Route folgen und sich dabei gleichzeitig besser orientieren.

Diese Arbeit konzentriert sich auf die Bereitstellung von Umgebungsmodellen und Navigationsdaten für zukünftige ortsbezogene Dienste in Gebäuden. Es wird untersucht, welche Anforderungen zukünftige I-LBS an den Umfang und die Funktionalität eines Umgebungsmodells stellen, welche relevanten Informationen aus existierenden Datenquellen gesammelt und aufbereitet werden können, wie sich Positionierungsverfahren einbinden und miteinander kombinieren lassen, wie ein Umgebungsmodell in einem offenen Format gespeichert und für Dienste verfügbar gemacht werden kann, und wie Navigationsdaten aus dem entsprechenden Umgebungsmodell gewonnen werden können.

Der Aufbau der Arbeit gliedert sich wie folgt: Im folgenden Kapitel 2 werden grundlegende Begriffe und Definitionen vorgestellt. Insbesondere wird der Begriff der ortsbezogenen Dienste in Gebäuden erläutert, und anhand eines Rollenmodells werden die beteiligten Rollen und Akteure diskutiert, die für Umgebungsmodelle und auch für die Navigation relevant sind. Zum Ende des Kapitels wird die Problemstellung der Arbeit zusammengefasst. Kapitel 3 stellt ausgehend von verschiedenen Anforderungen eine Klassifikation von Umgebungsmodellen und verwandte Arbeiten vor. Anschließend wird die Integration von Positionsdaten aus heterogenen Positionierungsverfahren diskutiert, sowie ein Ansatz zur energieeffizienten Kombination von Positionierungsverfahren vorgestellt. Als zentrales Element für die Erzeugung von Umgebungsmodellen werden verschiedene Ansätze zur Berechnung von Wegenetzen diskutiert und einem empirischen Vergleich unterzogen. Auch wird die Bereitstellung von Umgebungsmodellen für mobile Anwendungen betrachtet. In Kapitel 4 wird ein GML-basiertes Anwendungsschema zur Speicherung und Bereitstellung von Umgebungsmodellen vorgestellt, das die für I-LBS relevanten Informationen vereint. Kapitel 5 behandelt die Erzeugung von Indoor-Navigationsanweisungen, und am Beispiel eines Navigationssystems für Flughäfen wird eine prototypische Umsetzung diskutiert. Zusätzlich wird ein Ansatz zur spontanen und anonymen Navigation auf mobilen Endgeräten vorgestellt. Kapitel 6 schließt mit einer Zusammenfassung der Ergebnisse und gibt einen Ausblick auf zukünftige Fragestellungen.

2. Ortsbezogene Dienste in Gebäuden - Grundlagen und Definitionen

Leistungsfähige mobile Endgeräte, schnelle Kommunikationsnetze und verteilte Systemarchitekturen haben die Entwicklung zahlreicher neuer mobiler Dienste und Anwendungen begünstigt. In der Menge der weltweit verfügbaren Informationen und Systeme können heutzutage nur noch wenige Diensteanbieter alle für einen Nutzer relevanten Informationen vollständig aus eigenen Quellen bereitstellen. Vielmehr werden im Zeitalter mobiler und verteilter Systeme verschiedenste Funktionalitäten und Inhalte über Schnittstellen von anderen Anbietern bezogen und in den eigenen Dienst integriert. Durch (Re-)Kombination vorhandener Inhalte und Dienste können dabei neue, vermischte Inhalte und Dienste entstehen.

Eine zunehmende Anzahl von Diensten reagiert auch kontextabhängig, d.h. dass sich das Verhalten des Dienstes automatisch an den persönlichen, räumlichen, zeitlichen oder auch sozialen Kontext eines Nutzers anpasst (siehe Schilit und Theimer [130], Abowd und Dey [3]). Die Ermittlung des Kontexts kann zum einen über verschiedenste Arten von Sensoren erfolgen, zum anderen auch aus anderen Informationen abgeleitet werden. Für die Modellierung von Kontext unterscheiden Strang und Linnhoff-Popien [140] dabei verschiedene Möglichkeiten: Schlüssel-Wert-Modelle, Auszeichnungssprachen, graphische Modelle, objektorientierte Ansätze, logikbasierte Modelle und Ontologien.

Ortsbezogene Dienste (Location-based Services, LBS) bilden eine Teilmenge der Kontextabhängigen Dienste, für sie sind besonders Informationen über den Aufenthaltsort mobiler Objekte und Daten zur Umgebung eines Ortes von Interesse. Zur Bereitstellung solcher Inhalte bedarf es einerseits Standards und Protokolle zum Austausch der Daten, andererseits müssen auch viele andere Aspekte wie z.B. Aufgaben und Verantwortlichkeiten der beteiligten Akteure, Abrechnungsmodelle, sowie Sicherheits- und Datenschutzanforderungen berücksichtigt werden.

Fortschreitende Automatisierung in vielen Organisationen und das Aufkommen neuer Technologien begünstigen auch die Entwicklung von neuen LBS, die innerhalb von Gebäuden operieren. Ortsbezogene Dienste in Gebäuden (*Indoor Location-based Services, I-LBS*) sammeln, verarbeiten und stellen Informationen bereit, die auf der Grundlage von Positionsdaten stationärer oder mobiler Objekte sowie von Informationen über die Eigenschaften des Gebäudes berechnet werden. Die Gewinnung von Positionsdaten kann über verschiedenste Positionierungsverfahren erfolgen, und je nach vorhandenem Verfahren ergeben sich unterschiedliche Genauigkeiten, Latenzzeiten und Herausforderungen bei der Positionsbestimmung. Relevante Eigenschaften des Gebäudes können z.B. die Lage und Form von Räumen, die Begehbarkeit und Länge von kürzesten Wegen durch das Gebäude,

2. Ortsbezogene Dienste in Gebäuden - Grundlagen und Definitionen

die Positionen von Objekten von Interesse oder auch Kartenmaterial zur Visualisierung sein. Solche und weitere Eigenschaften werden in einem entsprechenden *Umgebungsmodell* des Gebäudes gespeichert.

An der Ausführung eines I-LBS können mehrere Komponenten beteiligt sein, und je nach Art des Dienstes werden Informationen von unterschiedlichen Akteuren bereit gestellt, die dabei in unterschiedlichen Rollen aktiv sein können. Insbesondere ist dabei zwischen der Rolle des Nutzers und der des Zielobjektes zu unterscheiden, denn bei vielen I-LBS werden diese beiden Rollen von unterschiedlichen Akteuren übernommen.

In diesem Kapitel werden zunächst grundlegende und für diese Arbeit relevante Begriffe definiert und erläutert. Ausgehend vom Begriff des *Dienstes* und des *ortsbezogenen Dienstes* werden Indoor Location-based Services und ihre beteiligten Rollen, Positionierungsverfahren und der Begriff der Navigation behandelt, sowie eine Klassifikation von Navigationssystemen vorgestellt. Ein weiterer zentraler Begriff in dieser Arbeit ist das Umgebungsmodell (Location Model), welches die räumlichen Eigenschaften von Orten beschreibt und in dem vorgestellten Rollenmodell eine wichtige Aufgabe übernimmt. Aufbauend auf den dargestellten Anwendungsgebieten werden anschließend die Problemstellung und das Ziel dieser Arbeit formuliert.

2.1. Grundlegende Begriffe und Definitionen

Der Begriff **Dienst (Service)** taucht an vielen verschiedenen Stellen in der Informationstechnologie auf und je nach Fachgebiet existieren unterschiedliche Definitionen. Während im Zeitalter der ersten Telekommunikationsnetze für verschiedene Sprach-, Text- und Datendienste noch jeweils eigene Netze verwendet wurden, sind viele der heutigen IT-Dienste entkoppelt vom darunter liegenden Kommunikationsnetz. Mit dem Aufkommen dienstorientierter Architekturen (Service-oriented Architectures) definiert Jones [68] einen Service folgendermaßen:

„A service’s intention is to undertake certain functions to provide value to the business; its specification isn’t just the direct service it provides but also the environment in which it undertakes those functions.“

Dabei werden verschiedene Kategorien unterschieden, anhand derer ein Dienst beschrieben werden kann, z.B. *performance, capacity, business organization, risks and issues, ownership, reliability, security, business impact, tolerance, service contract (preconditions, postconditions, invariants)* und *dependencies*.

2.1.1. Ortsbezogene und ortsabhängige Dienste

Ortsbezogene Dienste (Location-based Services, LBS) werden oftmals auch als *Local-based Services, Location Services* oder *Location-related Services* bezeichnet und bilden eine Klasse von Diensten, die Küpper [84] folgendermaßen definiert:

2. Ortsbezogene Dienste in Gebäuden - Grundlagen und Definitionen

„Location-based Services (LBSs) are IT services for providing information that has been created, compiled, selected, or filtered taking into consideration the current locations of the users or those of other persons or mobile devices.“

Eine andere Definition gibt die GSM Association [53] und beschreibt LBS als

„[...] services that use the location of the target for adding value to the service.“

Das 3rd Generation Partnership Project (3GPP) [1] gibt eine ähnliche Definition für LBS:

„[...] service provided either by teleoperator or a 3rd party service provider that utilizes the available location information of the terminal.“

Es existieren auch enger gefasste Definitionen, wie z.B. vom Open Geospatial Consortium [110]:

„A wireless-IP service that uses geographic information to serve a mobile user. Any application service that exploits the position of a mobile terminal.“

Generell lassen sich LBS in zwei verschiedene Klassen von Diensten einteilen:

- *Reaktive LBS* werden durch explizites Anfragen (pull) des Benutzers aktiviert und entsprechen einer synchronen Interaktion zwischen Benutzer und Dienst.
- *Proaktive LBS* werden automatisch beim Eintreten zuvor festgelegter ortsbezogener Ereignisse ausgelöst (push). Beispiele dafür sind Benachrichtigungen, die ein Benutzer erhält, sobald ein Zielobjekt einen bestimmten Bereich betreten oder verlassen hat oder auch automatisch angezeigte Informationen zu den in der Nähe eines Zielobjekts befindlichen Orten von Interesse (wobei in beiden Fällen die Rolle des Benutzers und des Zielobjekts gleichzeitig von ein und demselben Akteur übernommen werden können).

Bei der organisatorischen Betrachtung von LBS ist der Begriff der **Rolle** ein zentrales Element. Eine Rolle kennzeichnet eine Menge von Aufgaben, die von einem Akteur ausgeführt werden. Die an der Erbringung eines LBS beteiligten Rollen werden von verschiedenen Akteuren (Personen, Organisationen oder Unternehmen) übernommen, diese lassen sich in operative und nichtoperative Akteure unterteilen [84]. Operative Akteure sind z.B. das *Zielobjekt*, der *Benutzer*, der *Positionsermittler*, der *Location Provider*, der *Content Provider* oder der *LBS Provider*. Sie alle stellen entlang einer LBS-Wertschöpfungskette über Schnittstellen bestimmte Funktionen zur Verfügung und werden im nachfolgenden Abschnitt genauer beschrieben. Auf der anderen Seite stehen nichtoperative Akteure wie beispielsweise Standardisierungsgremien, Hersteller, Forschergruppen oder Regulierungsbehörden. Diese sind nicht direkt an der Ausführung eines LBS beteiligt, steuern jedoch generelle Aspekte der Entwicklung.

Im Gegensatz zu LBS bezeichnen die **Ortsabhängigen Dienste (Location-dependent Services)** die Bereitstellung von Dienst-Funktionalitäten in Abhängigkeit des Aufenthaltsortes des Nutzers. Das 3GPP [1] gibt dazu folgende Definition:

2. Ortsbezogene Dienste in Gebäuden - Grundlagen und Definitionen

„Location Dependent Service: service provided either by teleoperator or a 3rd party service provider that is available (pull type) or is activated (push type) when the user arrives to a certain area. It doesn't require any subscription in advance, but the push type activation shall be confirmed by the user. The offered service itself can be any kind of service (e.g. a public Xerox machine or the discount list in a store).“

Beispielsweise kann der Zugriff eines mobilen Nutzers auf bestimmte Ressourcen dadurch limitiert werden, dass die Verbindung über einen bestimmten drahtlosen lokalen Zugriffspunkt erfolgt muss.

2.1.2. Indoor Location-based Services

Ortsbezogene Dienste innerhalb von Gebäuden (Indoor Location-based Services, I-LBS) unterscheiden sich von den herkömmlichen LBS im Wesentlichen durch zwei Aspekte: zum einen sind die im Außenbereich verbreiteten Positionierungssysteme *GPS* und *Zell-basierte Ortung* oft nicht verfügbar, so dass verschiedene andere Positionierungsverfahren zum Einsatz kommen. Zum anderen werden generell Informationen über den Aufbau und die Topologie des Gebäudes benötigt, um so beispielsweise den kürzesten Weg zwischen zwei Räumen oder die Positionen von in der Nähe befindlichen Orten von Interesse zu berechnen. Solche und weitere Informationen werden durch ein entsprechendes **Umgebungsmodell** des Gebäudes bereit gestellt. Hinzu kommt, dass sowohl die Indoor-Positionierungsverfahren als auch das Umgebungsmodell eines Gebäudes nicht ohne Weiteres öffentlich verfügbar sind.

Die möglichen Szenarien für I-LBS verteilen sich auf unterschiedliche Bereiche: Dienste für den einzelnen Benutzer können beispielsweise die Visualisierung des eigenen Standortes, der Abruf von Informationen über Ressourcen in einem bestimmten Bereich oder die Anzeige von schrittweisen Navigationsanweisungen zum Finden eines Ziels sein. Aus der Sicht eines Unternehmens stellt die ortsbezogene Disposition und Steuerung von Ressourcen ein wichtiges Einsatzgebiet dar, wenn dadurch z.B. mobile Geräte in einem Krankenhaus verwaltet werden können. In Umgebungen mit vielen gleichzeitigen Besuchern können auch die verschiedenen Aufenthaltsorte von Personen miteinander in Bezug gesetzt werden, um so beispielsweise einen besseren Ablauf zu gewährleisten oder um Benutzer auf andere Benutzer in der Nähe aufmerksam zu machen.

Existierende Rollenmodelle für LBS im Outdoor-Bereich unterscheiden zwischen den Rollen *Target*, *Position-Enabler*, *Location-Provider*, *LBS-Provider*, *Content-Provider* und *LBS User* (Küpper et al. [85]). Je nach Anwendungsfall werden diese Rollen von verschiedenen Akteuren übernommen, dabei können Akteure auch manche Rollen gleichzeitig übernehmen. Wie bei bisherigen LBS erfordern die I-LBS auf der einen Seite das Ermitteln und Bereitstellen von Positionsdaten beteiligter Zielobjekte, auf der anderen Seite ist zusätzlich das Erzeugen und Bereitstellen von Informationen über die gewünschte Umgebung relevant. Bei der Ausführung von I-LBS sind daher zwei zusätzliche Rollen notwendig:

2. Ortsbezogene Dienste in Gebäuden - Grundlagen und Definitionen

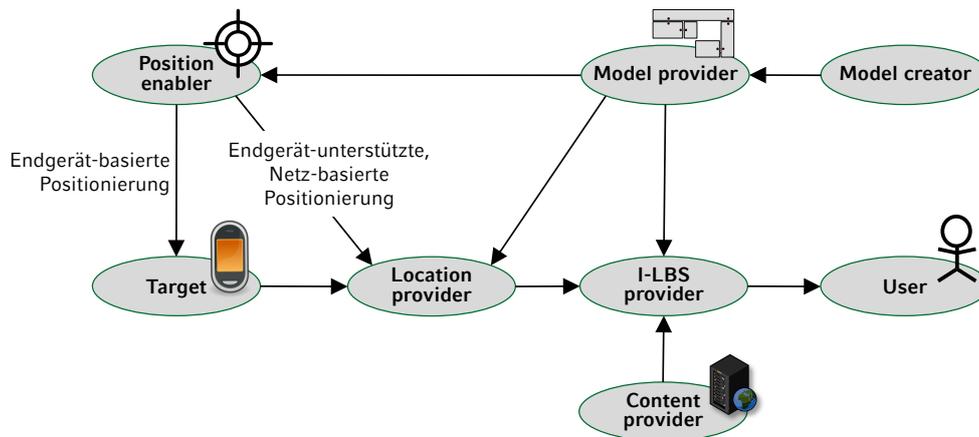


Abbildung 2.1.: Rollenmodell für I-LBS.

Model-Creator und *Model-Provider*. Abbildung 2.1 zeigt ein um diese beiden Rollen erweitertes Modell der bei I-LBS beteiligten Rollen (Ruppel et al. [127]):

- Das **Zielobjekt (Target)** ist eine Entität, dessen Aufenthaltsinformationen im Rahmen der Ausführung eines I-LBS verarbeitet werden. Zielobjekte können z.B. eine Person, ein Mobiltelefon, ein Laptop, sowie beliebige andere technische Geräte und Einrichtungsgegenstände wie beispielsweise ein Krankenhausbett oder ein mobiler Container sein.
- Die Rolle des **Positionsermittlers (Position-Enabler)** beschreibt eine Entität, die mittels eines Positionierungsverfahrens in der Lage ist, den Aufenthaltsort eines oder mehrerer Zielobjekte zu bestimmen. Dazu zählen beispielsweise ein GPS-Empfänger, ein RFID-Lesegerät oder eine Plattform zur netzbasierten Lokalisierung von mobilen WLAN-Stationen. Der Positionsermittler stellt eine Schnittstelle bereit, über die die Position des Zielobjekts in Form einer geometrischen oder symbolischen Koordinate ermittelt werden kann.
- Der **Location-Provider** nimmt eine vermittelnde Stellung zwischen dem Positionsermittler und dem LBS-Provider ein. Er stellt Informationen zum Aufenthaltsort eines Zielobjekts oder auch zu räumlichen Beziehungen zwischen mehreren Zielobjekten für den LBS-Provider bereit, indem er Positionsdaten sammelt, aufbereitet und für die Positionierung spezifische Aktionen wie z.B. eine synchrone Ortung (Polling) auslöst.
- Die Rolle des **I-LBS-Providers** entspricht dem Betreiber eines Location-based Service. Gemäß der obigen Definition stellt er Informationen bereit, die auf der Grundlage der Aufenthaltsorte der Zielobjekte sowie der Informationen über die Eigenschaften des umgebenden Gebäudes berechnet werden. Zur Erbringung seiner Funktionalitäten greift der I-LBS-Provider daher auf Informationen des Location-Providers, des

2. Ortsbezogene Dienste in Gebäuden - Grundlagen und Definitionen

Inhalte-Anbieters und des Umgebungsmodell-Providers zu. Der vom I-LBS-Provider angebotene Dienst definiert sich anhand der entsprechenden Funktionalitäten der Benutzerschnittstelle, über die ein Benutzer auf den Dienst zugreift.

- Der **Umgebungsmodell-Erzeuger (Location-Model-Creator)** ist zuständig für die Erstellung von Umgebungsmodellen. Dazu zählt beispielsweise die Ermittlung von Raumpolygonen und Wegegraphen aus Bauplänen oder CAD-Dateien sowie die Einordnung der Daten in verschiedene Koordinatensysteme. Oft ist zur Durchführung dieses Prozesses sehr domänenspezifisches Wissen notwendig, um beispielsweise lokale Konventionen korrekt umzusetzen.
- Ein **Umgebungsmodell-Anbieter (Location-Model-Provider)** stellt Schnittstellen für den Zugriff auf Umgebungsmodell-spezifische Informationen bereit. Dies können z.B. für gegebene geometrische Koordinaten die symbolischen Bezeichner der zugehörigen Orte oder auch die Länge des kürzesten Weges zwischen den Orten sein. Je nach Anwendungsfall können neben dem I-LBS Provider auch Positionsermittler oder Location-Provider auf diese Schnittstellen zugreifen, beispielsweise wenn der Location-Provider anhand einer vom Positionsermittler erhaltenen Koordinate vorab überprüft, ob ein Zielobjekt einen bestimmten Raum betreten oder verlassen hat und erst beim Eintreten einer bestimmten Bedingung den I-LBS-Provider benachrichtigt.
- Zusätzliche Inhalte wie z.B. Bilder, Kartenmaterial oder Multimedia-Daten werden dem I-LBS-Provider durch **Inhalteanbieter (Content-Provider)** bereit gestellt.
- Der **Benutzer (User)** greift auf den vom I-LBS-Betreiber bereitgestellten Dienst über entsprechende Benutzerschnittstellen z.B. auf einem mobilen Endgerät oder einem stationären Computer zu.

Die Interaktion zwischen den verschiedenen Rollen ist durch entsprechende Schnittstellen vorgegeben, die im Rahmen dieser Arbeit genauer erläutert werden. Das vorgestellte Rollenmodell sieht zunächst keine spezielle Zuordnung von bestimmten Akteuren zu bestimmten Rollen vor, d.h. dass sich die konkreten Verantwortlichkeiten eines Akteurs nach den jeweils zugeteilten Rollen richten, und diese können je nach Anwendungsfall variieren. In manchen Szenarien ist der Benutzer gleichzeitig auch das Zielobjekt, beispielsweise wenn die eigene Position auf einer Karte dargestellt werden soll. Eine Trennung zwischen der Rolle des Location-Providers und des I-LBS-Providers ist erst dann möglich, wenn tatsächlich eine dienstunabhängige Positionierungs-Infrastruktur vorhanden ist, die Positionsdaten in einem einheitlichen Format bereitstellt. Auch kann bereits durch Verwendung bestimmter Positionierungsverfahren die Rollenverteilung teilweise vorgegeben werden: endgerätbasierte Positionierungsverfahren (siehe auch 3.3) bedingen z.B., dass Positionsermittler und Zielobjekt als Akteure eine Einheit bilden. Bei netzbasierten Positionierungsverfahren liegt eine explizite Trennung zwischen beiden Rollen vor.

Die Aufgabe des Location-Model-Providers besteht darin, mit Hilfe der vom Location-Model-Creator erhaltenen Umgebungsmodelle eine Schnittstelle bereit zu stellen, die es

2. Ortsbezogene Dienste in Gebäuden - Grundlagen und Definitionen

dem I-LBS-Provider erlaubt, Anfragen auf dem Umgebungsmodell zu stellen. Diese Anfragen umfassen unter anderem:

- **A1:** Für eine gegebene Position die Umrechnung von geometrischen in symbolische Koordinaten und umgekehrt (Geocoding und Reverse-Geocoding).
- **A2:** Die Berechnung des kürzesten Weges (Walking-Distance) zwischen zwei Orten.
- **A3:** Für eine gegebene Position die Suche nach den k nächstgelegenen Orten von Interesse.
- **A4:** Die Berechnung von Euklidischen Distanzen zwischen zwei beliebigen Positionen.
- **A5:** Die Ausgabe von Lage und Form von Räumen, Stockwerken und Gebäuden.

Es existieren Anwendungsfälle, in denen zusätzlich weitere Anfragen sinnvoll sein können, beispielsweise wenn ein Dienst die Positionen mehrerer mobiler Zielobjekte verfolgen will. Zur effektiven und effizienten Erkennung räumlicher Beziehungen zwischen den Zielobjekten müsste der Location-Provider unter Umständen sehr viele Anfragen beim Location-Model-Provider stellen – z.B. zur kontinuierlichen Neuberechnung der gehbaren Entfernung. In diesem Fall wäre eine zusätzliche Funktionalität notwendig, die es ermöglicht,

- **A6:** für eine gegebene Position p , einen Radius r und eine Distanz-Funktion diejenigen Objekte im Umgebungsmodell zu bestimmen, deren Distanz zu p kleiner als r ist.

Diese Informationen könnte der Location Provider intern für eine effiziente Überwachung der Zielobjekte nutzen. Das vorgestellte Rollenmodell schließt auch mit ein, dass sich verschiedene Model Provider untereinander zusammenschließen können und Anfragen untereinander weiterreichen.

2.1.3. Positionierungsverfahren

Ein **Positionierungsverfahren** ermittelt den Aufenthaltsort einer mobilen Entität. Jedes Positionierungsverfahren wird bestimmt durch einen oder mehrere messbare Parameter, eine Methode zur Berechnung der Position anhand der gemessenen Parameter, ein beschreibendes oder räumliches Referenzsystem innerhalb dessen die Angabe der berechneten Position erfolgt, eine Positionierungs-Infrastruktur sowie Protokolle zur Koordination der Positionierung.

Messbare Parameter können z.B. der Empfang eines Pilotsignals, die Laufzeit oder der Pfadverlust von Pilotsignalen, die Laufzeitdifferenzen oder Pfadverlustdifferenzen zwischen verschiedenen Pilotsignalen oder auch die empfangene Signalstärke von Pilotsignalen sein. Weitere Parameter sind beispielsweise Bewegungsrichtung, Geschwindigkeit und zurückgelegte Entfernung, welche mittels eines Gyroskops, Beschleunigungsmessers und Wegstreckenzählers ermittelt werden können.

2. Ortsbezogene Dienste in Gebäuden - Grundlagen und Definitionen

Methoden zur Berechnung der Position lassen sich unterteilen in die Erkennung von räumlicher Nähe (Proximity-Sensing), Laufzeitmessung (Lateration), Winkelbildung (Angulation), Koppelnavigation (Dead-Reckoning), Mustererkennung (Pattern-Matching) sowie Kombinationen der vorgenannten Methoden. Je nach Anwendungsfall umfasst die zugrunde liegende **Positionierungs-Infrastruktur** unterschiedliche Komponenten. Im Falle von GPS setzt sie sich beispielsweise aus dem Raumsegment (den Satelliten), dem Bodensegment (den Kontrollstationen auf der Erde) und den GPS-Empfängern (in den Endgeräten) zusammen. Andere Infrastruktur-Komponenten können z.B. die Basisstationen eines Mobilfunknetzes, WLAN-Access-Points, WLAN- oder RFID-Tags, Smartcards, Sensornetze, Laptop-Computer oder auch Barcodes sein.

Abhängig von der Beteiligung der verschiedenen Rollen unterscheidet man im Allgemeinen zwischen drei verschiedenen Klassen von Positionierungsverfahren:

1. **Endgerät-basiert (Terminal-based Positioning):** Sowohl die Messung der Parameter als auch die Positionsbestimmung geschieht vollständig im Endgerät.
2. **Endgerät-unterstützt (Terminal-assisted Positioning):** Die Messung erfolgt zumeist im Endgerät und die Berechnung der Position anschließend durch eine leistungsfähige Komponente im Netz.
3. **Netz-basiert (Network-based Positioning):** Sowohl Messung als auch Berechnung werden außerhalb des Endgerätes durchgeführt, und das Endgerät selbst sendet beispielsweise lediglich ein periodisches Pilotsignal aus.

Endgerätbasierte Positionierungsverfahren haben für die Besitzer der mobilen Endgeräte den Vorteil, dass die Kontrolle über das Positionierungsverfahren vollständig innerhalb der Domäne des Besitzers liegt. Dadurch kann der Datenschutz der Ortsinformationen (*Location Privacy*) auf unterster Ebene sichergestellt werden. Gleichzeitig erfordern die endgerätbasierten Positionierungsverfahren in der Regel leistungsfähige Komponenten im Endgerät und verursachen dementsprechend einen erhöhten Energieverbrauch.

Vergleicht man die möglichen Kombinationen von Positionierungsverfahren und Anwendungen, so werden jeweils verschiedene Vor- und Nachteile erkennbar (siehe Tabelle 2.1): Terminal-basierte Positionierungsverfahren bieten dem Zielobjekt eine volle Kontrolle über die Erfassung von Positionsdaten. Ist die Anwendung ebenfalls rein endgerätbasiert, so bleibt auch die Verarbeitung der Positionsdaten unter der Kontrolle des Zielobjekts, und die *Location Privacy* bleibt geschützt. Die vom Endgerät unterstützten Positionierungsverfahren bieten den Vorteil, dass die gemessenen Parameter durch einen Positionierungs-Server verarbeitet werden, der in der Regel unabhängig von der Infrastruktur des jeweiligen Gebäudes betrieben werden kann.

Abbildung 2.2 zeigt ein Beispiel für ein endgeräatunterstütztes WLAN-Positionierungssystem. Das zu ortende Zielobjekt verwendet zur Messung von Parametern ein integriertes WLAN-Modul und überträgt die gemessenen Daten über ein Mobilfunknetz an den Positionierungsserver. Erfolgt die Positionierung rein Netz-basiert, so hat zwar der Betreiber der Positionierungsinfrastruktur die volle Kontrolle über die Erfassung und Verarbeitung der

2. Ortsbezogene Dienste in Gebäuden - Grundlagen und Definitionen

	Terminal-basierte Positionierung: ⊕ Einfache Nutzung, Datenschutzfreundliche Positionsbestimmung, Skalierbarkeit ⊖ Hoher Aufwand für das Endgerät	Terminal-unterstützte Positionierung: ⊕ Realisierung ohne Zugriff auf lokale Netze möglich ⊖ Mäßiger Aufwand für das Endgerät, abweichende Messwerte durch heterogene Hardware	Netz-basierte Positionierung: ⊕ Geringer Aufwand für das Endgerät ⊖ Datenschutzrisiko aufgrund zentraler Erfassung der Positionsdaten
Terminal-basierte Anwendung: ⊖ Nur selbstverweisende Dienste möglich, Speicherbedarf für Umgebungsmodell	⊕ Volle Kontrolle durch den Benutzer, alle Berechnungen auf dem Endgerät ⊖ Keine dynamischen Informationen verfügbar	⊖ Overhead bei fortlaufender Positionierung durch Upload von gemessenen Parametern und Download von berechneter Position	⊖ Authentifizierung und Autorisierung des Netz-basierten Positionierungssystems ist nicht trivial
Server-unterstützte Anwendung: ⊕ Querverweisende Dienste, dynamisches Laden benötigter Daten	⊕ Flexible Kombination ⊖ Bei vielen Zielobjekten: erhöhter Aufwand durch Austausch von Positionsdaten	⊕ Piggy-Backing von Anwendungs- und Positionsdaten möglich ⊖ Overhead bei fortlaufender Positionierung	⊕ Ermöglicht umfangreiche Anwendungen ⊖ Enge Kopplung zwischen Anwendung und Positionierungssystem
Server-basierte Anwendung: ⊕ Gesamtsicht auf alle Daten	⊕ Monitoring von autonomen mobilen Tags ⊖ Energieverbrauch	⊕ Flexible Ausbringung in fremden Umgebungen möglich	⊕ Volle Kontrolle durch den Dienstbetreiber, alle Berechnungen im Netz

Tabelle 2.1.: Vorteile (⊕) und Nachteile (⊖) von Kombinationen unterschiedlicher Positionierungsverfahren und Anwendungen.

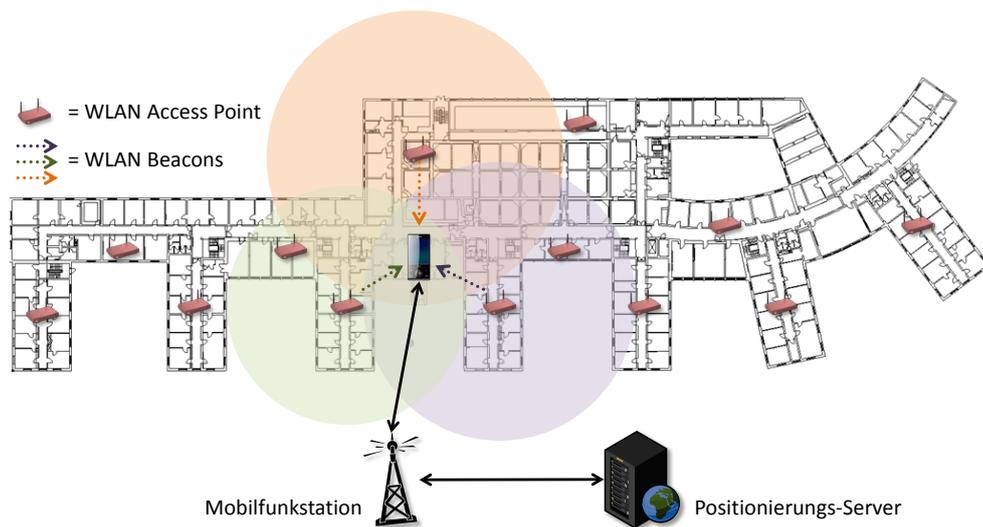


Abbildung 2.2.: Beispiel für eine endgerätunterstützte WLAN-Positionierung: Das Endgerät empfängt die Beacon-Frames umliegender Access-Points und schickt die gemessenen Daten (z.B. MAC-Adresse plus empfangene Signalstärke oder Signal-zu-Rauschverhältnis) über ein Mobilfunknetz an einen entfernten Positionierungs-Server.

2. Ortsbezogene Dienste in Gebäuden - Grundlagen und Definitionen

Positionsdaten, gleichzeitig stellt sich jedoch auch die Frage nach dem ausreichenden Schutz der Privatsphäre. Auch bei einer endgerätbasierten Positionierung in Verbindung mit einer Server-basierten Anwendung ist die Privatsphäre des Zielobjekts gefährdet, insbesondere dann, wenn die Positionsdaten mehrerer Zielobjekte miteinander korreliert werden sollen. Ein Ansatz zur Anonymisierung der Positionsdaten bei solchen Diensten wurde erstmals von Treu et al. [146] skizziert und wird ausführlich von Ruppel et al. [128] behandelt.

Auf konkrete Positionierungsverfahren und deren Technologien sowie auf die Kombination von Positionierungsverfahren wird im Abschnitt 3.3 genauer eingegangen.

2.1.4. Umgebungsmodelle

Positionsdaten von Zielobjekten werden bei der Ausführung eines ortsbezogenen Dienstes relativ zum räumlichen Kontext der Zielobjekte verarbeitet. Beispielsweise erfordern die Suche nach einem nächstgelegenen Ort von Interesse oder die Berechnung eines kürzesten Weges zu einem Zielpunkt Informationen über die Beschaffenheit der Umgebung.

Ein **Umgebungsmodell** (*Location Model*) beschreibt räumliche Eigenschaften eines Ortes. Dies können bei ortsbezogenen Diensten in Gebäuden beispielsweise die Lage und Form von Gebäuden, Räumen, Wegen, Treppen, Aufzügen und Ein- und Ausgängen, aber auch topologische Informationen über die Begehbarkeit des Ortes, Entfernungen zwischen Wegpunkten oder die symbolischen Bezeichner der Elemente des Ortes sein.

Man unterscheidet grundsätzlich zwei verschiedenen Arten von Umgebungsmodellen: **geometrische Umgebungsmodelle** bilden die Eigenschaften eines Ortes durch geometrische Koordinaten ab. **Symbolische Umgebungsmodelle** verwenden symbolische Bezeichner und lassen sich weiter untergliedern in **mengenbasierte** und **graphbasierte** Modelle. Durch Kombination von geometrischen und symbolischen entstehen **hybride Umgebungsmodelle**. Auf die Eigenschaften dieser verschiedenen Ansätze und auf verwandte Arbeiten wird im Abschnitt 3.1 genauer eingegangen.

Bezogen auf das Rollenmodell in Abbildung 2.1 ist ein Umgebungsmodell für die Rolle des I-LBS Providers, des Location Providers und des Positionsermittlers von Interesse:

- Der I-LBS Provider kann anhand des Umgebungsmodells z.B. die *walking distance* (kürzester Weg eines Fußgängers) zwischen zwei Räumen berechnen, aus der Folge von Wegpunkten Navigationsanweisungen erzeugen oder auch geometrische Formen von Orten bzw. Kartendaten zur Visualisierung abfragen. Weitere Funktionalitäten sind beispielsweise die Suche nach Orten von Interesse oder die Suche nach einem Zielobjekt innerhalb eines bestimmten Bereiches von Interesse. Insbesondere für Dienste, die auf mobilen Endgeräten ausgeführt werden, spielen die Komplexität des Modells sowie die effiziente Übertragung von Umgebungsmodell-Informationen eine wichtige Rolle. In diesem Fall kann es z.B. sinnvoll sein, Teilausschnitte eines Modells bereitzustellen, so dass bestimmte Anfragen auf dem mobilen Endgerät schneller beantwortet werden können.
- Ein Akteur in der Rolle des Positionsermittlers kann ebenfalls von den Informationen eines Umgebungsmodells profitieren, um die Genauigkeit und Präzision eines Posi-

2. Ortsbezogene Dienste in Gebäuden - Grundlagen und Definitionen

tionierungsverfahrens zu verbessern. Wenn ein Positionierungsverfahren rein auf der Messung von Signalstärken und daraus resultierenden Fingerprints basiert, so kann es durch Störungen in der Umgebung (z.B. durch viele anwesenden Personen, bauliche Änderungen, andere Radiosignale) zu starken Abweichungen in den berechneten Positionen kommen. Diese Abweichungen werden in der Regel durch entsprechende Filter korrigiert, die jedoch lediglich auf der Grundlage Euklidischer Distanzen operieren. Durch die Verwendung topologischer Distanzen können diese Filtermechanismen verbessert werden. Sei e ein mobiles Zielobjekt und $p(e_t)$ die durch einen Positionsermittler berechnete Position von e zum Zeitpunkt t . Ferner sei $d_{\text{euclid}}(p_i, p_j)$ eine Funktion zur Berechnung der Euklidischen Distanz zwischen zwei Positionen und $d_{\text{topo}}(p_i, p_j)$ eine Funktion zur Berechnung der topologischen Distanz (walking distance) zwischen zwei Orten. Vergleicht nun der Positionsermittler die verschiedenen Distanzen zwischen zwei aufeinander folgenden Zeitpunkten und stellt fest, dass $d_{\text{euclid}}(p(e_t), p(e_{t+1})) \ll d_{\text{topo}}(p(e_t), p(e_{t+1}))$, so ist ein solcher Sprung in der topologischen Distanz ein Indiz dafür, dass in der Positionsberechnung ein Ausreißer aufgetreten ist. Da e in der Realität mit einer zuvor bekannten maximalen Bewegungsgeschwindigkeit in einem kurzen Zeitintervall keine große topologische Distanz hätte zurücklegen können, deutet diese Situation auf eine Störung des Positionierungssystems hin.

- Der Location Provider kann ebenfalls Informationen aus dem Umgebungsmodell verwenden, um beispielsweise höherwertige Funktionen für den I-LBS Provider zur Verfügung zu stellen. Ein mögliches Szenario hierfür ist beispielsweise das Erkennen des Zeitpunktes, an dem ein Zielobjekt einen bestimmten Bereich betritt oder verlässt, wobei der Bereich aufgrund des Umgebungsmodells auf einfache Weise durch symbolische Koordinaten beschrieben werden kann. Eine andere Möglichkeit besteht darin, die Annäherung zweier Zielobjekte (Proximity detection) nicht nur auf Grund der Euklidischen, sondern auch der topologischen Distanz zwischen den beiden Zielobjekten zu erkennen.

Die Speicherung eines Umgebungsmodells kann auf unterschiedliche Weisen erfolgen. Jedoch ist dabei zu bedenken, dass der Zugriff auf das Umgebungsmodell aufgrund der Heterogenität von I-LBS möglichst plattformunabhängig und ohne großen Rechenaufwand möglich sein sollte. Insbesondere Letzteres erfordert z.B., dass die Funktionalitäten des Umgebungsmodells möglichst durch explizit im Modell vorhandene Informationen bereitgestellt werden können und diese Informationen nicht erst bei jedem Aufruf durch zusätzliche Algorithmen aufwändig gesammelt werden müssen. Ein Beispiel dafür ist die explizite Bereitstellung eines entsprechenden Wegenetzes, anhand dessen Routinganfragen schnell beantwortet werden können. Ist kein Wegenetz vorhanden und werden beispielsweise nur Lage und Form von Räumen sowie Türen und Verbindungen im Modell gespeichert, so müsste jeder Anwendungsentwickler mit einem vergleichbar großen Aufwand zuerst ein eigenes Wegenetz berechnen.

Für den Austausch von Umgebungsmodellen zwischen verschiedenen Akteuren bietet sich die Verwendung einer Auszeichnungssprache (Markup Language) an. Dies bietet ein

2. Ortsbezogene Dienste in Gebäuden - Grundlagen und Definitionen

durch Menschen lesbares Format und ist einfach zu bearbeiten. Ein im Rahmen dieser Arbeit entwickeltes Modell baut auf der Extensible Markup Language (XML) des World Wide Web Consortium (W3C) auf und ist ein Applikationsschema gemäß der vom Open Geospatial Consortium (OGC) standardisierten Geography Markup Language 3.2 (GML). Damit ist zusätzlich eine große Plattformunabhängigkeit gewährleistet, und es existieren sehr viele verschiedene Programme zur Verarbeitung der XML-Daten.

2.1.5. Navigation

Navigation bezeichnet den zeit- und ortsabhängigen Prozess der Wegefindung von einem Start- zu einem Zielpunkt. Ein **Navigationssystem** stellt Navigationsanweisungen zur Verfügung, die eine Person zu ihrem Ziel leiten: dies können akustische, visuelle, oder auch haptische Anweisungen sein. Die Bereitstellung von **Navigationsanweisungen** umfasst dabei drei Schritte:

1. Die Positionsbestimmung der mobilen Entität/des Benutzers.
2. Die Berechnung eines Weges zum Zielpunkt auf der Grundlage eines Umgebungsmodells und einer Metrik.
3. Die Erzeugung von Navigations-Instruktionen für die mobile Entität.

Innerhalb der vergangenen Jahre hat die Verbreitung von Navigationssystemen in Fahrzeugen stark zugenommen, und auch auf mobilen Endgeräten sind vermehrt Navigationssysteme verfügbar. Die existierenden Ansätze zielen jedoch auf die Verwendung im Outdoor-Bereich ab und stellen dementsprechend Navigationsinstruktionen zur Wegefindung auf Straßen bereit.

Auch bei den ortsbezogenen Diensten innerhalb von Gebäuden spielt Navigation eine wichtige Rolle. Da neben der reinen Positionierung und der Visualisierung der Positionsdaten zusätzlich auch noch die räumlichen und topologischen Beziehungen zwischen Orten und Objekten berücksichtigt werden müssen, stellen die Indoor-Navigationssystemen zahlreiche Anforderungen gleichzeitig.

Neben den technischen Ansätzen der Informatik haben in den vergangenen Jahren besonders die *Neurowissenschaften* wichtige Erkenntnisse auf dem Gebiet der menschlichen Navigation beigetragen. So geben Maguire et al. [98] einen ausführlichen Überblick über bisherige Arbeiten auf diesem Gebiet. Dabei ist eine wichtige Feststellung, dass die kognitiven Prozesse bei der Navigation durch eine Umgebung eine eigene Struktur aufweisen. Sie unterscheiden sich von kognitiven Problemlösungsprozessen zur räumlichen Orientierung (z.B. wenn ein Proband aus der Vogelperspektive auf einer Tischplatte Aufgaben zur räumlichen Orientierung löst). Auch wurde gezeigt, dass aus den Erkenntnissen aus Versuchen mit einem allozentrischen Bezugssystem keine direkten Schlüsse auf Darstellungen mit egozentrischem Bezugssystem gezogen werden können und umgekehrt. Um entsprechende Versuche in kontrollierten Laborumgebungen durchführen zu können, werden in vielen Arbeiten virtuelle Umgebungen verwendet, so dass sich ein Proband nicht selbst, sondern

2. Ortsbezogene Dienste in Gebäuden - Grundlagen und Definitionen

vor einem Monitor sitzend durch eine virtuelle Umgebung bewegt [97]. Dieses Vorgehen unterliegt zwar einigen Einschränkungen, jedoch zeigen Ruddle et al. [123], dass die in virtuellen Umgebungen erzeugten kognitiven Karten vergleichbar sind mit jenen, die in der realen Welt erlernt werden. Ein weiterer Punkt, der in verschiedenen Arbeiten untersucht wurde, ist die Frage, ob die Fähigkeiten der menschlichen Navigation abhängig vom Geschlecht sind. In einer Studie von Moffat et al. [101] wird untersucht, ob sich die Leistung von männlichen und weiblichen Probanden unterscheidet, wenn die Probanden mehrmals hintereinander in einem virtuellen Labyrinth ohne Landmarken den Ausgang suchen und sich dabei die Route merken sollen. Die Studie kommt zu dem Ergebnis, dass die männlichen Probanden signifikant schneller dem Ausgang finden. Astur et al. [9] betrachten das Verhalten in einem virtuellen Morris-Wasserlabyrinth und zeigen, dass Männer konsistent schneller zu der versteckten Plattform navigieren. Das Morris-Wasserlabyrinth (*Morris water task*) ist ein ursprünglich von Richard Morris entwickelter Versuchsaufbau zur Untersuchung des Verhaltens von Nagetieren. Es besteht aus einem runden, mit trübem Wasser gefüllten Becken, das seitlich mit Markierungen versehen ist (so genannten Hinweisreizen). Die Aufgabe in dem Experiment besteht darin, selbstständig eine unter der Wasseroberfläche befindliche, nicht sichtbare Plattform zu finden und sich deren räumliche Position zu merken. Jedoch müssen solche Studien auch vor dem Hintergrund zahlreicher anderer Erkenntnisse interpretiert werden. So weisen Sandstorm et al. [129] beispielsweise darauf hin, dass sich Frauen vorwiegend an Landmarken und weniger an geometrischen Hinweisen orientieren, wohingegen Männer geometrische Hinweise und Landmarken in gleichem Maße zu Rate ziehen.

Voraussetzungen für die erfolgreiche Suche eines Weges sind die Orientierung im Raum, sowie die Fähigkeiten, einem Weg auf einer kognitiven Karte zu folgen und das gewünschte Ziel zu erkennen. Herrmann et al. [57] unterscheiden dabei zwischen dem Wissen über Landmarken in der Umgebung sowie Wissen über Routen. Das Routenwissen im kognitiven Sinne bezeichnet – ähnlich wie in der Informatik – eine Abfolge von Aktionen, die notwendig sind, um einen bestimmten Weg zu gehen.

Redish [119] unterscheidet fünf verschiedene Strategien bei der Navigation und bezieht sich dabei vor allem auf die Versuche im Morris-Wasserlabyrinth:

1. Falls keine Informationen über das Ziel vorliegen, so muss rein zufällig gesucht werden (*Random-Navigation*).
2. Ein Hinweis ist sichtbar, in dessen Richtung das Ziel liegt (*Taxon-Navigation*).
3. Das Ziel kann durch eine immer gleiche Abfolge von Bewegungen erreicht werden (*Praxic-Navigation*).
4. Durch die Verkettung von sichtbaren Hinweisen, einer damit verbundenen Richtung und bekannten (Teil-)Abfolgen wird das Ziel erreicht (*Routen-Navigation*).
5. Wurde die Position des Ziels in Bezug auf eine Konstellation von verschiedenen Hinweisen erlernt, so ist eine Karte der Umgebung bekannt. Redish bezeichnet dies als *locale navigation*.

2. Ortsbezogene Dienste in Gebäuden - Grundlagen und Definitionen

All diese Arbeiten beziehen sich auf die neurologischen Hintergründe der Navigation, nicht jedoch auf die technischen Implikationen für die Realisierung von Navigationsanwendungen. Nachfolgend wird daher eine technische Klassifizierung von Navigationssystemen gegeben.

2.1.5.1. Klassifikation von Navigationssystemen

Bei der Entwicklung eines Indoor Navigationssystems gibt es zahlreiche mögliche Ausprägungen. Betrachtet man allgemeine Merkmale, anhand derer sich die Systeme unterscheiden und einordnen lassen, so fallen insbesondere folgende Eigenschaften auf (vgl. Ruppel und Gschwandtner [125]): die Mobilität der Anwendung, der Grad der Personalisierung, die Granularität der Navigationsanweisungen, die Art des Positionierungssystems, die Art der Routenberechnung, Dynamisches Verhalten und der Schutz der Privatsphäre. Diese werden im Folgenden genauer erläutert und sind dabei nicht notwendigerweise an die Verwendung eines mobilen Endgerätes gekoppelt, sondern können auch mittels im Gebäude befindlicher Komponenten realisiert werden und sind in der Art der Darstellung von Navigationsanweisungen nicht auf Karten oder Pfeile beschränkt.

1. **Mobilität der Anwendung:** Eine Indoor-Navigationsanwendung kann entweder auf einem mobilen Endgerät ausgeführt werden oder mittels stationärer, im Gebäude befindlicher Geräte realisiert werden. Stationäre Komponenten können dabei z.B. Bildschirme, Lichter, Lautsprecher oder andere Ein- und Ausgabegeräte sein. Auch sind gemischte Ansätze möglich, bei denen ein Benutzer beispielsweise allgemeine Informationen auf öffentlichen Bildschirmen an den Wänden und personalisierte Routing-Informationen auf seinem mobilen Endgerät angezeigt bekommt.
2. **Grad der Personalisierung:** In manchen Umgebungen wie z.B. Flughäfen oder Krankenhäusern kann es wünschenswert sein, dass entweder alle Personen oder bestimmte Personengruppen zu ein und demselben Ziel geleitet werden. Demgegenüber stehen persönliche Navigationsysteme, welche individuelle Routen berechnen und die persönlichen Präferenzen des Nutzers berücksichtigen.
3. **Granularität der Navigationsanweisungen:** Je nach Szenario können die Genauigkeit und Komplexität von Navigationsanweisungen variieren. Ausgehend von dem einfachen Anzeigen von Start- und Zielpunkt auf einer Karte können komplexe Navigationssysteme beispielsweise vollständige Wege auf einer Karte anzeigen, Anweisungen zum Erreichen des nächstgelegenen Wegpunktes durch Symbole oder akustische Signale geben, Landmarken zur besseren Orientierung des Benutzers mit einbeziehen oder auch kontextabhängig die Komplexität der Anweisungen anpassen. Umfangreichere Navigationsanweisungen benötigen dabei jedoch auch komplexere Umgebungsmodelle und unterliegen einer höheren Berechnungskomplexität.
4. **Positionierungssystem:** Hinsichtlich des verwendeten Positionierungsverfahrens ist es für Navigationssysteme entscheidend, wie schnell, wie genau und wie zuverlässig

2. Ortsbezogene Dienste in Gebäuden - Grundlagen und Definitionen

Positionen von Zielobjekten berechnet werden können (siehe 3.3). Bei mobilen Systemen spielt zusätzlich der Energieverbrauch auf dem Endgerät eine wichtige Rolle.

5. **Routenberechnung:** Die Berechnung von kürzesten Wegen und optimalen Routen durch ein Gebäude kann in komplexen Umgebungen ein rechenintensiver Prozess sein und dadurch eine Server-seitige Berechnung erfordern. Gleichzeitig können mit einem einfachen Umgebungsmodell und effizienten Algorithmen auf einem mobilen Endgerät schnell lokale Navigationsanweisungen erzeugt werden. Je nach Anwendung sind daher sowohl endgerätbasierte, netzbasierte als auch gemischte Routenberechnungen möglich.
6. **Dynamisches Verhalten:** Der für einen Benutzer kürzeste Weg durch ein Gebäude kann sich im Laufe der Zeit verändern, wenn beispielsweise ein bestimmter Aufzug ausfällt oder wenn sich auf einem Weg mit mehreren Stationen an einer Station die Länge einer Warteschlange ändert (z.B. bei einem Sicherheits-Check am Flughafen). Dynamische Navigationssysteme können solche Informationen bei der Wegewahl berücksichtigen und so einen Benutzer gegebenenfalls auf einen anderen Weg umleiten.
7. **Location Privacy:** Wie bei allen LBS ist der Schutz der Privatsphäre in Navigationssystemen ein wichtiger Aspekt. Dazu zählt insbesondere, welche Komponenten und Akteure Zugriff auf die Ortsinformationen eines Zielobjekts haben. Im Falle eines Benutzers der sich mit Hilfe eines mobilen Endgerätes durch ein Gebäude navigieren lässt, wird im Idealfall sowohl Positionierung als auch Wegewahl vollständig autonom auf dem Endgerät durchgeführt, dadurch ist es für Dritte nicht möglich, die Bewegung des Benutzers zu verfolgen. Hybride Ansätze erlauben beispielsweise ein dynamisches Nachladen von Positions- und Routing-Daten, wodurch jedoch gleichzeitig auch ein Teil von Privatsphäre für den Benutzer verloren geht. Im Falle eines vollständig netzbasierten Ansatzes muss durch geeignete technische Schutzmaßnahmen sichergestellt werden, dass die Ortsinformationen nicht an unberechtigte Dritte gelangen.

Überträgt man diese Klassifikation auf das in Abbildung 2.1 vorgestellte Rollenmodell, so ergeben sich für den Location Model Provider zusätzliche Anforderungen hinsichtlich der möglichen Anfragen (siehe Abschnitt 2.1.2):

- **A7:** Für einen kürzesten Weg zwischen zwei Orten A und B muss eine Liste von Wegpunkten bestimmt werden können, die den Verlauf des Weges von A nach B möglichst genau widerspiegeln.
- **A8:** Bei der Berechnung von Routen müssen Benutzer-spezifische Präferenzen berücksichtigt werden können. Dazu zählen beispielsweise die Vermeidung von Treppen, die Berücksichtigung einer minimalen Durchgangsbreite oder auch bestimmte Zugangsrechte des Zielobjekts.

2. Ortsbezogene Dienste in Gebäuden - Grundlagen und Definitionen

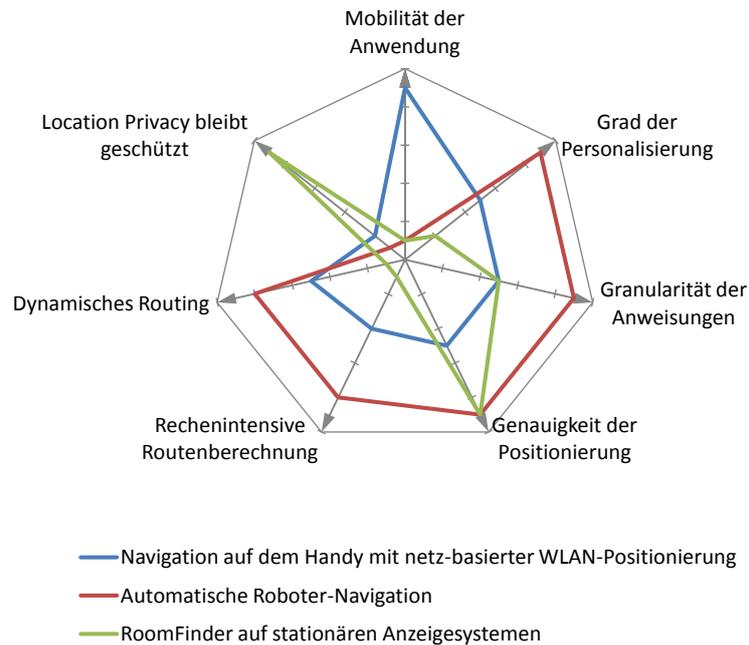


Abbildung 2.3.: Beispiele für Navigationssysteme und die Ausprägung ihrer Merkmale.

- **A9:** Die Granularität von berechneten Wegpunkten muss an die Anforderungen eines bestimmten Dienstes angepasst werden können.

Abbildung 2.3 zeigt ein Beispiel, in dem für drei verschiedene Szenarien die jeweiligen Ausprägungen der Merkmale dargestellt sind: Eine Navigationsanwendung auf einem Smartphone mit netzbasierter WLAN-Positionierung weist dementsprechend eine hohe Mobilität auf, jedoch hat der Benutzer geringe Kontrolle über den Zugriff auf seine Positionsdaten. Eine Anwendung zur Suche von Räumen über ein stationäres Anzeigesystem bietet zwar die Möglichkeit der anonymen Nutzung, jedoch sind die Navigationsanweisungen grundsätzlich nur in geringem Maße personalisiert. Gleichzeitig ist jedoch eine genaue Positionsbestimmung anhand der Standorte der Anzeige möglich. Eine automatische Roboter-Navigation, die sowohl terminal- als auch netzbasierte Positionierung einsetzt und dabei zentral gesteuert wird, kann dementsprechend sehr dynamisch reagieren und beinhaltet sehr fein aufgelöste Anweisungen. Ein weiteres Merkmal, anhand dessen sich Navigationssysteme klassifizieren lassen, ist die Unterscheidung zwischen zwei- und dreidimensionaler Navigation. Dazwischen existiert die Möglichkeit, die Berechnung auf 2D-Datenmodellen durchzuführen und anschließend innerhalb eines 3D-Modells zu visualisieren.

Ein Überblick über bestehende Systeme und verwandte Arbeiten der Indoor Navigation wird im Kapitel 5 gegeben.

2.2. Problemstellung und Ziel dieser Arbeit

Auf der Grundlage der vorgestellten Begriffe und Definitionen kann nun die Problemstellung dieser Arbeit präzisiert werden. Diese Arbeit behandelt die Bereitstellung von Umgebungsmodellen und Navigationsdaten für ortsbezogene Dienste in Gebäuden. Darunter fällt zunächst die Frage, wie und auf welcher Grundlage ein Umgebungsmodell erzeugt werden kann und welche Eigenschaften ein Modell unter Berücksichtigung der Anforderungen zukünftiger Dienste aufweisen muss. Zweitens bedarf es einer geeigneten Form zur Speicherung, so dass verschiedene Dienste die Umgebungsinformationen direkt nutzen und verarbeiten können. Eine dritte Herausforderung ist die Navigation in Gebäuden, denn sie erfordert unter anderem eine Berechnung von Navigationsanweisungen und -hinweisen, die einen Benutzer zu seinem Ziel leiten.

Erzeugung von Umgebungsmodellen Bei der Erzeugung eines Umgebungsmodells besteht die Herausforderung darin, die verschiedenen Elemente und Eigenschaften des Modells so flexibel zu gestalten, dass sie von unterschiedlichen Arten von Diensten und in Kombination mit unterschiedlichen Positionierungsverfahren genutzt werden können. Dabei müssen verschiedene Faktoren berücksichtigt werden:

- Da ein Umgebungsmodell durch unterschiedliche Dienste verwendet werden kann, muss es unabhängig vom verwendeten Positionierungsverfahren sein. Für die verschiedenen Positionierungsverfahren stellt sich die Herausforderung, diese effektiv und effizient miteinander zu kombinieren.
- Ein zentrales Element in einem Umgebungsmodell ist das *Wegenetz*. Es beschreibt die *Begehbarkeit* der Umgebung und charakterisiert damit den genauen Verlauf von Wegen durch Räume, Stockwerke und ganze Gebäude. Dadurch können beispielsweise Orte und Wege anhand der *gehbaren* Entfernung gefunden werden oder auch Navigationsanweisungen erzeugt werden. Die Herausforderung bei der Erzeugung eines Wegenetzes besteht darin, die tatsächliche Begehbarkeit möglichst detailliert abzubilden und gleichzeitig weitere Eigenschaften wie Komplexität, Weglängen oder Abdeckung zu optimieren.
- Für die Erzeugung eines Umgebungsmodells besteht die Herausforderung darin, alle für die Dienste relevanten Eigenschaften explizit abzubilden, d.h. dass beispielsweise der Übergang zwischen zwei verbundenen Räumen dargestellt wird und nicht nur implizit durch die Nachbarschaft der beiden Raumpolygone gegeben ist. Als Datengrundlage können hier elektronische Baupläne dienen, aus denen vor allem die Lage und Form von Räumen, Stockwerken etc. gewonnen werden kann.
- Auf mobilen Endgeräten besteht das Problem, dass mehrere Anwendungen parallel auf Umgebungsinformationen zugreifen wollen beziehungsweise eine einzelne Anwendung nur einen kleinen Ausschnitt des Modells zur Ausführung benötigt.

2. Ortsbezogene Dienste in Gebäuden - Grundlagen und Definitionen

Speicherung Am Ende des Erzeugungsprozesses steht die Speicherung und Bereitstellung des Modells für die verschiedenen Dienste. Dazu wird ein geeignetes Format benötigt, in dem sich alle im Modell enthaltenen Informationen abbilden lassen.

- Eine Herausforderung bei der Speicherung eines Umgebungsmodells besteht darin, dass es für unterschiedlichste Arten von Diensten zugänglich gemacht wird. Gleichzeitig ist die Erweiterbarkeit des Modells ein wichtiges Kriterium, um es einem Dienst zu ermöglichen, das Modell unter Umständen mit eigenen Informationen anzureichern.
- Besonders das Wegenetz eines Gebäudes kann sehr unterschiedliche Formen annehmen, und für unterschiedliche Arten von Diensten werden auch unterschiedliche Wegenetze. Daher müssen verschiedene Varianten des Wegenetzes gespeichert werden können.

Navigationsdaten Vielen I-LBS benötigen zur Ausführung nicht nur Positions- sondern auch Navigationsdaten. Navigationsdaten basieren auf den Eigenschaften von kürzesten Wegen und deren Begehbarkeit und können zusätzlich mit Informationen über Landmarken kombiniert werden, um so verständliche und anschauliche Navigationsanweisungen zu generieren. Die so gewonnenen Informationen und Anweisungen können anschließend sowohl auf stationären Anzeigen, als auch auf mobilen Endgeräten verwendet werden.

- Ein Problem stellt die Berechnung der Navigationsanweisungen dar, wenn die Art des zugrunde liegenden Wegenetzes nicht im Vorhinein bekannt ist. In diesem Fall kann man nicht von einer regelmäßigen Struktur des Wegenetzes ausgehen und eine Sequenz von aufeinander folgenden Navigationsanweisungen kann nicht eins zu eins aus dem Wegenetz generiert werden.
- Eine weitere Herausforderung stellen Landmarken dar, diese bieten dem Benutzer eine zusätzliche Orientierung. Um jedoch Landmarken in Navigationsanweisungen mit einzubeziehen, müssen geeignete Kriterien und Regeln bekannt sein, nach denen eine Landmarke verwendet werden kann.
- Navigation auf mobilen Endgeräten stellt einige zusätzliche Anforderungen an die Bereitstellung der Navigationsdaten. Insbesondere, wenn ein Benutzer spontan und anonym einen Navigationsdienst auf seinem mobilen Endgerät nutzen will, besteht die Herausforderung darin, alle Daten im Voraus bereit zu stellen und eine anonyme Positionierung zu ermöglichen.

Die vorliegende Arbeit entwickelt Konzepte und Lösungen für die genannten Probleme. Das folgende Kapitel beschäftigt sich zunächst mit den allgemeinen Eigenschaften und Anforderungen von Umgebungsmodellen und geht anschließend näher auf Positionierungsverfahren, Wegenetze und die Bereitstellung von Umgebungsmodellen ein. Für Positionierungsverfahren wird ein neuer Ansatz zur Energie-effizienten Kombination vorgestellt, und es wird ein neues Konzept für die Erzeugung von Wegenetzen und Umgebungsmodellen

2. Ortsbezogene Dienste in Gebäuden - Grundlagen und Definitionen

entwickelt sowie empirisch untersucht. Kapitel 4 befasst sich mit dem Problem der Speicherung von Umgebungsmodellen. Dazu wird ein neues Modell vorgestellt, das als Anwendungsschema für die *Geography Markup Language* umgesetzt wird. Herausforderungen und Lösungen für die Erzeugung von Navigationsanweisungen werden in Kapitel 5 behandelt und anhand einer prototypischen Umsetzung diskutiert.

2.3. Zusammenfassung

In diesem Kapitel wurden zunächst grundlegende Begriffe und Definitionen eingeführt, die für ortsbezogene Dienste innerhalb von Gebäuden relevant sind. Anhand eines Rollenmodells wurden die unterschiedlichen Aufgaben für die Erbringung von Indoor Location-based Services identifiziert. Im Rahmen dieser Arbeit stehen die beiden Rollen Umgebungsmodell-Erzeuger und Umgebungsmodell-Anbieter im Zentrum. Dabei werden die Anforderungen an ein Umgebungsmodell letztendlich durch die möglichen Arten von Anfragen an einen Umgebungsmodell-Anbieter durch einen Dienst beeinflusst. Ausgehend von den vorgestellten Begriffen wurden anschließend die Problemstellung und das Ziel dieser Arbeit formuliert.

3. Bereitstellung von Umgebungsmodellen für I-LBS

Eine zentrale Frage bei der Realisierung zukünftiger ortsbezogener Dienste in Gebäuden ist die Frage nach geeigneten Mitteln und Wegen zur Bereitstellung von Umgebungsmodellen. Am Ende eines solchen Bereitstellungsprozesses steht der Umgebungsmodell-Anbieter, der mit Hilfe des erzeugten Umgebungsmodells entsprechende Anfragen des I-LBS Providers (oder auch Location Providers oder Positionsermittlers) beantworten kann. Diese Schnittstelle des Umgebungsmodell-Anbieters bestimmt letztendlich die grundlegenden Anforderungen an die im Umgebungsmodell benötigten Informationen. Gleichzeitig stellt sich dabei die Frage, auf welche Art und Weise der Umgebungsmodell-Anbieter ein Umgebungsmodell vorhält bzw. von einem Umgebungsmodell-Erzeuger entgegen nimmt. Beispielsweise könnte ein Umgebungsmodell lediglich Raumpolygone und Verbindungen zwischen Räumen modellieren. Dies hätte auf der einen Seite den Vorteil eines schlanken und kleinen Modells, auf der anderen Seite müsste der Umgebungsmodell-Anbieter zusätzliche Routing-Algorithmen realisieren, um in dem Modell eine Anfrage nach einem kürzesten Weg zu beantworten. In einem anderen Fall könnte das Umgebungsmodell zusätzlich ein vom Umgebungsmodell-Erzeuger berechnetes Wegenetz enthalten, in dem alle begehbaren Wege eines Gebäudes explizit als fertiger Graph gespeichert sind. Ein solches Modell wäre zwar größer und komplexer, hätte jedoch den Vorteil, dass mehrere Umgebungsmodell-Anbieter mit vergleichsweise geringem Aufwand sofort eine Routing-Funktionalität anbieten können.

Betrachtet man die Rolle des Umgebungsmodell-Erzeugers, so lassen sich zwei große Herausforderungen erkennen: Zum einen müssen die relevanten Informationen eines Ortes (z.B. Form, Lage oder Verbundenheit mit anderen Orten) erfasst und geeignet gespeichert werden. Zum anderen erweist sich die Erstellung eines Umgebungsmodells, insbesondere dessen Wegegraphen, gerade bei großen Gebäuden als eine sehr zeit- und damit kostenintensive Aufgabe. Der Prozess der Erzeugung sollte daher *soweit wie möglich automatisiert* ablaufen. In vielen Organisationen sind heutzutage digitale Gebäudepläne z.B. in Form von CAD-Dateien vorhanden und können entsprechend als Datengrundlage genutzt werden. Die Formate dieser Daten sind jedoch vorrangig für die Verwendung durch Architekten und für den Bau des Gebäudes konzipiert, so dass gerade Informationen über die Begehbarkeit eines Gebäudes nur implizit enthalten sind. Liegt ein Gebäudeplan mit den in diesem Kapitel beschriebenen Mindestanforderungen vor, so kann mit Hilfe verschiedener Algorithmen ein Wegenetz automatisch erzeugt werden. Bei der Erzeugung von Wegenetzen sind verschiedene Aspekte zu berücksichtigen, beispielsweise sollte die Abdeckung des Wegenetzes im Gebäude möglichst hoch sein. Gleichzeitig sollte aber auch der Verzweigungsgrad bzw. die Komplexität des Wegenetzes möglichst gering sein, um so den Berechnungsauf-

3. Bereitstellung von Umgebungsmodellen für I-LBS

wand für die Wegeberechnung zu minimieren. Neben einem Wegenetz sind für viele I-LBS auch die *Lage* und *Form* von Räumen relevant, diese müssen ebenfalls aus den digitalen Gebäudeplänen gewonnen werden. Da auch hier die Informationen in den CAD-Daten ursprünglich für andere Zwecke erstellt wurden, müssen viele Polygone z.B. erst „poliert“ werden, d.h. überflüssige Polygoneckpunkte müssen entfernt werden. Auch runde Formen können für viele Algorithmen zur Erzeugung von Wegenetzen ein Problem darstellen und müssen gegebenenfalls in neue Polygoneckpunkte umgewandelt werden.

In diesem Kapitel werden zunächst die verschiedenen Arten von Umgebungsmodellen betrachtet. Diese Klassifikation bildet die Grundlage zur Formulierung grundlegender Anforderungen an ein Umgebungsmodell für zukünftige I-LBS. Unabhängig vom Umgebungsmodell haben auch die Positionsdaten aus den verschiedenen möglichen Positionierungsverfahren Auswirkungen auf die Funktionalität der beteiligten I-LBS und werden entsprechend vorgestellt. Anschließend erfolgt eine Diskussion verschiedener Ansätze und Algorithmen zur Erzeugung von Wegenetzen, wie z.B. der Corner Graph, die Konvexe Partitionierung oder auch verschiedene Triangulierungen. Abschließend wird ein Konzept zur automatisierten Erzeugung eines Umgebungsmodells vorgestellt.

3.1. Klassifikation von Umgebungsmodellen und verwandte Arbeiten

Im Umfeld der geographischen Informationssysteme wurden in den vergangenen Jahrzehnten viele Ansätze entwickelt, um die räumlichen Beziehungen zwischen Objekten zu kennzeichnen und auszuwerten. Zur Modellierung der Umgebung für mobile Roboter unterscheiden bereits Chatila und Laumond [28] drei verschiedene Arten von *world models*: geometrische, topologische und semantische Modelle. Dabei wird der Begriff des Ortes folgendermaßen definiert:

„Let us define the concept of *place* as an area that is a functional or topological unit. For example, a room, a corridor are topological units. The concept of place is defined at several levels: a room, a storey, or a whole building are places.“
[28]

White [150] beschreibt in seinem 1984 erschienenen Artikel die grundlegenden Anforderungen für ein „*Multipurpose Geographic Data System*“: In Bezug auf die geometrischen Elemente von digitalen Karten geht er dabei vor allem auf die *topologischen* Eigenschaften ein, d.h. die Relationen zwischen *0-cells* (Punkten), *1-cells* (Liniensegmente beliebiger Form) und *2-cells* (Flächen), siehe auch Corbett [32]. Viele geographische Schemata sind hierarchisch aufgebaut (z.B. ein Park, der innerhalb einer Stadt liegt, die wiederum von einem Land umschlossen ist). Diese Inklusion stellt für eine Menge M von geographischen Orten eine *partielle Ordnung* dar, denn sie ist

- *reflexiv*, da jede Fläche A in sich selbst enthalten ist ($A \leq A$),

3. Bereitstellung von Umgebungsmodellen für I-LBS

- *transitiv*, da für drei Flächen A, B und C gilt: $A \leq B \leq C \Rightarrow A \leq C$,
- und *antisymmetrisch*, da gilt: $(A \leq B) \wedge (B \leq A) \Rightarrow A = B$.

Manche Elemente stehen miteinander in Relation (der Park liegt vollständig innerhalb der Stadt), jedoch nicht notwendigerweise alle Elemente (die Stadt München schließt weder die Stadt Heidelberg ein, noch wird sie von Heidelberg eingeschlossen). Allerdings reicht diese rein hierarchische Betrachtung nicht aus, um alle Zusammenhänge abzubilden. Ein Beispiel dafür wäre ein Park, durch den eine Landesgrenze verläuft. Die beiden Hälften des Parks tauchen nicht in der Hierarchie auf.

Kainz et al. [69] greifen dieses Konzept der partiell geordneten Mengen auf und gehen näher darauf ein, wie sich die räumlichen Zusammenhänge besser durch einen *Verbund* (*lattice*) darstellen lassen. Ein Verbund ist eine partielle Ordnung, in der es zu je zwei Elementen immer ein Supremum und ein Infimum gibt. Dadurch wird gewährleistet, dass es für alle Elemente in einem Verbund von geographischen Orten eine gemeinsame obere Schranke gibt („überall“) und gleichzeitig für alle Elemente eine größte untere Schranke („nirgendwo“) existiert.

Die Unterscheidung zwischen der Modellierung von Koordinaten (Geometrie) sowie deren Verbund (Topologie) findet sich beispielsweise auch in den Arbeiten von Bauer et al. [13], Ranganathan et al. [117], Hu und Lee [61] oder Jiang und Steenkiste [66]. Unter dem Begriff des *symbolischen Umgebungsmodells* werden verschiedene Eigenschaften zusammengefasst, vor allem jedoch, dass *symbolische Koordinaten* zum Einsatz kommen. So unterscheidet Leonhardt [92] zwischen einfachen (möglicherweise überlappenden), exklusiven (überlappungsfreien) und baumförmigen Modellen mit einer Inklusionsbeziehung zwischen Orten, sowie azyklischen Ortsgraphen. Kolodziej und Danado [82] bezeichnen eine symbolische Karte auch als *spatial model graph* oder *spatial tree*. Roth [121] verwendet anstelle des symbolischen Modells den Begriff des *semantischen Modells*, dieses Synonym taucht auch in weiteren verwandten Arbeiten auf.

Becker und Dürr [15] unterteilen die symbolischen Umgebungsmodelle in mengenbasierte (*set-based*), hierarchische und graphbasierte Modelle. Die *hybriden Umgebungsmodelle* vereinen die Vorzüge der geometrischen und symbolischen Modelle. Auf diese Art und Weise können beispielsweise die Räume eines Gebäudes in einem Verbund mittels symbolischer Koordinaten organisiert werden und zusätzlich auch die Geometrie, d.h. geometrische Koordinaten, gespeichert werden. Nachfolgend wird näher auf die Eigenschaften von geometrischen, symbolischen und hybriden Modellen eingegangen.

Geometrische Umgebungsmodelle Ein *geometrisches* Umgebungsmodell beschreibt die Eigenschaften eines Ortes in einem zwei- oder dreidimensionalen Koordinatensystem durch Punkte, Linien, Flächen oder Körper. Dadurch kann zwar eine Enthaltensein-Beziehung zwischen zwei Objekten ausgedrückt werden, jedoch noch keine topologische Relation zwischen zwei Orten. Zur Unterscheidung zweier übereinander liegender Orte (z.B. zwei Räume auf unterschiedlichen Stockwerken) können entweder dreidimensionale Koordinaten oder unterschiedliche lokale Koordinatensysteme verwendet werden, was wiederum Informationen

3. Bereitstellung von Umgebungsmodellen für I-LBS

über die Ausrichtung der einzelnen Koordinatensysteme sowie die nötigen Transformationen zwischen den Systemen erfordert. Die Speicherung von geometrischen Gebilden hat den Vorteil, dass diese unmittelbar zur Visualisierung auf einer Karte verwendet werden können. Da viele Indoor-Positionierungsverfahren die Aufenthaltsorte von Zielobjekten als Koordinaten in einem lokalen Koordinatensystem liefern, kann in einem geometrischen Umgebungsmodell durch eine Punkt-in-Polygon-Abfrage zu einer gegebenen Koordinate schnell der Bezeichner, z.B. Name des Raumes, ermittelt werden.

Symbolische Umgebungsmodelle *Symbolische* Umgebungsmodelle verwenden abstrakte Bezeichner zur Beschreibung eines Ortes, z.B. „Raum G003“ oder „Erdgeschoss“. Diese symbolischen Koordinaten sagen zunächst nichts über die geometrische Form des Ortes aus, es werden lediglich Beziehungen zwischen den Orten ausgedrückt. Im Wesentlichen werden dabei Enthaltensein, Überlappung und Disjunktheit von Orten betrachtet. Je nachdem, ob die symbolischen Bezeichner als Menge oder als Knoten in einem Graphen aufgefasst werden, lassen sich später unterschiedliche Arten von Anfragen beantworten.

- **Mengenbasierte Umgebungsmodelle:** In einem mengenbasierten Modell werden ausgehend von einer Menge L symbolischer Koordinaten die einzelne Orte durch Teilmengen von L ausgedrückt. Zwei Orte L_1 und L_2 ($L_1, L_2 \subseteq L$) überlappen sich wenn $L_1 \cap L_2 \neq \emptyset$. Gilt $L_1 \cap L_2 = L_1$, dann ist der Ort L_1 in L_2 enthalten. Einfache symbolische bzw. mengenbasierte Modelle kommen in vielen verschiedenen Bereichen zum Einsatz. Beispielsweise werden die Zellen eines GSM-Mobilfunknetzes über einen symbolischen Bezeichner (location area identifier + cell id) identifiziert, sie können sich jedoch mit anderen Zellen überlappen. Wird innerhalb eines Gebäudes der Basic Service Set Identifier (BSSID) eines ein Reichweite befindlichen drahtlosen lokalen Netzes zur Positionsangabe verwendet, so entspricht dies ebenfalls einem einfachen symbolischen Bezeichner. Auch Brumitt und Shafer [21] geben ein Beispiel für ein einfaches mengenbasiertes Modell.

Ist die Menge der Orte nicht überlappungsfrei, aber hierarchisch angeordnet bzw. es existiert eine partielle Ordnung der Symbole, so entsteht ein azyklischer Ortsgraph (*acyclic location graph*), siehe auch Domnitcheva [40]. Ist die Menge der Orte überlappungsfrei (*exklusives Modell*), und es existiert zusätzlich eine partielle Ordnung der Symbole, so entsteht ein baumförmiges Modell (*location tree*). Anders ausgedrückt erweitert ein baumförmiges Modell das exklusive Modell um eine Hierarchie.

Die Hierarchie der Orte in einem Verbund lässt sich als Hasse-Diagramm visualisieren, wie auf der linken Seite in Abbildung 3.1 gezeigt. Rechts daneben ist die äquivalente Repräsentation als Menge dargestellt.

- **Graphbasierte Umgebungsmodelle:** In einem graphbasierten Modell werden die Beziehungen zwischen symbolischen Koordinaten durch einen Graphen $G = (V, E)$ ausgedrückt. Die symbolischen Koordinaten bilden die Menge V der Knoten und eine Kante $e = (v_i, v_j) \{e \in E; v_i, v_j \in V\}$ drückt aus, dass zwischen v_i und v_j eine

3. Bereitstellung von Umgebungsmodellen für I-LBS

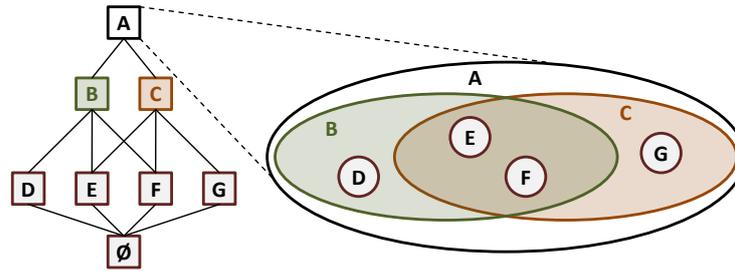


Abbildung 3.1.: Hierarchie von Orten. Links: Hesse-Diagramm. Das unterste \emptyset -Symbol entspricht dem Infimum (untere Grenze) des Verbandes. Rechts: Darstellung als Menge.

direkte Verbindung besteht. Zusätzlich können die Knoten und Kanten gewichtet, gefärbt oder mit zusätzlichen Eigenschaften versehen werden um beispielsweise die euklidische, topologische oder begehbare Entfernung (walking distance) zwischen zwei Orten auszudrücken.

Beispielsweise werden in dem von Hu und Lee [61] vorgestellten Modelle die jeweiligen Ausgänge (*exits*) von unmittelbar miteinander verbundenen Orten betrachtet und eine *Exit-Hierarchie* gebildet. Ein Pfad zwischen zwei entfernten Orten besteht dann aus einer Sequenz von *exits* (siehe auch Li und Lee [93]).

Die Knoten eines Graphen können zum einen einzelne Räume repräsentieren, wie in Abbildung 3.2a dargestellt. Zum anderen kann auch jedem Übergang eines Raumes ein eigener Knoten zugeordnet werden (siehe Abbildung 3.2b). Im zweiten Fall verändern sich die Pfadlängen entsprechend und geben die reale Entfernung zwischen Orten genauer an. Dieser zweite Graph lässt sich auch als Verfeinerung des Ersten auffassen und es gibt zahlreiche Möglichkeiten, durch Hinzunahme weiterer Knoten den Graphen zu erweitern. Auf diese Art und Weise können auch *hierarchische Graphen* gebildet werden: Auf verschiedenen Ebenen angeordnet repräsentieren die Knoten der Ebene n jeweils eine Generalisierung einer Menge von Knoten, die auf der darunter liegenden Ebene $n + 1$ liegen. In Abbildung 3.2a würde der Kante von B nach F dann ein Gewicht entsprechend der Länge des kürzesten Weges von B nach F zugeordnet werden. Dieser Wert ließe sich wiederum anhand des verfeinerten Graphen in Abbildung 3.2b bestimmen und wäre in diesem Fall (wenn man den Mittelpunkt von B als Startpunkt voraussetzt) gleich der Länge des Weges, der über E nach F führt. An diesem Beispiel wird auch gleichzeitig deutlich, dass jede Generalisierung ihr Vor- und Nachteile hat: Würde man nicht den Mittelpunkt eines Raumes als Startpunkt verwenden, so würde der kürzeste Weg von B nach F nicht über E führen, sondern direkt von B nach F . Dennoch bieten hierarchische Graphen viele Vorteile, besonders hinsichtlich der Beschleunigung von Pfadberechnungen.

Hybride Umgebungsmodelle Kombiniert man die Eigenschaften symbolischer und geometrischer Modelle, so entstehen hybride Umgebungsmodelle. Sie beinhalten sowohl Infor-

3. Bereitstellung von Umgebungsmodellen für I-LBS

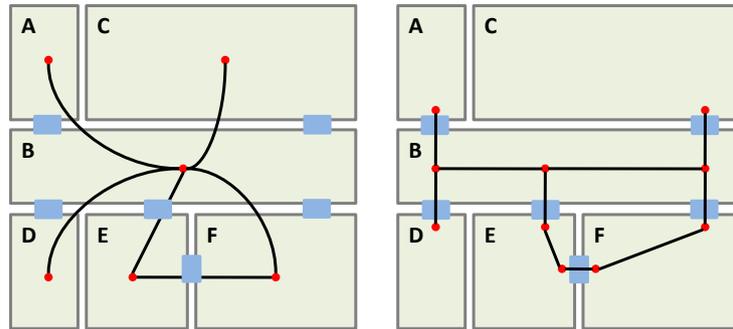


Abbildung 3.2.: Zwei Varianten eines einfachen graphbasierten Modells. a) Knoten repräsentieren ganze Räume. b) Jeder Knoten repräsentiert einen bestimmten Übergang. Blaue Rechtecke kennzeichnen Übergänge zwischen Räumen.

mationen über die geometrische Form und Lage von Orten als auch über die Hierarchie bzw. Beziehung und Verbundenheit der Orte untereinander. Dadurch lassen sich die Vorteile der beiden verschiedenen Ansätze zu einem für viele Anwendungen wesentlich verbesserten Modell vereinen. Indem man die symbolischen Bezeichner und Koordinaten den entsprechenden Positionen im geometrischen Teil des Modells zuordnet, kann so einerseits das Enthaltensein einer symbolischen Koordinate in einem Raumpolygon einfach überprüft werden, andererseits kann zur Berechnung eines kürzesten Weges der graphbasierte Teil des Modells herangezogen werden.

So kombiniert z.B. das von Jiang und Steenkiste [66] vorgestellte Modell eine hierarchische Anordnung der Orte mit lokalen Koordinatensystemen. Dazu wird die Umgebung zunächst in verschiedene Unterräume (*sub-spaces*) unterteilt, die jeweils einen bestimmten Bereich repräsentieren, beispielsweise einen Gebäudetrakt, einen Saal oder ein Zimmer. Auf diese Art und Weise entsteht ein Baum, in dem sich die Unterräume anhand des gegenseitigen Enthaltenseins anordnen lassen. Jedem Unterraum ist zusätzlich ein lokales Koordinatensystem zugeordnet, so dass eine Position innerhalb des Gebäudes eindeutig anhand des Namens des Unterraumes und der Koordinate im lokalen Koordinatensystem angegeben werden kann. Ähnliche Ansätze finden sich auch in Beigl et al. [18], Dürr und Rothermel [41], Ranganathan et al. [117] oder Stevenson et al. [137].

Eine Kombination aus Verbund und Graph wird von Ye et al. [157] beschrieben. Der Verbund wird darin verwendet, um Enthaltensein, Überlappung und Disjunktheit von Orten zu modellieren, und der Graph dient der Speicherung von Adjazenz (Nachbarschaft) und Zusammenhang. Jedoch wird dabei nicht auf den Aspekt der Begehbarkeit oder der Sichtbarkeit zwischen Orten eingegangen.

Hybride Umgebungsmodelle bieten darüber hinaus weitere interessante Möglichkeiten: durch die Kombination von geometrischen und topologischen Informationen können z.B. Wege berechnet werden, die eine bestimmte Breite oder Höhe nicht unterschreiten, oder Navigationsanweisungen können unter Berücksichtigung der Sichtbarkeit von bestimmten Landmarken erzeugt werden.

3. Bereitstellung von Umgebungsmodellen für I-LBS

Im folgenden Abschnitt werden Anforderungen an ein umfassendes Umgebungsmodell für zukünftige I-LBS definiert. Dabei sind die wichtigsten Unterschiede zu den oben diskutierten bestehenden Ansätzen die Forderung nach der Austauschbarkeit des Wegenetzes und der Modellierung der Begehbarkeit, sowie die Bereitstellung von Navigationsdaten.

3.2. Anforderungen an ein Umgebungsmodell für I-LBS

Betrachtet man die bisherige generelle Entwicklung von ortsbezogenen Diensten, so lassen sich verschiedene Faktoren identifizieren, die die Verbreitung von LBS zum Teil wesentlich begünstigt haben [19]: Während die erste Generation von LBS einen überwiegend reaktiven Charakter hatte und die Interaktion zwischen Benutzer und Dienst zumeist synchron (pull-basiert) erfolgte, so nimmt bei neueren Diensten die Proaktivität zu, und Benutzer werden mehr und mehr automatisch (push-basiert) über die für sie interessanten ortsbezogenen Ereignisse informiert. Beispiele dafür sind proaktive Benachrichtigungen über POIs und Angebote in der Nähe des Zielobjekts bzw. Nutzers oder auch asynchrone Status-Änderungen von Ortsinformationen mehrerer mobiler Zielobjekte. Zweitens ändert sich der Bezug zwischen verschiedenen LBS-Benutzern, indem die Dienste nicht mehr nur selbst-, sondern auch querverweisend operieren und damit auch jeweils andere Benutzer einbinden. Drittens lässt sich ein Wandel beobachten von Diensten, die *einzelne Zielobjekte* betrachten (single target services) hin zu Diensten, die *die Positionen vieler verschiedener Zielobjekte* verarbeiten und miteinander in Bezug setzen (multi target services). Dies geht einher mit einer zunehmenden Verfügbarkeit dynamischer Informationen, so dass längst nicht mehr nur rein statische Ortsinformationen berücksichtigt werden. Ein weiterer Faktor ist schließlich, dass nicht mehr nur der *Dienst-Betreiber* Informationen bereit stellt, sondern auch die *Benutzer* mehr und mehr Inhalte generieren und untereinander gegenseitig zur Verfügung stellen. Die Dienste müssen dabei mit sehr unterschiedlichen Angaben zu einem Ort umgehen können, da der Ort auf viele Arten und Weisen interpretiert werden kann. Dazu gibt beispielsweise Dobson [39] eine Taxonomie für den Begriff der *location* im *Pervasive Computing* an und unterscheidet insgesamt 18 Arten: Unter die *bekannt*en Orte fällt zum Beispiel eine absolute Angabe in Form einer GPS-Koordinate, eines benannten Bereiches (*im Raum G003*) oder einer benannten Klasse von Bereichen (*in einem Besprechungsraum*). Relative Angaben können sich entweder auf das Zielobjekt selbst (*in seinem Büro*) oder auch auf andere Zielobjekte beziehen (*unterwegs mit Bob*). Die *ungefähren* Angaben kennzeichnen einen Aufenthaltsort beispielsweise anhand von *in der Nähe des Eingangs, auf dem Weg zur Kantine* oder durch temporale Ausdrücke wie *gegen 10 Uhr im Besprechungsraum*. Aber auch *negierte* Aussagen können einen Aufenthaltsort kennzeichnen, wenn eine Person z.B. *nicht im Büro* ist. Zuletzt muss auch immer davon ausgegangen werden, dass für einen bestimmten Dienst der Aufenthaltsort einer Person temporär *nicht bekannt* sein kann.

Eine grundlegende Voraussetzung für alle LBS ist nach wie vor die Verfügbarkeit von geeigneten Positionierungsverfahren. Da heute nahezu jedes neue Mobiltelefon über einen eingebauten GPS-Empfänger verfügt, ist darüber eine entsprechend sehr genaue Ortung

3. Bereitstellung von Umgebungsmodellen für I-LBS

möglich. Gleichzeitig haben auch die Möglichkeiten der Zell-basierten Ortung über Mobilfunknetze bzw. drahtlose lokale Netze stark zugenommen. Positionierungsverfahren innerhalb von Gebäuden finden im Vergleich dazu eine bisher eher geringe Verbreitung. Dies liegt zum einen an den vergleichsweise hohen Investitionskosten zum Aufbau einer geeigneten Infrastruktur, zum anderen erfordern fast alle Indoor-Positionierungsverfahren einen entsprechend hohen Aufwand für den Betrieb. Ausgehend von dem im Jahr 2000 von Bahl und Padmanabhan [10] vorgestellten *RADAR-System* wurden jedoch zunehmend leistungsfähige und immer kostengünstigere Verfahren entwickelt, die insbesondere auf der Nutzung von heutzutage weitläufig verbreiteten drahtlosen lokalen Netzen basieren. So existieren mittlerweile mehrere kommerzielle Lösungen zur WLAN-basierten Lokalisierung mobiler Endgeräte innerhalb von Gebäuden und werden beispielsweise von AeroScout Inc., Eka-hau Inc., Cisco Systems Inc. oder Zebra Technologies Corp. angeboten. Betrachtet man die existierenden Ansätze genauer, so fällt auf, dass diese Systeme vorrangig zur reinen Positionsbestimmung bzw. Visualisierung von Positionsdaten (z.B. Darstellung einer Koordinate auf einer Karte) ausgelegt sind. Des Weiteren existieren bisher keine einheitlichen Schnittstellen zum Austausch von Positionsdaten zwischen den einzelnen Systemen, d.h. zur Integration eines solchen Systems ist ein entsprechend hoher Aufwand nötig.

Ein entscheidender Faktor zur Realisierung zukünftiger I-LBS ist daher die Verfügbarkeit von einheitlichen Schnittstellen für den Zugriff auf Indoor-Positionsdaten. Dazu bedarf es verschiedener Komponenten, die die Koordinaten aus den (zumeist lokalen) Koordinatensystemen der verschiedenen Positionierungsverfahren entsprechend transformieren. Ebenso wichtig ist die Verfügbarkeit symbolischer Bezeichner wie beispielsweise Raumnamen, Raumnummern, Stockwerke oder Abteilungen. Zur Vereinigung von Positionsdaten aus verschiedenen Positionierungssystemen in einem einheitlichen Umgebungsmodell müssen alle in dem Umgebungsmodell gespeicherten Informationen so abgebildet sein, dass es möglich ist, die zugehörigen Koordinaten in ein beliebiges anderes Koordinatensystem zu überführen.

Zwei wesentliche Anwendungsgebiete von I-LBS sind *Wegefindung* und *Navigation*. Beide erfordern umfassende Informationen über die Topologie eines Gebäudes sowie dessen Begehbarkeit und sind daher auf Umgebungsmodelle angewiesen. Dabei ist zu beachten, dass die Berechnung von Routen und Navigationsanweisungen je nach Anwendungsgebiet von unterschiedlichen Akteuren sowie unterschiedlichen Arten von (End-)Geräten durchgeführt werden. Eine grundlegende Anforderung an ein Umgebungsmodell für zukünftige I-LBS besteht daher darin, dass alle enthaltenen Informationen möglichst *explizit* abgespeichert werden und ohne zusätzliche bzw. aufwändige Berechnungsprozesse wieder aus dem Modell extrahiert werden können. Implizit enthaltene Informationen erschweren die Verwendung des Modells auf weniger leistungsfähigen Systemen und erfordern jeweils zusätzlichen Programm-Code.

Im Folgenden werden grundlegende *funktionale Anforderungen* an ein Umgebungsmodell für zukünftige I-LBS vorgestellt:

- **Abbildung von Positionsdaten:** Die von verschiedenen Indoor-Positionierungsverfahren bereitgestellten Positionsdaten müssen über eine entsprechende Schnittstelle

3. Bereitstellung von Umgebungsmodellen für I-LBS

in das Koordinatensystem des Umgebungsmodells transformiert werden können. D.h. es wird eine Funktion benötigt, die Positionsdaten aus einem anderen Referenzsystem in das Koordinatensystem des Umgebungsmodells transformiert und umgekehrt Positionen aus dem Umgebungsmodell in ein anderes Referenzsystem überführen kann. Viele der existierenden Positionierungsverfahren verwenden bereits lokale Koordinatensysteme, jedoch wird deren Nutzung oftmals dadurch erschwert, dass beispielsweise für jedes Stockwerk eines Gebäudes ein anderer Maßstab für den zugrunde liegenden Gebäudeplan verwendet wird. Dies ist zumeist darauf zurück zu führen, dass der Betreiber eines solchen Positionierungssystems bereits für die Kalibrierung Rasterbilder wie z.B. JPEG- oder PNG-Dateien verwenden muss, und diese Dateien nicht ohne Weiteres in identischen Auflösungen vorliegen. Viele Positionierungsverfahren (zumindest solche, die auf der Messung von Radiosignalen beruhen) bergen einen Meßfehler, d.h. dass die berechnete Position eines Zielobjekts von der tatsächlichen Position mehr oder minder abweicht. Bei nahezu allen gängigen Positionierungsverfahren ist dieser Meßfehler zusätzlich zeit- und ortsabhängig, so dass kein konstanter Fehler angenommen werden kann. Weicht nun die berechnete Position eines Zielobjekts zu sehr von der tatsächlichen ab, so stellt dies eine zusätzliche Herausforderung für die Abbildung der Positionsangabe in das verwendete Umgebungsmodell dar und muss entsprechend behandelt werden.

- **Abbildung von Orten von Interesse:** Die in einem Gebäude relevanten *Orte von Interesse* (points of interest, POI) müssen in dem Umgebungsmodell anhand ihrer Koordinaten sowie von Attributen identifizierbar sein. Entsprechend den Positionsdaten mobiler Objekte müssen auch POIs sowohl über geometrische als auch symbolische Koordinaten referenzierbar sein. Bei der Navigation durch ein Gebäude ist es zusätzlich auch relevant, die *Sichtbarkeit* von POIs zu berücksichtigen.
- **Abbildung von Bereichen von Interesse:** Im Gegensatz zu einem POI bezeichnet ein *Bereich von Interesse* (Area of Interest, AOI) eine bestimmte, nicht notwendigerweise zusammenhängende Zone im Gebäude, die in ihren Ausmaßen durch Polygone oder symbolische Koordinaten beschrieben wird. Ausgehend von einem Stockwerk, einem Gebäudeflügel oder einer bestimmten Abteilung kann eine AOI beispielsweise auch Sicherheits- und Gefahrenzonen, nicht öffentliche Bereiche oder andere thematische Gebiete umgrenzen. Für eine gegebene geometrische oder symbolische Koordinate und eine AOI muss es in dem Umgebungsmodell möglich sein, die Koordinate sowohl auf Enthaltensein zu überprüfen, als auch die Euklidische oder topologische Entfernung zur AOI zu bestimmen.
- **Durchführung von Bereichsanfragen:** Eine Bereichsanfrage in einem Umgebungsmodell identifiziert alle Objekte, die sich innerhalb des angefragten Bereiches befinden. Die Auswahl des Bereiches kann durch Angabe einer Position verbunden mit einem Euklidischen oder topologischen Radius, durch ein Polygon oder durch eine Menge von symbolischen Koordinaten erfolgen. Im Falle eines I-LBS, der vollständig

3. Bereitstellung von Umgebungsmodellen für I-LBS

auf einem mobilen Endgerät ausgeführt wird, kann es sinnvoll sein, über eine Bereichsanfrage ganze Ausschnitte aus dem Umgebungsmodell auf das mobile Endgerät herunterzuladen, um dadurch die Ausführung des Dienstes zu beschleunigen und eine erhöhte Client-Server-Kommunikation zu vermeiden.

- **Abbildung der Gebäude-Topologie und Gebäude-Geometrie:** Das Umgebungsmodell soll die Berechnung kürzester Wege ermöglichen. Für viele Anwendungen ist dabei jedoch nicht nur die topologische Distanz zwischen zwei Orten von Interesse, sondern zusätzlich auch die genaue Lage und Form der entsprechenden Route zwischen den beiden Orten. Dies ist beispielsweise für die Visualisierung von Wegen auf einer Karte oder für die Generierung von entsprechenden Navigationsanweisungen relevant. Für zwei gegebene Positionen p_{start} und p_{dest} soll es möglich sein, den kürzesten Weg von p_{start} nach p_{dest} zu berechnen. Zusätzlich soll eine Liste von Wegpunkten $WP = (p_{start}, p_1, p_2, \dots, p_n, p_{dest})$ erhältlich sein, mit der Eigenschaft, dass für jedes Paar von direkt aufeinander folgenden Wegpunkten eine direkte Sichtverbindung besteht. Dazu werden verschiedene Eigenschaften des zugrunde liegenden Wegegraphen benötigt. Dieser sollte zum einen die tatsächliche Begehbarkeit des Gebäudes möglichst gut widerspiegeln und eine möglichst geringe Komplexität aufweisen, zum anderen eine möglichst gute Abdeckung gewährleisten und gleichzeitig die reale Weglänge möglichst gut abbilden. Für die Raumpolygone gilt, dass diese untereinander überlappungsfrei sein müssen.
- **Bereitstellung von Navigationsdaten:** Ein wesentliches Ziel des Umgebungsmodells ist es, auch Navigationsanwendungen zu ermöglichen. Dazu werden nicht nur die Funktionalitäten der Wegwahl benötigt, sondern zusätzlich auch weitere Informationen wie z.B. Landmarken zur Verbesserung von Navigationsanweisungen. Wie bereits in Kapitel 2 beschrieben, bezeichnet der Begriff „Navigation“ an dieser Stelle nicht das bloße Anzeigen von Informationen auf einer Karte, sondern insbesondere auch die Erzeugung dedizierter Anweisungen, die den Benutzer bei seiner Wegfindung durch das Gebäude unterstützen. Daher sollte das von einem Routing-Algorithmus verwendete Wegenetz nicht nur zur Berechnung kürzester Wege geeignet sein, sondern zusätzlich auch eine sinnvolle Nutzung der Wegpunkte für Navigationsanwendungen erlauben.
- **Bereitstellung von Visualisierungsdaten:** Für die Anzeige der Position eines Zielobjekts auf einer Karte, Routen-Darstellungen oder die Visualisierung umliegender POIs sind außerdem Informationen über die Lage und Form von Räumen, Stockwerken und ganzen Gebäuden erforderlich. Daraus ergeben sich verschiedene Anforderungen: zu einer gegebenen Position müssen der umliegende Raum, dessen Form, das zugehörige Stockwerk sowie eventuell umliegende Räume dargestellt werden können. Die Visualisierungsdaten sollten dabei in einer möglichst plattformunabhängigen Form bereitgestellt werden, so dass sie sowohl graphisch umfangreich auf einem leistungsfähigen Computer, als auch in reduzierter Form auf einem mobilen Endgerät dargestellt werden können.

3. Bereitstellung von Umgebungsmodellen für I-LBS

- **Austauschbarkeit des Wegenetzes:** Unterschiedliche Arten von I-LBS stellen unterschiedliche Anforderungen an die Eigenschaften des Wegenetzes. Während zum Beispiel für eine Anwendung zur Auswahl eines nächstgelegenen Raumes ein vergleichsweise einfaches Wegenetz ausreicht, so benötigt eine Navigations-Anwendung unter Umständen wesentlich genauere Informationen über die Begehbarkeit des Gebäudes, um beispielsweise den zu gehenden Weg exakt zu visualisieren. Gleichzeitig beeinflusst die Komplexität (Anzahl und Verzweigungsgrad der Knoten) des Wegenetzes die Antwortzeit von Routen-Berechnungen.

Daher ist es aus Sicht eines Modell-Providers sinnvoll, für ein und dasselbe Gebäude unterschiedliche Varianten von Wegenetzen bereit zu stellen, um so besser auf die Anforderungen eines spezifischen I-LBS reagieren zu können. Die Austauschbarkeit des Wegenetzes bedingt zum einen, dass ein vollständiges und korrektes Modell der Gebäude-Geometrie vorliegt, zum anderen müssen bestimmte Informationen wie zum Beispiel die Übergänge zwischen Räumen und zwischen Stockwerken bekannt sein. Liegen diese Informationen vor, so kann die Begehbarkeit innerhalb eines jeden Raumes entsprechend den Dienst-Anforderungen durch verschiedenste Methoden bestimmt werden.

Diesen funktionalen Anforderungen stehen weitere, *nicht-funktionale* Anforderungen gegenüber:

- **Automatisierte Erzeugung:** Die Erzeugung des Umgebungsmodells sollte möglichst automatisiert erfolgen, denn gerade bei großen Gebäudekomplexen wäre eine manuelle Erzeugung des Modells mit erheblichem Aufwand verbunden. Für die Automatisierung sind daher insbesondere Algorithmen notwendig, die für die Abbildung der Gebäudetopologie und die Bereitstellung von Navigationsdaten geeignet sind.
- **Speicherung des Modells:** Des Weiteren stellt sich die Frage nach der geeigneten Art und Weise zur Speicherung und Bereitstellung des Modells für die verschiedenen Dienste. Ein Umgebungsmodell für I-LBS sollte plattformunabhängig und offen sein, sowie einfach zwischen verschiedenen Komponenten austauschbar sein. Alle für die beteiligten Rollen relevanten Informationen und Beziehungen zwischen Objekten sollten explizit im Modell gespeichert werden. Dies umfasst insbesondere Enthaltensein, topologische Verbindungen und Querverweise zwischen Objekten. Würden diese Informationen nur implizit modelliert, so müsste sie jeder einzelne Dienst mit entsprechend großem Aufwand selbst herleiten. Daneben erleichtert ein durch Menschen lesbares Format die Kontrolle und Wartung von Daten.

Zur Erfüllung der oben genannten Anforderungen werden in einem Umgebungsmodell sowohl die geometrischen, als auch die symbolischen Elemente benötigt. Die geometrische Koordinate eines Zielobjekts lässt sich beispielsweise nur dann einem Raumbezeichner zuordnen, wenn für diesen Bezeichner im Modell auch eine geometrische Beschreibung hinterlegt wurde. Dasselbe gilt für Orte von Interesse. Bei den Bereichen von Interesse stellt sich

3. Bereitstellung von Umgebungsmodellen für I-LBS

die Frage, ob diese immer vollständig innerhalb eines übergeordneten Ortes liegen, oder ob eine AOI z.B. auch zwei verschiedene Räume umfassen kann. Da häufig eher Letzteres der Fall sein wird, bietet sich hierfür die Organisation in einem Verbund an. Bereichsanfragen bedingen, dass alle Elemente in einem Gebäude sich (direkt oder indirekt) in einem gemeinsamen Koordinatensystem miteinander in Bezug setzen lassen.

Die Anforderung, für einen kürzesten Weg zusätzlich auch eine Liste von Wegpunkten zu berechnen, wirft eine zusätzliche Frage auf: *Wie* sollen diese Wegpunkte gewählt werden? Das manuelle Erstellen eines Wegpunktgraphen (siehe z.B. Stahl und Hauptert [133] oder Shafer [132]) ist eine Möglichkeit bei kleinen Gebäuden, bei großen und komplexen Anlagen jedoch mit erheblichem Aufwand verbunden. Werden die Wegpunkte automatisch erzeugt, so gibt es dafür viele verschiedene mögliche Ansätze. Hinsichtlich des Wegpunktgraphen bzw. der Begehrbarkeit eines Gebäudes gehen die existierenden Arbeiten jedoch nicht auf das *Problem der unterschiedlichen Pfadlängen* ein. Je nachdem, welcher Ansatz bei der Graph-Erzeugung gewählt wurde, unterscheidet sich die berechnete Länge eines Pfades zwischen zwei Orten. Ein Beispiel dafür sind zwei verschiedene Modelle, wobei in dem einen die Mittelpunkte der Räume herangezogen werden und in dem anderen nur die Durchgänge der Räume betrachtet werden (siehe auch Abbildung 3.2). Diese unterschiedliche Wahl der Wegpunkte wirkt sich auch darauf aus, wie gut sich das Modell zur Erzeugung von Navigationsanweisungen eignet. Je feinmaschiger ein Wegpunktenetz aufgebaut ist, desto besser lassen sich damit genaue Navigationsanweisungen berechnen.

Um Navigationsanweisungen zu berechnen und dabei insbesondere auch Landmarken zu berücksichtigen, ist es erforderlich, dass die *Sichtbarkeit* zwischen Positionen ermittelt werden kann. Die Sichtbarkeit zwischen Orten steht nicht notwendigerweise eins zu eins mit z.B. deren Enthaltensein, der Überlappung oder auch Nachbarschaft in Verbindung. So können zwei Orte zwar benachbart sein, obwohl sie gegenseitig nicht sichtbar sind, und entsprechend gilt dies auch im umgekehrten Fall. Auch ein Raum, der in einem Stockwerk enthalten ist, ist nicht notwendigerweise von allen Positionen im Stockwerk sichtbar. Aus diesem Grund muss die Sichtbarkeit zwischen Orten separat erfasst und abgebildet werden. Eine Möglichkeit wäre, diese Beziehung wieder über eine Menge abzubilden. Besser eignet sich jedoch ein Graph, da darin auch gleichzeitig die Entfernung zum Ziel abgebildet werden kann.

Um das Wegenetz in einem Umgebungsmodell austauschen zu können, bedarf es einiger grundlegender Informationen, anhand derer die verschiedenen Versionen des Wegenetzes berechnet werden können. Dazu zählen zum einen eine Hierarchie der Räume bzw. Orte, zum anderen auch die Lage und Form der Orte sowie ihre Übergänge. Die Lage und Form des Ortes wird benötigt, um das Wegenetz innerhalb eines einzelnen Raumes (bzw. Polygons) zu erzeugen, und die anhand der Hierarchie können diese Teile zu einem Gesamtmodell zusammengefügt werden. Auf beide Aspekte wird in den Abschnitten 3.4 und 3.5 näher eingegangen.

Zusammengefasst werden folgende Eigenschaften des Modells benötigt:

- Abbildung von *Enthaltensein*, *Überlappung* und *Disjunktheit* von Orten.
- Beschreibung von *Nachbarschaft* und *Verbundenheit* zwischen Orten.

3. Bereitstellung von Umgebungsmodellen für I-LBS

- Darstellung von *Lage* und *Form* von Orten.
- Beschreibung der *Sichtbarkeit* zwischen Orten.
- Abbildung der *begehbaren Bereiche*. Daraus lässt sich in Verbindung mit dem Wissen über die Verbundenheit von Orten auch die *gehbare* Distanz ermitteln.

Die nachfolgenden Abschnitte in diesem Kapitel konzentrieren sich auf die Aspekte der *begehbaren Bereiche*, sowie der *Nachbarschaft* und *Verbundenheit* zwischen Orten. Dazu werden im Abschnitt 3.4 verschiedene Algorithmen zur Erzeugung von Wegenetzen betrachtet und im Abschnitt 3.5 wird diskutiert, wie ein Umgebungsmodell automatisiert erzeugt werden kann. Eine Evaluierung der verschiedenen Wegenetze findet sich im Abschnitt 3.5.2. In Kapitel 4 wird auf die Speicherung eines Umgebungsmodells in einem einheitlichen Format eingegangen und Kapitel 5 behandelt die Bereitstellung von Navigationsdaten.

Zunächst wird jedoch im nachfolgenden Abschnitt 3.3 näher auf den Begriff des *Positionierungsverfahrens* eingegangen. Positionsdaten aus verschiedenen Positionierungsverfahren stellen elementare Informationen für die Erbringung eines LBS dar, und bei der Erzeugung eines Umgebungsmodells müssen diese unterschiedlichen Arten von Positionsdaten berücksichtigt werden. Ebenfalls im nachfolgenden Abschnitt wird am Beispiel von GPS- und GSM-Cell-Id-Positionierung gezeigt, wie sich verschiedene Positionierungsverfahren miteinander kombinieren lassen, um so z.B. Energie auf dem mobilen Endgerät zu sparen.

3.3. Positionsdaten aus heterogenen Positionierungsverfahren

Die zur Ausführung eines I-LBS benötigten Informationen über die Aufenthaltsorte der beteiligten Zielobjekte können über verschiedenste Positionierungsverfahren gewonnen werden. Bei der Entwicklung eines umfassenden Umgebungsmodells für I-LBS ist daher zu beachten, in welchem Format und in welcher Qualität die jeweiligen Positionsdaten vorliegen. Wie bereits im Abschnitt 2.1.3 erwähnt, lassen sich alle Verfahren grundsätzlich in drei verschiedene Klassen einteilen (endgerätbasiert, vom Endgerät unterstützt und netzbasiert). Dabei bezeichnet der Begriff „Endgerät“ eine beliebige mobile Entität, die durch Sensoren, Mess- und Berechnungsverfahren in der Lage ist, die Position eines Zielobjektes anhand von Pilotsignalen oder -informationen entweder autonom oder, im Falle der Endgerät-unterstützten und netzbasierten Positionierung, durch Kooperation mit anderen Komponenten zu ermitteln.

Das Ergebnis einer Positionsermittlung ist der Aufenthaltsort eines Zielobjektes. Die Repräsentation des Aufenthaltsorts erfolgt innerhalb eines bestimmten lokalen oder globalen Referenzsystems. Gleichzeitig kann die Angabe des Aufenthaltsortes auch zusätzliche Informationen wie beispielsweise Genauigkeit und Präzision des verwendeten Messverfahrens beinhalten.

3. Bereitstellung von Umgebungsmodellen für I-LBS

Viele existierende Ansätze für Indoor-Positionierungsverfahren basieren auf der Verarbeitung der Funksignale von IEEE 802.11-konformen¹ drahtlosen lokalen Netzen (Wireless Local Area Network, WLAN). Frühe Beispiele sind das bereits angesprochene System von Bahl und Padmanabhan [10] oder auch Castro et al. [26] und Youssef und Agrawala [158]. Die Verbreitung und Verfügbarkeit von WLANs hat in den vergangenen Jahren sprunghaft zugenommen und mittlerweile ist auch ein großer Anteil mobiler Endgeräte mit entsprechend kompatibler Hardware ausgestattet. Ein großer Vorteil der WLAN-basierten Positionierung ist daher eine oftmals flächendeckende Verfügbarkeit sowohl auf Seiten der Infrastruktur als auch bei den mobilen Endgeräten. Jedoch hängt die Genauigkeit stark von der Dichte der umliegenden Access Points sowie der baulichen Beschaffenheit eines Gebäudes ab. Gerade letzteres wirkt sich stark auf die beobachteten Parameter wie z.B. die empfangene Signalstärke oder das Signal-zu-Rausch-Verhältnis aus. Dadurch kann die Genauigkeit in einem Teil eines Gebäudes im Bereich weniger Meter liegen, wohingegen in einem anderen Bereich die Genauigkeit auf über 30 Meter steigt oder überhaupt keine Positionierung mehr möglich ist. Übertragen auf Umgebungsmodelle bedeutet dies, dass nicht grundsätzlich davon ausgegangen werden kann, dass eine Positionsbestimmung mit Raumgenauigkeit möglich ist und es daher erforderlich ist, auch ungenaue Positionsangaben einem Bereich im Gebäude zuzuordnen.

Neben WLAN-basierten Ansätzen sind beispielsweise auch RFID- (Radio Frequency Identification), Ultrabreitband- (Ultra Wide Band, UWB) [50], Ultraschall- oder Infrarot-Sensoren [149] zur Positionierung geeignet. Auch existieren Ansätze zur Positionierung über die Kamera eines mobilen Endgerätes durch Erkennung von umliegenden visuellen Mustern (z.B. Barcodes) [71, 125]. Einen Überblick über verschiedene Positionierungsverfahren geben auch Kolodziej und Hjelm [83] sowie Hightower und Borriello [58].

Unabhängig von der Art und Weise der verwendeten Pilotsignale bzw. Messdaten lassen sich generell fünf verschiedene Methoden zur Positionierung unterscheiden:

1. **Proximity Sensing:** Das Erkennen räumlicher Nähe erfordert Sensoren, welche in der Lage sind, zuvor bekannte Pilotsignale von in der Umgebung befindlichen Sendern zu empfangen. Im Falle von endgerätbasierter Positionierung befinden sich die Sensoren im Endgerät und die Pilotsignale werden von Netz-Komponenten gesendet. Im umgekehrten Fall der netzbasierten Positionierung sendet das Endgerät ein Pilotsignal aus, welches von Netz-Komponenten empfangen und ausgewertet wird.
2. **Laufzeitmessung (Lateration):** Sind Zeitpunkt des Absendens und Empfangs eines Pilotsignals bekannt, so kann anhand der Signallaufzeit die räumliche Distanz zwischen Sender und Empfänger bestimmt werden. Anhand der Signallaufzeit zu mindestens drei verschiedenen Sendern, deren Aufenthaltsorte bekannt sind, lässt

¹Der Begriff „IEEE 802.11“ wird an dieser Stelle stellvertretend für die verschiedenen Standards und Erweiterungen der 802.11-Protokollfamilie verwendet. Das ursprüngliche Protokoll 802.11 wurde zwischenzeitlich durch zahlreiche Erweiterungen (z.B. 802.11a, 802.11e, 802.11g, 802.11n usw.) ergänzt, welche sich vor allem in den verwendeten Modulations- und Mehrfachzugriffsverfahren unterscheiden. Ein Großteil dieser Erweiterungen ist inzwischen unter dem Standard IEEE 802.11-2007 zusammengefasst.

3. Bereitstellung von Umgebungsmodellen für I-LBS

sich eine eindeutige Position des Empfängers berechnen. Voraussetzung dafür ist, dass Sender und Empfänger untereinander zeitlich synchronisiert sind und eine hinreichend genaue Uhr für die Messung zur Verfügung steht.

3. **Winkelmessung (Angulation):** Die Positionierung über Winkelmessung erfolgt z.B. über sektorisierte Antennen oder Antennen-Arrays, die in der Lage sind, den Einfallswinkel eines empfangenen Signals zu bestimmen. Werden die Winkelmessungen von mehreren empfangenden Stationen kombiniert, so kann der Aufenthaltsort eines Senders entsprechend eingegrenzt werden.
4. **Mustererkennung (Pattern Matching):** Neben Lateration und Angulation existieren zahlreiche weitere Positionierungs-Methoden, die bestimmte für die Umgebung charakteristische Merkmale betrachten und allgemein unter dem Begriff *Pattern Matching* zusammengefasst werden. Beispielsweise ergeben die Basic Service Set Identifier (BSSIDs) von umliegenden WLAN-Access-Points zusammen mit den gemessenen Signalstärken einen charakteristischen Vektor (Fingerprint), die sich jeweils in Abhängigkeit des Raumes in einem Gebäude unterscheidet. Ebenso kann das Signal-zu-Rausch-Verhältnis (Signal to Noise Ratio, SNR) verschiedener Pilotsignale unterschiedliche Orte charakterisieren. Dabei lassen sich Fingerprint-basierte Methoden unterteilen in probabilistische und modellbasierte Ansätze. Erstere erfordern die Durchführung von initialen Referenzmessungen in der Umgebung, um so eine Menge von zu erwartenden Signalstärkevektoren an verschiedenen Punkten zu erhalten. Zur Durchführung der Positionierung wird dann der aktuell gemessene Fingerprint mit der vorhandenen Menge an Referenzmessungen verglichen und in Abhängigkeit einer Metrik der wahrscheinliche Aufenthaltsort bestimmt. Die modellbasierten Ansätze erfordern keine Referenzmessungen, sondern berechnen den Aufenthaltsort anhand von Signalausbreitungsmodellen.
5. **Dead Reckoning:** Ausgehend von einer bekannten Startposition eines Zielobjekts kann ein zukünftiger Aufenthaltsort bei bekannter und gleich bleibender Bewegungsrichtung sowie Geschwindigkeit durch Koppelnavigation berechnet werden. Dies stellt an sich keine vollständige Methode zur Positionierung dar, wird jedoch oftmals in Kombination mit einer der voran genannten Methoden eingesetzt.

Entscheidend für viele Dienste ist nicht nur die Genauigkeit eines Positionierungsverfahrens, sondern vor allem auch die Robustheit gegenüber Störungen und Änderungen in der Umgebung. Insbesondere die Mehrwegeausbreitung (multipath propagation) ist ein Effekt, der innerhalb von Gebäuden häufig zu Störungen führt. In Verbindung mit baulichen Änderungen in einem Gebäude kann ein Positionierungssystem ebenfalls beeinflusst werden, beispielsweise wenn dadurch die Messwerte von zuvor aufgenommenen Radio-Fingerprints verändert werden. Des Weiteren können im geplanten Empfangsbereich eines drahtlosen lokalen Netzes auch Funklöcher auftreten, innerhalb derer kein Empfang mehr möglich ist. In diesem Bereichen bietet sich eine Kombination mit dem Mobilfunknetz an, indem dann die umliegenden Basisstationen dieses Netzes verwendet werden (siehe beispielsweise Borriello

3. Bereitstellung von Umgebungsmodellen für I-LBS

et al. [20]). Ein zusätzliches Problem ist durch die Heterogenität der drahtlosen Adapter in den Endgeräten gegeben. Bei den auf Fingerprinting basierenden Ansätzen entstehen dann zum Teil große Fehler, wenn für Kalibrierung und Positionierung unterschiedliche Hardware verwendet wird. Um diesen Fehler auszugleichen, stellt Kjærgaard [78] einen Ansatz vor, bei dem anstelle absoluter Signalstärken das Verhältnis von Signalstärken zwischen Paaren von Basisstationen betrachtet wird. Des Weiteren ist es für zukünftige Dienste auch entscheidend, dass sie die aktuelle Ungenauigkeit des Positionierungsverfahrens kennen, um entsprechend darauf reagieren zu können. Ein Ansatz zur Fehlerabschätzung bei der WLAN-Positionierung wird beispielsweise von Lemelson [91] vorgestellt. Neben den alltäglichen Diensten und Anwendungen für die private und geschäftliche Nutzung konzentriert sich eine wachsende Zahl von Projekten auch auf die Positionierung und Navigation von Feuerwehrleuten und Rettungskräften in Gebäuden, wie eine aktuelle Übersicht von Fischer und Gellersen [47] zeigt. Viele der darin vorgestellten Ansätze versuchen vor allem, mit Hilfe von Sensoren wie Gyroskop, Kompass und Beschleunigungsmesser die Rettungskräfte zu verfolgen. Ein weiteres Einsatzgebiet sind Krankenhäuser, dort bietet sich der Einsatz von Positionierungsverfahren vor allem zur Optimierung von Arbeitsabläufen an (Stantchev et al. [134]).

3.3.1. Positionsdaten und Koordinatensysteme

Die ermittelten Aufenthaltsorte der Zielobjekte werden in Form geometrischer oder symbolischer Koordinaten ausgedrückt. Im Falle von geometrischen Koordinaten liefern existierende Indoor-Positionierungsverfahren zumeist nur eine Koordinate innerhalb eines lokalen zweidimensionalen Koordinatensystems zurück. Verbunden damit wird beispielsweise ein Stockwerksbezeichner oder die ID einer bestimmten Karte angegeben und in manchen Systemen ist es möglich, Bereichen (Polygonen) symbolische Bezeichner zuzuordnen und somit ein rudimentäres hybrides Umgebungsmodell anzulegen. Jedoch beschränken sich diese Funktionalitäten auf die Geometrie von Räumen, und eine Topologie des Gebäudes kann daraus nicht abgeleitet werden. Gleichzeitig erschwert die enge Kopplung von Positionierungssystem und Raumgeometrie eine Verwendung durch andere Akteure bzw. ist aufgrund nicht vorhandener Schnittstellen gar nicht erst möglich. Natürlich kann es sinnvoll sein, sämtliche benötigte Informationen über ein Gebäude (Raumgeometrien, Topologie der Räume etc.) innerhalb des Positionierungssystems zu halten und zu verwalten. Da es sich bei solchen Ansätzen allerdings um geschlossene Systeme handelt, die in der Regel eine ganz spezielle Aufgabe übernehmen und nicht mit anderen Systemen in Interaktion stehen, werden diese im Folgenden nicht weiter beachtet.

In der Praxis basieren viele Indoor-Positionierungssysteme bislang auf rasterbasiertem Kartenmaterial, wobei die Angabe einer Position z.B. in Form von Pixelkoordinaten auf einem zugehörigen Stockwerksplan erfolgt. Das nachfolgende Beispiel zeigt eine Positionsangabe eines Endgerät-unterstützten WLAN-Positionierungssystems:

```
1 <position>  
   <mac>38:E7:D8:0E:C9:0B</mac>  
3   <posx>1320</posx>
```

3. Bereitstellung von Umgebungsmodellen für I-LBS

```

    <posy>2037</posy>
5   <posmapid>3</posmapid>
    <posquality>61</posquality>
7   <postimestamp>2010-09-08 17:12:42+0200</postimestamp>
    <poscounter>469</poscounter>
9   <battery>61</battery>
    <!-- ... -->
11 </position>

```

In diesem Beispiel befindet sich das Endgerät an der Pixel-Position (1320, 2037) auf der Karte mit der Id 3. Ohne weitere Informationen über die zugehörige Karte, sowie deren Lage und Maßstab kann diese Positionsangabe nicht von anderen Systemen ausgewertet werden.

Zur Integration von Positionsdaten aus heterogenen Indoor-Positionierungssystemen ist es daher erforderlich, dass die Koordinaten aus verschiedenen Positionierungssystemen miteinander in Bezug gesetzt werden können. Gleichzeitig ist es sinnvoll, dass auch eine Einordnung in globale Referenzsysteme möglich ist, insbesondere zur Kombination mit den auf vielen Endgeräten vorhandenen GPS-Positionsdaten.

Die Umrechnung zwischen zwei verschiedenen Kartesischen Koordinatensystemen kann durch eine Helmert-Transformation erreicht werden: $X_z = T + R * M * X_s$, wobei X_s ein zu transformierender Startpunkt, X_z der Zielpunkt im neuen Koordinatensystem, T ein Verschiebungsvektor, R eine Rotationsmatrix, und M den Maßstab zwischen den beiden Systemen bezeichnet. M kann entweder für jede Achse ein spezieller Maßstabsfaktor oder (wie zumeist angewandt) ein Maßstabsfaktor für alle Achsen sein. Die Rotationsmatrix R setzt sich im dreidimensionalen Fall aus drei aufeinander folgenden Rotationen $R = R_3\{\alpha_3\}R_2\{\alpha_2\}R_1\{\alpha_1\}$ zusammen, genauer gesagt:

$$R = \begin{bmatrix} \cos(\alpha_2)\cos(\alpha_3) & \cos(\alpha_1)\sin(\alpha_3) & \sin(\alpha_1)\sin(\alpha_3) \\ -\cos(\alpha_2)\sin(\alpha_3) & \cos(\alpha_1)\cos(\alpha_3) & \sin(\alpha_1)\cos(\alpha_3) \\ \sin(\alpha_2) & -\sin(\alpha_1)\cos(\alpha_2) & \cos(\alpha_1)\cos(\alpha_2) \end{bmatrix}$$

In der Ebene ergibt sich dadurch:

$$\begin{bmatrix} x_z \\ y_z \end{bmatrix} = \begin{bmatrix} T_x \\ T_y \end{bmatrix} + \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} * \begin{bmatrix} M_x & 0 \\ 0 & M_y \end{bmatrix} \begin{bmatrix} x_s \\ y_s \end{bmatrix}$$

Sind die Transformationsparameter unbekannt, so können sie anhand gemeinsamer (identischer) Punkte bestimmt werden. D.h. dass für eine Menge von Punkten die jeweiligen Koordinaten in beiden Koordinatensystemen bekannt sein müssen. Zur Bestimmung einer 7-Parameter-Helmert-Transformation sind für dreidimensionale Koordinaten theoretisch zwei Punkte und mindestens eine zusätzliche gemeinsame Komponente eines dritten

3. Bereitstellung von Umgebungsmodellen für I-LBS

Punktes (z.B. die Höhe) ausreichend, um das resultierende Gleichungssystem zu lösen. Praktisch werden jedoch meist mehrere Punkte verwendet und die unbekannt Parameter über die Methode der kleinsten Quadrate angeglichen. Eine umfassende Diskussion über Koordinaten- und Datumstransformationen findet sich in Hofmann-Wellenhof et al. [59, S. 278 - 307].

Um die lokalen Koordinaten aus einem Indoor-Positionierungsverfahren mit den Positionsdaten oder Inhalten eines Drittanbieters in Bezug zu setzen, bietet sich eine Transformation in WGS84-Koordinaten an. Da dieses Referenzsystem von nahezu allen Outdoor-LBS genutzt bzw. unterstützt wird, können so beispielsweise Live-Positionen in einer globalen Karte angezeigt werden. Umgekehrt ermöglicht eine solche Transformation auch, dass mehrere lokale Positionierungsverfahren (bzw. ihre Koordinatensysteme) anhand globaler gemeinsam bekannter Punkte miteinander in Bezug gesetzt werden.

Im Folgenden wird die Umrechnung von WGS84-Koordinaten $(\phi, \lambda, h)_{gps}$ in ein lokales, nicht geozentrisches Koordinatensystem betrachtet. Die lokalen Koordinaten sind durch $(x, y)_{lokal}$ und eine ellipsoidale Höhe h_{LS} gegeben. Es wird davon ausgegangen, dass mindestens drei gemeinsame Punkte in beiden Koordinatensystemen bekannt sind. Zunächst werden $(\phi, \lambda, h)_{GPS}$ wie folgt in Kartesische Koordinaten $(X, Y, Z)_{gps}$ umgerechnet:

$$X = (N + h) \cos \phi \cos \lambda$$

$$Y = (N + h) \cos \phi \sin \lambda$$

$$Z = \left(\frac{b^2}{a^2} N + h \right) \sin \phi$$

wobei

$$N = \frac{a^2}{\sqrt{a^2 \cos^2 \phi + b^2 \sin^2 \phi}}$$

ist und die Werte der großen und kleinen Halbachse zur Umrechnung von WGS84-Koordinaten wie folgt definiert sind:

$$a = 6378137.0m$$

$$b = 6356752.314245m$$

Anschließend werden gemäß [59] folgende Schritte durchgeführt:

1. Die lokalen Koordinaten $(x, y)_{lokal}$ der gemeinsam bekannten Punkte werden in Kugelkoordinaten $(\varphi, \lambda)_{lokal}$ umgewandelt.
2. Die elliptischen Koordinaten $(\varphi, \lambda, h)_{lokal}$ der gemeinsam bekannten Punkte werden in Kartesische Koordinaten $(X, Y, Z)_{lokal}$ umgewandelt.
3. Bestimmung der Transformationsparameter der Helmert-Transformation unter Verwendung der Koordinaten $(X, Y, Z)_{lokal}$ und $(X, Y, Z)_{gps}$ der gemeinsam bekannten Punkte.

3. Bereitstellung von Umgebungsmodellen für I-LBS

4. Transformation der gewünschten GPS-Koordinaten anhand der in Schritt 3 berechneten Transformationsparameter in Kartesische Koordinaten $(X, Y, Z)_{\text{lokal}}$.
5. Transformation der in Schritt 4 berechneten Kartesischen Koordinaten in elliptische Koordinaten $(\varphi, \lambda, h)_{\text{lokal}}$.
6. Abbildung der in Schritt 5 berechneten Kugelkoordinaten $(\varphi, \lambda)_{\text{lokal}}$ auf lokale Koordinaten $(x, y)_{\text{lokal}}$.

Die Transformation von Kartesischen Koordinaten (X, Y, Z) in (ϕ, λ, h) ist dabei gegeben durch:

$$\phi = \arctan \frac{Z + e'^2 b \sin^3 \theta}{p - e^2 a \cos^3 \theta}$$

$$\lambda = \arctan \frac{Y}{X}$$

$$h = \frac{p}{\cos \phi} - N$$

wobei a und b die große bzw. kleine Halbachse des Ellipsoiden und

$$p = \sqrt{X^2 + Y^2}$$

$$\theta = \arctan \frac{Za}{pb}$$

$$e^2 = \frac{a^2 - b^2}{a^2}$$

und

$$e'^2 = \frac{a^2 - b^2}{b^2}$$

ist. Neben dieser geschlossenen Form existiert auch ein anderes Verfahren, das ausgehend von einem ungefähren Wert $\tan \phi_{(0)}$ und dem resultierenden N_0 den Wert von ϕ iterativ bestimmt:

$$\tan \phi_{(0)} = \frac{Z}{p} (1 - e^2)^{-1}$$

$$N_{(0)} = \frac{a^2}{\sqrt{a^2 \cos^2 \phi_{(0)} + b^2 \sin^2 \phi_{(0)}}}$$

$$h = \frac{p}{\cos \phi_{(0)}} - N_{(0)}$$

Im nächsten Schritt wird für ϕ ein verbesserter Wert berechnet:

$$\tan \phi = \frac{Z}{p} \left(1 - e^2 \frac{N_{(0)}}{N_{(0)} + h} \right)^{-1}$$

3. Bereitstellung von Umgebungsmodellen für I-LBS

Falls nun $\phi = \phi_{(0)}$ ist, so ist die Berechnung beendet. Ansonsten wird $\phi_{(0)} = \phi$ gesetzt und beginnend mit dem zweiten Schritt wird ein neuer Wert für $N_{(0)}$ berechnet.

Nachfolgend soll die Genauigkeit dieser Berechnungen anhand eines Beispiels betrachtet werden, bei dem GPS-Koordinaten zuerst in Kartesische Koordinaten und anschließend wieder zurück gewandelt werden. Alle Zahlen werden mit *double*-Genauigkeit behandelt. Sei

$$\phi = 48.1497392, \lambda = 11.5944393, h = 508.0$$

Mit sieben signifikanten Nachkommastellen beim Längen- und Breitengrad lässt sich eine Position bereits mit einer Genauigkeit von ca. 1 cm angeben. Eine Transformation in Kartesische Koordinaten ergibt:

$$X = 4176667.8126530196$$

$$Y = 856924.3471521297$$

$$Z = 4728379.42305642$$

Transformiert man diese Koordinaten mittels des iterativen Verfahrens wieder zurück, so erhält man nach insgesamt acht Iterationen:

$$\phi = 48.14973919999999, \lambda = 11.5944393, h = 508.0$$

Auch bei Verwendung der geschlossenen Form bleiben die signifikanten Stellen unverändert und es ergibt sich:

$$\phi = 48.149739200000006, \lambda = 11.5944393, h = 508.00000000186265$$

Dieses Beispiel zeigt, dass die erreichbare Genauigkeit der Transformationen für die Verarbeitung von Positionsdaten mehr als ausreichend ist. Wenn allerdings zwei Koordinaten miteinander verglichen werden sollen, so müssen die durch die Gleitkommazahlen verursachten Rundungsfehler beachtet sowie ein entsprechender Toleranzbereich verwendet werden.

Die oben betrachteten Koordinatentransformationen werden nicht nur zur Transformation von GPS-Koordinaten benötigt. Die Helmert-Transformation kommt vor allem auch immer dann zum Einsatz, wenn die Koordinaten aus mehreren verschiedenen lokalen Positionierungsverfahren miteinander in Bezug gesetzt werden müssen. Auf diese Kombination wird im nächsten Abschnitt näher eingegangen.

3.3.2. Kombination von Positionierungsverfahren

Bei der Konzeption und Implementierung von I-LBSs ist zu beachten, dass das verwendete Positionierungssystem während der Ausführung des Dienstes wechseln kann. Dies ist insbesondere dann sinnvoll, wenn dadurch beispielsweise Energie auf dem Endgerät eingespart werden kann oder insgesamt eine bessere Abdeckung möglich wird. Nicht jeder Dienst benötigt zu jedem Zeitpunkt eine konstante Genauigkeit der Positionierung. Vielmehr sind

3. Bereitstellung von Umgebungsmodellen für I-LBS

es die jeweiligen Aufgaben des Dienstes, welche die momentan benötigte Genauigkeit und Präzision bestimmen. Ein Beispiel dafür ist das Erkennen des Betretens oder Verlassens einer Zone durch ein Zielobjekt, wobei ein Alarm nur außerhalb eines Toleranzbereichs von 5 Metern entlang der Zonenbegrenzung ausgelöst werden soll. Angenommen, es stehen dafür zwei Positionierungsverfahren POS_A und POS_B zur Verfügung. POS_A hat eine Genauigkeit von 20m in 95% aller Messungen und POS_B hat eine Genauigkeit von 3m in 95% aller Messungen. POS_A basiert auf der passiven Messung von Signalen umliegender Stationen und POS_B verwendet zum Positionierung aktive Pilotsignale, die jede Sekunde vom Zielobjekt ausgesendet und von umliegenden Stationen empfangen werden. POS_B benötigt dementsprechend wesentlich mehr Energie als POS_A . In einem solchen Szenario kann der Energieverbrauch auf dem Endgerät optimiert werden, indem dynamisch zwischen POS_A und POS_B gewechselt wird. Solange das Zielobjekt eine Entfernung von 20m zum Rand der Zone nicht unterschreitet, kann das energiesparende Verfahren POS_A verwendet werden. Erst beim Annähern an den Rand der Zone ist das Umschalten auf POS_B notwendig. Handelt es sich um eine hinreichend große Zone, so kann möglicherweise auch innerhalb der Zone wieder zu POS_A gewechselt werden.

Dieses Prinzip kann nicht nur beim Erkennen des Betretens und Verlassens einer Zone verwendet werden, sondern eignet sich bei der Verwaltung von Positionsinformationen auch für viele andere Basismechanismen, wie z.B. das Erkennen der gegenseitigen Annäherung von mobilen Zielobjekten beziehungsweise von Clustern mobiler Zielobjekte (siehe Treu [145]).

Für ein Umgebungsmodell bedeutet dies, dass sich die Positionsdaten aus den verschiedenen Positionierungsverfahren zu entsprechenden Positionen im Modell zuordnen lassen müssen, dafür ist auch die Beschreibung von bestimmten Bereichen erforderlich. Beispielsweise können mit zellbasierten Positionierungsverfahren keine exakten Positionen bestimmt werden, denn der ermittelte Aufenthaltsort entspricht dem Empfangsbereich der jeweiligen Funkzelle, und diese deckt einen nicht notwendigerweise zusammenhängenden Bereich ab.

Die Kombination von Positionierungsverfahren ist sowohl im Indoor- als auch im Outdoor-Bereich relevant. Insbesondere der Wechsel vom Außenbereich in ein Gebäude stellt eine Herausforderung dar, wenn eine nahtlose Verbindungsübergabe von einem zum anderen Positionierungsverfahren erfolgen soll. Zum einen müssen dafür die Positionierungsverfahren vorhanden und für das jeweilige Zielobjekt verfügbar sein, zum anderen muss der Location Provider Zugriff auf die Positionsinformationen haben und diese dem Zielobjekt zuordnen können.

Im folgenden Abschnitt wird zunächst ein neuer Ansatz zur Kombination von Positionierungsverfahren im Outdoor-Bereich vorgestellt. Die Grundidee basiert dabei auf der Kombination zweier Verfahren und ihrer spezifischen Vorzüge: ein Verfahren, das besonders genau und präzise ist, wird mit einem Verfahren kombiniert, das zwar weniger genau, dafür aber wesentlich Energie-effizienter und nahezu flächendeckend verfügbar ist. Dazu wird auf einem mobilen Endgerät dynamisch zwischen einer GPS-Positionierung und einer GSM-Zellen-basierten Ortung gewechselt. Im Anschluss an dieses Verfahren wird diskutiert, wie sich dieses Prinzip auf den Indoor-Bereich bzw. auf Indoor-Positionierungsverfahren übert-

3. Bereitstellung von Umgebungsmodellen für I-LBS

ragen lässt und welche Auswirkungen sich daraus für ein Umgebungsmodell ergeben.

Kombination von GPS und GSM-Cell-Id: Im Outdoor-Bereich sind heutzutage vor allem zwei verschiedene Arten von Positionierungsverfahren verbreitet: zum einen die satellitengestützte Positionierung (GPS), zum anderen Zell-basierte Positionierung (z.B. anhand der GSM-Cell-ID oder der BasicServiceSetID eines WLANs). Letztere erfordern, dass für eine bestimmte Zelle die geographische Position der zugehörigen Station bekannt ist.

Das GPS bietet für zivile Nutzer je nach Umgebung und Leistungsfähigkeit des Empfängers eine Genauigkeit im Bereich weniger Meter. Demgegenüber steht jedoch ein hoher Energieverbrauch. In einer Untersuchung von Kjærgard [76] liegt der Verbrauch eines GPS-Empfängers in einem Nokia N95 bei 0,32 Watt. Carrol und Heiser [25] messen auf einem Openmoko Neo Freerunner einen Verbrauch von 0,14 Watt. Die Unterschiede im Energieverbrauch von GPS-Empfängern lassen sich dabei nicht nur auf den konkreten GPS-Chip zurückführen. Sie sind auch vom Betriebssystem abhängig, insbesondere davon, wie die CPU des Endgerätes bei der Positionierung involviert ist und aus welchem Software-Framework heraus auf den GPS-Empfänger zugegriffen wird.

Eine Zell-basierte Positionierung in Mobilfunknetzen liefert zwar nur vergleichsweise ungenaue Positionen in der Größenordnung von 50m bis mehreren Kilometern, jedoch braucht dieser Ansatz neben dem GSM-Modul keine zusätzliche Komponente im Endgerät. Die Positionierung anhand von WLAN-Parametern ist heutzutage in manchen dicht besiedelten Umgebungen zwar möglich, erfordert jedoch eine umfangreiche Datensammlung im Voraus und gleichzeitig ein aktiviertes WLAN-Modul auf dem Endgerät.

Küpper und Treu [90] beschreiben einen Ansatz zur effizienten Erkennung von Nachbarschaftsbeziehungen zwischen mobilen Zielobjekten (siehe auch Küpper et al. [85]). Dieser basiert auf der Idee, dass ein zentraler Server (Location Provider) die Positionen mehrere Zielobjekte überwacht und so erkennen kann, wann zwei Zielobjekte eine bestimmte Distanz d_p (*proximity distance*) zueinander unterschritten haben bzw. wann eine bestimmte Distanz d_s (*separation distance*) zueinander überschritten wird. Dazu wird jedem Zielobjekt dynamisch ein kreisförmiger Bereich zugeteilt, innerhalb dessen sich das Zielobjekt frei bewegen kann, ohne dass es seine Position erneut dem Location Provider mitteilen muss. Die Berechnung des Bereiches kann nach verschiedenen Kriterien und unter Berücksichtigung von verschiedenen Schwellwerten zur Kompensation der Ungenauigkeit eines Positionierungsverfahren erfolgen. Dies wird durch die Verwendung einer Grenzlinientoleranz b (*borderline tolerance*) erreicht, innerhalb derer das Unter- und Überschreiten von d_p und d_s erkannt werden kann, aber nicht muss. Beim Verlassen des momentan zugewiesenen Bereiches (des so genannten *distance jobs*) könnte sich das Zielobjekt einem anderen Zielobjekt angenähert haben und meldet daher seine aktuelle Position an den Location Provider. Darauf hin werden allen betroffenen Zielobjekten neue *distance jobs* mitgeteilt. Der Ansatz bietet im Vergleich zu anderen Strategien (z.B. periodische Übertragung der Koordinaten oder Verwendung von Polygonen) eine deutlich bessere Effizienz bezüglich der Anzahl der übertragenen Nachrichten zwischen Zielobjekt und Location Provider.

Dieser und ähnliche Ansätze von Xu und Jacobsen [155, 156], Amir et al. [5] oder Bamba

3. Bereitstellung von Umgebungsmodellen für I-LBS

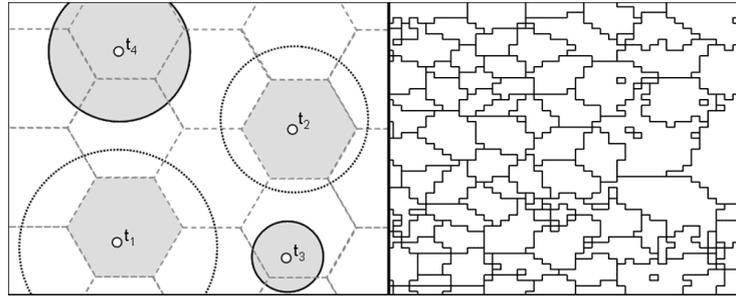


Abbildung 3.3.: Links: Kombination von GPS und GSM-Cell-ID (idealisierte Zellen) zur Erkennung von Nachbarschaftsbeziehungen zwischen Zielobjekten. Rechts: reale Form von 120° -GSM-Zellen im städtischen Bereich, Auflösung des Rasters: 63m.

et al. [11] sind zunächst unabhängig von dem zugrunde liegenden Positionierungsverfahren. Auf der Ebene der Positionierungsverfahren finden sich jedoch Möglichkeiten, die Energieeffizienz aus Sicht der beteiligten Endgeräte weiter zu verbessern.

Nachfolgend wird ein neuer Ansatz vorgestellt, der erstmals in [35] präsentiert wurde und der eine energieeffiziente Kombination von GPS und GSM-Cell-ID für proaktive LBS ermöglicht: Wann immer möglich wird anstatt GPS die ungenauere, aber dafür energieeffizientere Positionierung über das Mobilfunknetz verwendet. Dabei wird davon ausgegangen, dass der Location Provider bei gegebenem Mobile Network Code, Location Area Identifier und Cell-ID die zugehörige Koordinate der GSM-Basisstation sowie eine Abschätzung über die Abdeckung der Zelle (die so genannten *best server estimates*) kennt bzw. dass diese Informationen über eine Schnittstelle beim Mobilfunknetzbetreiber abgefragt werden können.

Abbildung 3.3 (linke Seite) zeigt vier Zielobjekte t_1, \dots, t_4 , die Kreise entsprechen den aktuell berechneten Bereichen in denen sich die Zielobjekte frei bewegen können, ohne dass sie einander zu nahe kommen. Die hexagonalen Flächen entsprechen den idealisierten Flächen eines GSM-Mobilfunknetzes. t_1 und t_2 befinden sich jeweils in einer *safe cell*, diese Zellen liegen vollständig innerhalb des *distance jobs*. In diesem Fall bekommen t_1 und t_2 einen so genannten *cell job* zugeordnet und können ihren GPS-Empfänger deaktivieren. Die weitere Positionierung erfolgt mittels der energieeffizienteren zellbasierten Methode und erst beim Verlassen der *safe cell* ist es notwendig, den GPS-Empfänger wieder zu aktivieren.

In der Realität sind die Zellen eines Mobilfunknetzes meist in Segmenten angeordnet, so dass an einem Standort drei Antennen einen Bereich von jeweils 120° abdecken. Aufgrund von Gebäuden und Erhebungen in der Landschaft sowie der Tatsache, dass der Mobilfunknetzbetreiber nicht an jedem gewünschten Ort eine Basis-Station errichten kann, sind die Zellen nicht gleichmäßig verteilt. Abbildung 3.3 (rechte Seite) zeigt einen Ausschnitt eines Mobilfunknetzes in der Münchener Innenstadt, die Auflösung des Rasters beträgt 63m. Diese Darstellungen lassen sich auf der Grundlage der so genannten *best server estimates*

3. Bereitstellung von Umgebungsmodellen für I-LBS

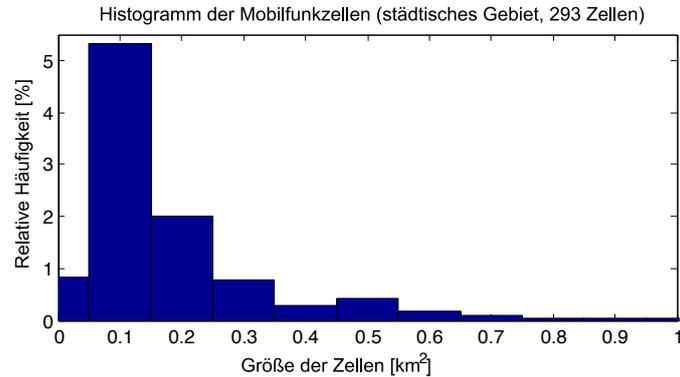


Abbildung 3.4.: Histogramm der Größe von GSM-Zellen in einem städtischen Mobilfunknetz. Gesamtanzahl der betrachteten Zellen: 293

berechnen: für jede Gitterzelle wird anhand von Position und Abstrahlwinkel der Antennen umliegender Basistationen, Pfadverlustmodellen, Höhe und Beschaffenheit von Gebäuden und Bepflanzungen sowie weiteren Parametern bestimmt, in welcher Funkzelle ein Endgerät mit der größten Wahrscheinlichkeit eingebucht ist. Betrachtet man das in Abbildung 3.4 dargestellte Histogramm der Zell-Flächen für einen innerstädtischen Bereich, so fällt auf, dass über 70% der 293 betrachteten Zellen nur zwischen $0,1 - 0,2 \text{ km}^2$ groß sind und damit für viele Anwendungen eine ausreichend große Genauigkeit bieten.

Zur Evaluierung einer dynamischen Kombination von GPS und Cell-Id-Positionierung lässt sich der prozentuale Anteil der Zeit messen, innerhalb derer der GPS-Empfänger nicht benötigt wird. Dazu werden folgende Strategien unterschieden, die beeinflussen was beim Verlassen einer *safe cell* geschieht:

- Strategie 1: der GPS-Empfänger wird aktiviert und der zugehörige *distance job* (Kreis) wird überwacht.
- Strategie 2a: der GPS-Empfänger wird aktiviert und ein Positions-Update mit der aktuellen WGS84-Koordinate wird an den Server geschickt.
- Strategie 2b: die Cell-Id der neu betretenen Zelle wird an den Server geschickt.

Eine weitere Strategie folgt einem von Xu und Jacobsen [155] vorgestellten Ansatz. Darin werden zwei effiziente Methoden zur Bestimmung räumlicher Beziehungen zwischen mobilen Objekten beschrieben. Eine Methode besteht darin, den Raum in gleichgroße Gitterzellen aufzuteilen und die Positionen von Zielobjekten den jeweiligen Zellen zuzuordnen. Basierend auf dem vorgestellten Index können dann räumliche Bedingungen (so genannte *n-body constraints*) abgefragt und in drei Klassen unterteilt werden: a) eine Bedingung ist sicher erfüllt, b) eine Bedingung ist sicher nicht erfüllt, c) die Bedingung kann nicht anhand der Gitterzellen überprüft werden.

Basierend auf diesem Ansatz wird anstelle der Gitterzelle die Cell-ID verwendet und die vierte Strategie setzt sich folgendermaßen zusammen:

3. Bereitstellung von Umgebungsmodellen für I-LBS

- Strategie „Server“: die Zielobjekte werden einer von zwei Listen zugeordnet: die *Cell-ID-Liste* enthält alle Zielobjekte, für die momentan ein Unterschreiten von d_p (bzw. ein Überschreiten von d_s) anhand der GSM-Zellen entschieden werden kann. Die *GPS-Liste* enthält alle Zielobjekte, für die diese Überprüfung nicht anhand der Cell-IDs möglich ist.

In einer Simulation wird eine variierende Anzahl von Zielobjekten betrachtet, die sich auf einem Gebiet von $7,5\text{km} \times 5,5\text{km}$ mit einer konstanten Geschwindigkeit von 50km/h über einen Zeitraum von ca. 90 Minuten (5000 Sekunden) bewegen. Die proximity distance beträgt dabei $d_p = 150\text{m}$, die separation distance $d_s = 300\text{m}$ und die borderline tolerance $b = 50\text{m}$. Die durchschnittliche Fläche der GSM-Zellen beträgt $0,25\text{km}^2$. Jedes Zielobjekt bewegt sich mit gleich bleibender Geschwindigkeit auf einer geraden Linie, solange bis es den Rand des Gebietes erreicht. Anschließend wird zufällig eine neue Richtung gewählt. Von Küpper und Treu [90] wurde gezeigt, dass sich die Wahl des Bewegungsmusters beim Vergleich unterschiedlicher Strategien nicht auf die Simulationsergebnisse auswirkt. Die betrachteten Szenarien umfassen $n = 2$ bis 20 Zielobjekte. Auch wenn diese Zahl zunächst gering erscheinen mag: in der Praxis werden nicht immer alle Zielobjekte einer Gruppe gleichzeitig online sein bzw. wird sich nur ein Teil in einer relevanten Entfernung zueinander befinden, so dass ein Location Provider auch immer nur eine begrenzte Anzahl von Zielobjekten überwachen muss.

Die Simulationsergebnisse in Abbildung 3.5(a) zeigen den Anteil der Zeit, in der eine GPS-Positionierung notwendig ist, in Relation zu einer variierenden Anzahl von gleichzeitig zu überwachenden Zielobjekten. In der Referenz-Strategie wird ausschließlich GPS verwendet, dementsprechend beträgt der Nutzungsanteil 100% der Zeit. Dass der Anteil der GPS-Nutzung mit einer größer werdenden Anzahl von Zielobjekten steigt ist offensichtlich, denn mit jedem neu hinzukommenden Zielobjekt reduziert sich der *free movement space*. Die Server-Strategie übertrifft die anderen Strategien für $n < 10$, verschlechtert sich jedoch dann geringfügig. Strategie 1 übertrifft die Strategien 2a und 2b immer mit einer Einsparung von ca. 10% in der GPS-Nutzungsdauer.

Die Effizienz der Strategien hängt wesentlich mit der Größe der GSM-Zellen zusammen. Die Ergebnisse in Tabelle 3.1 zeigen, dass die Effizienz mit kleiner werdender Größe der Zellen deutlich steigt. Es muss zusätzlich betont werden, dass das simulierte Szenario einen extremen Fall darstellt: eine konstante Geschwindigkeit von 50km/h innerhalb einer Stadt liegt weit über der durchschnittlichen Geschwindigkeit eines normalen Zielobjektes und in der Praxis würde man nochmals effizientere Ergebnisse erwarten.

Die offensichtliche Korrelation zwischen der Anzahl n der Zielobjekte und dem Grad der Effizienz betrifft die verschiedensten Arten von ortsbezogenen Diensten. Szenarien für $1 < n < \approx 20$ sind durchaus realistisch, wenn man bedenkt, dass in einer betrachteten Gruppe mit $n \gg 20$ nicht immer alle Zielobjekte eine Distanz zueinander aufweisen, innerhalb derer eine Verfolgung nach dem oben beschriebenen Muster notwendig ist.

Ein weiterer entscheidender Faktor bei der Kombination von Positionierungsverfahren ist die Zeit, die für eine einzelne (vor allem für eine erstmalige) Positionsbestimmung benötigt wird. Die so genannte *time to first fix (TTFF)* nach einem *cold start* eines GPS-Empfängers,

3. Bereitstellung von Umgebungsmodellen für I-LBS

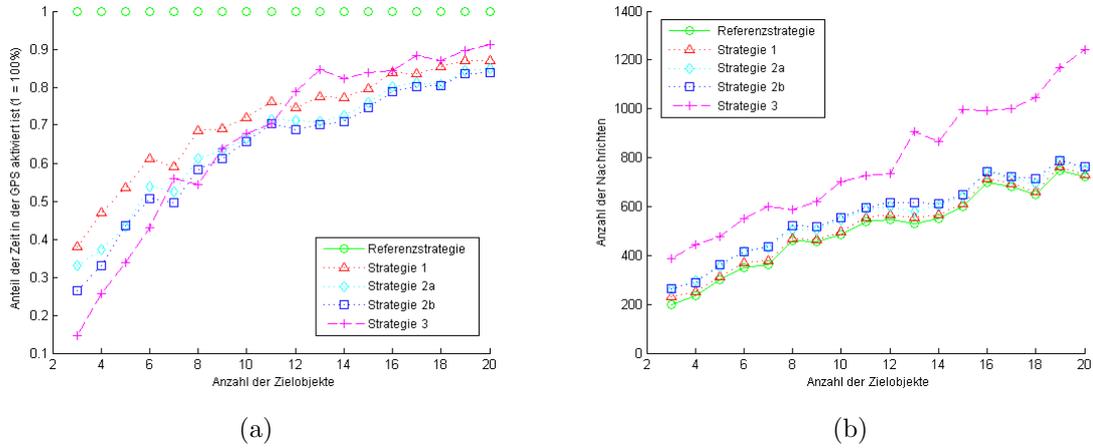


Abbildung 3.5.: Kombination von GPS und GSM-Cell-ID-Positionierung: Simulationsergebnisse. (a) Anteil der Zeit, in der eine GPS-Positionierung notwendig ist. (b) Anzahl der Nachrichten, die zwischen Location Server und Zielobjekten ausgetauscht werden.

Zell-Fläche (Zell-Radius)	$0.22km^2$ ($\approx 500m$)	$0.14km^2$ ($\approx 400m$)	$0.08km^2$ ($\approx 300m$)	$0.03km^2$ ($\approx 200m$)
5 Zielobjekte	60%	62%	70%	80%
10 Zielobjekte	33%	42%	60%	67%
15 Zielobjekte	25%	32%	44%	61%
20 Zielobjekte	13%	25%	40%	57%

Tabelle 3.1.: Anteil der Zeit, in der der GPS-Empfänger ausgeschaltet ist, in Abhängigkeit der Größe der GSM-Zellen für 5, 10, 15 und 20 Zielobjekte (Strategie 2b).

d.h. eine Positionsbestimmung ohne die vorherige Kenntnis von Almanach und Ephemeriden, kann mehrere Minuten in Anspruch nehmen. Die Gültigkeit dieser Daten beträgt eine Woche bzw. vier Stunden und typischerweise werden sie heutzutage auf vielen Endgeräten mit Hilfe von *Assisted GPS* aktuell gehalten. Dazu erfolgt eine Übertragung der Bahndaten z.B. über das Mobilfunknetz. Die Kenntnis aktueller Daten ermöglicht dem GPS-Empfänger einen so genannten *warm start*, der typischerweise 7 - 15 Sekunden in Anspruch nimmt. Ist zusätzlich eine zeitliche Synchronisierung erfolgt, so kann ein GPS-Empfänger nach einem *hot start* innerhalb weniger Sekunden eine Position berechnen. In [36] wird ein neuer Ansatz vorgestellt, bei dem anhand eines Zustandsmodells der GPS-Empfänger im Hintergrund dynamisch ein- und ausgeschaltet wird, um so den Empfänger in einem bestimmten Zustand zu halten. Auf diese Art und Weise kann ein LBS z.B. eine gewünschte TTFB vorgeben, und anhand dieser Anforderung werden die dafür benötigten Informationen im GPS-Empfänger aktuell gehalten.

Abbildung 3.6 zeigt ein Histogramm der GPS-Nutzung verteilt auf fünf verschiedenen lan-

3. Bereitstellung von Umgebungsmodellen für I-LBS

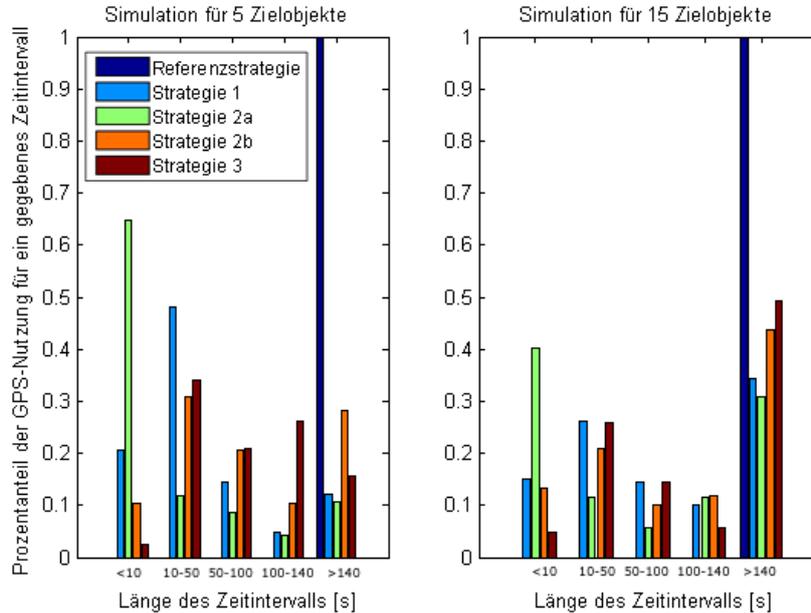


Abbildung 3.6.: Länge von Zeitintervallen, in denen GPS aktiviert ist. Durchschnittliche Fläche der GSM-Zellen: $0,22\text{km}^2$.

ge Zeitintervalle. Auch wenn die verschiedenen Start-Phasen des GPS-Empfängers in der Simulation nicht explizit berücksichtigt wurden, so lässt sich doch erkennen, dass z.B. Strategie 3 in weniger als 5% der Fälle ein Nutzungsintervall von weniger als 10 Sekunden verursacht. Auch Strategie 2b reduziert den Anteil der kurzen Nutzungsintervalle auf 10%. Erhöht sich die Anzahl der Zielobjekte auf $n = 15$, so bleibt ein einmal aktivierter GPS-Empfänger in über 50% der Fälle länger als 140 Sekunden aktiv. Strategie 2a verursacht kurze Nutzungsintervalle in über 65% der Fälle, dies ist jedoch direkt auf die Vorgehensweise der Strategie zurückzuführen. Da alle *proximity*- und *separation*-Ereignisse korrekt und ohne Verzögerung erkannt werden müssen, ist es nicht möglich, diese häufigen GPS-Positionsbestimmungen zu vermeiden.

Um die Grenzwerte für die Nahbereichs- und Trennungserkennung festlegen zu können, muss eine bestimmte Genauigkeit des GPS-Empfängers angenommen werden, beispielsweise eine Genauigkeit von 15m in 95% aller Fälle. Aufgrund der Tatsache, dass der GPS-Empfänger in dem präsentierten Ansatz wechselnd ein- und ausgeschaltet wird, muss ein weiterer Umstand berücksichtigt werden: Die Genauigkeit des *first fix*, d.h. der erstmalig nach dem Einschalten ermittelte Position, weicht zum Teil erheblich von der üblichen Genauigkeit ab. Abbildung 3.7 zeigt die Ergebnisse einer Messung, bei der der GPS-Empfänger eines *htc p3600* vor dem Eingang eines mehrstöckigen Gebäudes insgesamt 63 Mal ein- und ausgeschaltet wurde. Nach jedem Einschalten wurde die erste vom Empfänger als gültig markierte Position gespeichert und anschließend der Empfänger für 60s deaktiviert. Die Messungen zeigen, dass die Genauigkeit des *first fix* zum Teil um mehrere hundert Meter von der tatsächlichen Position (in der Abbildung mit $\text{Ref}_{\text{Accuracy}}$ bezeichnet) abweicht. Zwar

3. Bereitstellung von Umgebungsmodellen für I-LBS

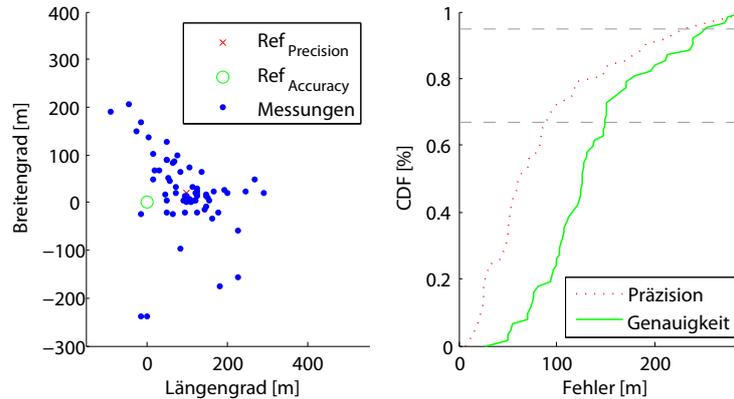


Abbildung 3.7.: Genauigkeit des *first fix* nach dem Einschalten eines GPS-Empfängers in einem *htc p3600*. Zwischen jeder der 63 Messungen wurde der GPS-Empfänger 60s lang deaktiviert.

reduziert sich dieser Fehler normalerweise bereits nach wenigen Sekunden auf ein gängiges Maß, jedoch muss dieser Umstand bei einer praktischen Implementierung berücksichtigt werden. Bei einer prototypischen Implementierung des vorgestellten Ansatzes wurde eine GPS-Position erst dann als gültig anerkannt, wenn mehrere aufeinander folgende Messungen nicht weiter als 20m auseinander lagen. Da die Positionen in der vorgestellten Simulation aus einem Bewegungsmuster resultieren und in diesem Sinne keinem Fehler unterliegen, wurde in der Simulation jeweils die erste Position verwendet.

Zur Evaluierung der oben vorgestellten Strategien muss zuletzt auch die Anzahl der zwischen Zielobjekt und Location Server übertragenen Nachrichten betrachtet werden. Die Anzahl sollte möglichst gering sein, da jede Übertragung Energie kostet und die Bandbreite drahtloser Verbindungen begrenzt ist. Abbildung 3.5(b) zeigt, dass Strategie 1 gegenüber der Referenzstrategie (*Dynamic Centered Circles* [90]) nicht signifikant mehr Nachrichten verursacht. Die Steigerung der Nachrichtenanzahl bei den anderen Strategien resultiert aus den zusätzlichen Anfragen (*polling*) durch den Server: falls das Zielobjekt momentan nur die Cell-Id-Positionierung nutzt, so antwortet es auf ein Polling mit der aktuellen Cell-ID. Ist dies nicht ausreichend so fordert der Server mit einer zusätzlichen Nachricht einen GPS-Fix an. Das erhöhte Nachrichtenaufkommen bei Strategie 3 lässt sich dadurch erklären, dass im Gegensatz zu den Client-basierten Strategien hier jeweils nur eine einzelne Zelle verwendet wird. Der große Vorteil der Zell-basierten Strategien kommt vor allem dann zum tragen, wenn die Zielobjekte über einen längeren Zeitraum stehen bleiben (z.B. innerhalb eines Gebäudes verbleiben) oder sich mit wesentlich geringeren Geschwindigkeiten bewegen (z.B. Fußgänger).

Die hier vorgestellten Strategien zur Reduzierung der GPS-Nutzungsdauer wurden in einer vergleichsweise sehr dynamischen Umgebung simuliert. In der Praxis lässt sich durchaus noch eine weitere Steigerung der Effizienz erwarten. Strategie 3 ist in vielen Fällen vorteilhaft, da sie eine sehr gute bis akzeptable Reduzierung der GPS-Nutzung bietet und gleich-

3. Bereitstellung von Umgebungsmodellen für I-LBS

zeitig nur sehr selten kurze GPS-Zeitintervalle erfordert. Diese Eigenschaft ist beispielsweise auch im innerstädtischen Bereich von Häuserschluchten bzw. allgemein bei schlechten DOP-Werten (ein Maß für die Qualität der Satellitenkonstellation) von Vorteil. Die Client-basierten Strategien 2a und 2b eignen sich im Vergleich zueinander gleich gut, Strategie 2b bietet jedoch etwas bessere Ergebnisse in Bezug auf die Länge der GPS-Zeitintervalle. Und auch die einfachste Strategie 1 bietet bereits eine effiziente Möglichkeit zur Reduzierung der GPS-Nutzungsdauer.

Eine Anwendung in der Praxis setzt folgende Gegebenheiten voraus: zum einen werden Informationen über die Zellen des Mobilfunknetzes benötigt. Dabei sind neben den Koordinaten der Basisstationen in optimaler Weise auch die *best server estimates* der Cell-IDs bekannt. Diese könnten über eine entsprechende Schnittstelle des Mobilfunknetzbetreibers bereit gestellt werden. Zum anderen ist es erforderlich, dass auf dem Endgerät alle Anwendungen nicht direkt auf die Positionierungsverfahren zugreifen, sondern ihre Anfragen an einen Vermittler stellen, der das jeweils passende Positionierungsverfahren aktiviert. Auch hier könnte dies durch eine enge Zusammenarbeit zwischen Dienstanbieter und Mobilfunknetzbetreiber realisiert werden. Großes Potential liegt mit Sicherheit in den zukünftigen Generationen leistungsfähiger *Subscriber Identity Modules (SIM)*. Bereits heute bieten diese Module die Möglichkeit, Programme innerhalb eines geschützten und exklusiv vom Mobilfunknetzbetreiber steuerbaren Bereiches auszuführen. Wenn man bedenkt, dass die Struktur und genaue Konfiguration des Mobilfunknetzes ein vom Betreiber gern gehütetes Geheimnis ist, so könnte mit Hilfe der SIMs eine Schnittstelle geschaffen werden, die einerseits die effiziente und effektive Nutzung der Cell-ID-Positionierung ermöglicht, andererseits aber auch netzspezifische Daten angemessen vor der Öffentlichkeit verschattet. Denkbar ist auch, dass in Zukunft der Mobilfunknetzbetreiber stärker als Location Provider auftritt und die Kombination der Positionierungsverfahren gegenüber den externen Diensten und Anwendungen koordiniert. In einem solchen Szenario könnten dann vor allem auch Dienste profitieren, die nicht nur auf die aktuellen Positionen der Zielobjekte angewiesen sind, sondern auch auf eine Historie. Ananthanarayanan et al. [6] beschreiben dazu einen vielversprechenden Ansatz, mit dem sich die Spuren mehrerer Zielobjekte gruppieren und miteinander vergleichen lassen. Eine ausführliche Diskussion über die Positionierung in GSM-Netzen gibt Deblauwe [34].

Kombination von Indoor Positionierungsverfahren: Innerhalb von Gebäuden stehen ebenfalls verschiedenste Positionierungsverfahren zur Verfügung, und das Prinzip der zuvor diskutierten Kombination lässt sich auch auf den Indoor-Bereich übertragen. Ein dynamischer Wechsel kann dabei nicht nur im Sinne eines geringen Stromverbrauchs vorteilhaft sein. Auch lassen sich damit z.B. eine bessere Verfügbarkeit von Positionsinformationen, bessere Robustheit gegen Störeinflüsse sowie Mechanismen zur automatischen Kalibrierung eines Positionierungsverfahrens realisieren.

Viele Indoor-Positionierungsverfahren liefern im Gegensatz zum GPS keine auf wenige Meter genaue Position, sondern bestimmen den Aufenthaltsort z.B. anhand unterschiedlicher Räume. Für diese Systeme lässt sich der oben beschriebene Mechanismus für proakti-

3. Bereitstellung von Umgebungsmodellen für I-LBS

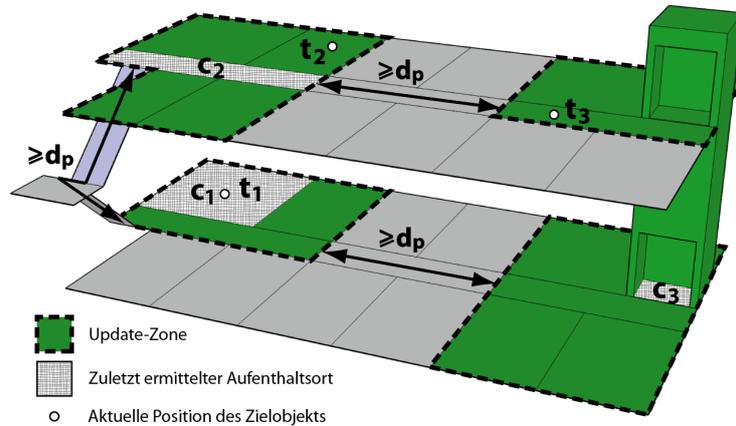


Abbildung 3.8.: Zonen für die Überwachung von Zielobjekten in Gebäuden.

ve ortsabhängige Benachrichtigung vergleichsweise einfach implementieren: setzt man das Konzept der Zelle einzelnen Räumen gleich, so lassen sich *safe rooms* identifizieren, innerhalb derer sich Zielobjekte frei bewegen können, ohne dabei in die Nähe anderer Objekte oder Bereiche von Interesse zu kommen. Abbildung 3.8 zeigt ein Beispiel für ein Gebäude mit zwei Stockwerken, auf denen sich drei Zielobjekte t_1 , t_2 und t_3 befinden. Die Räume c_1 , c_2 und c_3 repräsentieren die dem Location Provider zuletzt bekannte Position. Anhand der *walking distance* zwischen Räumen und gegebener proximity distance d_p kann für jedes Zielobjekt die Menge von *safe rooms* (Update-Zone, grün) berechnet werden. Erst wenn ein Zielobjekt seine Update-Zone verlässt, kann ein potenziell interessantes Ereignis aufgetreten sein. Dieser Ansatz lässt sich weiter verfeinern, wenn in dem Gebäude weitere Positionierungsverfahren verfügbar sind. So kann beim Verlassen einer Update-Zone zunächst auf ein genaueres Positionierungsverfahren gewechselt werden, ähnlich einem Wechsel von GSM-Cell-Id zu GPS. Alternativ können auch Inertialnavigationssysteme (*inertial navigation systems*) eingesetzt werden, um die Bewegung am Rande einer Update-Zone zu überwachen.

Eine Energie-effiziente Überwachung der Zielobjekte ist auch hier nur möglich, wenn innerhalb einer Update-Zone die Positionierung ohne umfangreiche Kommunikation mit einem Server geschieht. Wie in [79] beschrieben, kann dies z.B. über Signalstärke-Muster (RSS pattern) realisiert werden. Dazu wird dem Zielobjekt dynamisch ein RSS-Muster mitgeteilt, das die Menge der aktuell in der Update-Zone enthaltenen Räume repräsentiert. Anhand dieses Fingerprints und eines Bayes-Schätzers kann das Zielobjekt anschließend ermitteln, wann die Update-Zone wieder verlassen wird [77].

Sehr genaue und präzise Indoor-Positionierungsverfahren gehen meist mit hohen Kosten für die Installation einher, was eine großflächige Verfügbarkeit erschwert. So existieren heutzutage bereits Systeme, die über Ultrabreitband-Signale (Ultra Wide Band, UWB) anhand von Signallaufzeit- und Winkelmessungen eine dreidimensionale Genauigkeit in der Größenordnung von 15cm bieten. Die zeitliche Auflösung beträgt bis zu ca. 10Hz bei einer Verzögerung von ca. 200ms. Solche Systeme sind jedoch vergleichsweise teuer, da eine

3. Bereitstellung von Umgebungsmodellen für I-LBS

dedizierte Verkabelung benötigt wird und die Anzahl der beteiligten Infrastrukturkomponenten linear mit der zu überwachenden Fläche wächst. Auch hier kann eine Kombination mit anderen Positionierungsverfahren sinnvoll sein, um dadurch in weniger kritischen Bereichen eines Gebäudes eine ungenauere Positionierung über ein günstigeres Verfahren zu realisieren. Einen Überblick über UWB-Positionierung geben Gezici et al. [50].

Zuletzt können zwei parallele Positionierungssysteme auch dahingehend kombiniert werden, dass das genauere Verfahren als Referenz bei der Kalibrierung des ungenaueren Verfahrens zum Einsatz kommt. Ein Beispiel dafür ist die automatische Kalibrierung von WLAN-Positionierungssystemen anhand von GPS-Referenzpositionen (siehe [52, 51]). Anhand einer simultanen Auswertung von ermittelter Position und Referenz-Position kann die Kalibrierung eines Positionierungssystems automatisiert angestoßen werden und so der manuelle Aufwand minimiert werden. Dadurch lassen sich beispielsweise großflächige WLAN-Positionierungssysteme an Flughäfen oder Unternehmensstandorten schnell und effektiv kalibrieren. In solchen Umgebungen existieren zahlreiche Fahrzeuge, die über den Tag eine große Fläche befahren. Bereits wenige Kalibrierungs-Module in ein paar Fahrzeugen können ausreichen, um innerhalb kurzer Zeit ein großes Areal zu kalibrieren. Dazu muss jedes Kalibrierungs-Modul in der Lage sein, notwendige Messungen (z.B. Signalstärke-Pegel) zu sammeln und diese an das zu kalibrierende System zu übergeben. Wird das Kalibrierungs-Modul in einen Bereich hinein bewegt, in dem bisher noch keine Kalibrierungsdaten vorliegen und die Genauigkeit des zu kalibrierenden Systems zu sehr von der mittels GPS ermittelten Referenz-Position abweicht, wird ein Kalibrierungsvorgang gestartet. Während der laufenden Kalibrierung kann anhand der Referenz-Positionen überwacht werden, ob das Modul bewegt wird und die Kalibrierung gegebenenfalls unterbrochen werden. Manche Positionierungssysteme sind in der Lage, neben der ermittelten Position auch eine Abschätzung über den zu erwartenden Fehler in der Berechnung abzugeben. Auch diese Information kann von einem Kalibrierungsmodul verwendet werden, sofern eine Messung an einem bestimmten Ort nach der Kalibrierung eine Verkleinerung des Fehlers verspricht.

Diskussion: Koordinatensysteme für heterogene Positionierungsverfahren Existierende Positionierungsverfahren für Indoor-Umgebungen werden bislang auf der Grundlage von lokalen Karten und lokalen Koordinatensystemen realisiert. Die gleichzeitige Verwendung von Positionsdaten aus heterogenen Positionierungsverfahren bedingt, dass die Transformations-Regeln zur Umrechnung zwischen den Koordinaten bekannt sind. Im Falle von geometrischen Koordinaten werden die Transformations-Matrizen (vgl. 3.3.1) benötigt, im Falle von symbolischen Koordinaten muss eine Zuordnung zwischen den natürlichsprachlichen Bezeichnern der Orte und ihren geometrischen Koordinaten erfolgen.

Die ausschließliche Verwendung von globalen Referenzsystemen wie z.B. dem WGS84 ist für die Nutzung von Kartenmaterial und Geo-Daten im Outdoor-Bereich eine komfortable und einfache Möglichkeit. Jedoch eignet sich diese nicht unbedingt für die Verarbeitung von Ortsinformationen in Gebäuden, wie folgende Beispiele zeigen:

- Karten von Stockwerken repräsentieren von Natur aus zweidimensionale Flächen. Eine Position auf einer solchen Karte lässt sich leicht durch eine xy-Koordinate angeben

3. Bereitstellung von Umgebungsmodellen für I-LBS

und auch visualisieren. Eine Umrechnung in ein globales elliptisches Koordinatensystem ist zwar möglich, aber nicht immer sind exakte Stützpunkte mit global bekannten Koordinaten vorhanden und eine Umrechnung ist auch immer fehlerbehaftet.

- Einfache xy-Koordinaten haben den Vorteil, dass sie sich unmittelbar mit gängigen Algorithmen zur effizienten Speicherung und Suche verarbeiten lassen.
- Grafische Anwendungen zur Visualisierung von Indoor-Umgebungen sind einfacher mittels Kartesischer Koordinaten zu realisieren, zumal viele Programm-Bibliotheken und Tools ihre Schnittstellen darauf aufsetzen.
- Höhenangaben sind (auch durch Menschen) leichter zu interpretieren, wenn sie in einem lokalen Kontext vorliegen.

Im Gegensatz zu den Berechnungen innerhalb eines Gebäudes sind die globale Referenzsysteme eine notwendige Voraussetzung, um lokalen Positionsdaten und Modelle in einem größeren Maßstab miteinander in Bezug zu setzen. Betrachtet man beispielsweise mehrere Gebäude, die über eine Stadt verteilt einer Organisation gehören, so können die einzelnen Grundrisse, Pläne und auch Daten aus Positionierungssystemen über ein globales Referenzsystem mit entfernten Standorten in Verbindung gebracht werden.

Die Nutzung von rein rasterbasierten Bilddaten in einem Positionierungssystem kann jedoch ein Problem darstellen, wie folgendes Beispiel zeigt: Angenommen, ein lokales Positionierungsverfahren verwendet als Datengrundlage ein Rasterbild mit einer Auflösung von 1800x2700 Pixeln. Der darin befindliche Plan repräsentiert einen Flughafen mit einer Fläche von 3,6x5,4km. Zwar sind Detailansichten von einzelnen Gebäuden vorhanden, jedoch beziehen sich deren Lage-Informationen auf wenige Referenzpositionen im Gesamtplan. Daraus resultiert, dass jede Positionsangabe auf dem Gesamtplan unabhängig von der Genauigkeit des Positionierungssystems nur auf maximal 2m akkurat ist. Positionsangaben in den Detailansichten können genauer sein, allerdings nur, wenn zuvor eine Kalibrierung mit Bezug auf die höher auflösende Karte durchgeführt wurde. Eine Genauigkeit von 2m mag zunächst ausreichend erscheinen, jedoch kann diese Distanz oftmals darüber entscheiden, ob ein Zielobjekt innerhalb oder außerhalb eines bestimmten Raumes platziert wird. Aus diesem Grund sind vektorbasierte Daten vorzuziehen, denn sie erlauben einen wesentlich flexibleren Einsatz. Aus der Sicht eines einzelnen Positionierungssystems mit zahlreichen Karten lokaler Gebäude und Stockwerke kann es sinnvoll sein, zunächst alle Karten – beispielsweise anhand ihrer linken unteren und rechten oberen Ecke – in ein gemeinsames lokales Koordinatensystem einzuordnen. Im Falle von zweidimensionalen Karten sind dafür nur vier Transformationsparameter notwendig: zwei Verschiebungen, ein Maßstab, eine Drehung. Für das gemeinsame Koordinatensystem können dann anhand einiger weniger global bekannter Stützpunkte die Transformationsparameter zur Umrechnung in ein globales Referenzsystem bestimmt werden.

Letztendlich hängt zukünftig vieles davon ab, wie die Anbieter von lokalen Positionierungsverfahren ihre Schnittstellen für den Zugriff auf Positionsdaten durch Dritte öffnen.

3. Bereitstellung von Umgebungsmodellen für I-LBS

Durch eine zu enge Kopplung zwischen einem Positionierungssystem und einer speziellen Anwendung wird die Entwicklung gebremst und der Zugriff aus anderen Anwendungen heraus erschwert. Konkret werden in Zukunft drei verschiedene Eigenschaften eines Positionierungssystems benötigt:

1. Die von einem Indoor-Positionierungssystem ermittelten Koordinaten müssen mit zusätzlichen Angaben über das verwendete Referenzsystem und entsprechenden Stützpunkten angereichert werden. Nur so lassen sich die Positionsdaten ohne Weiteres in anderen Umgebungen weiter verarbeiten. Analog zur dem im Jahr 2003 veröffentlichten *Java Specification Request 179* könnten dann von einer Anwendung Positionsdaten mit bestimmten Kriterien angefordert werden und die Auswahl des tatsächlichen Positionierungssystems erfolgt transparent im Hintergrund (unter Berücksichtigung von vorhandenen Datenschutzrichtlinien und -präferenzen).
2. Die Rohdaten, die zur Berechnung einer Position geführt haben, sollten ebenfalls verfügbar gemacht werden. Dadurch wird ermöglicht, dass sich verschiedene Positionierungsverfahren besser miteinander kombinieren lassen.
3. Die Modellierung eines Gebäudes sollte außerhalb des Positionierungssystems geschehen bzw. klar von der Positionierung getrennt sein. Nur so können beispielsweise Anfragen nach der *walking distance* zwischen Positionsangaben aus unterschiedlichen Positionierungssystemen beantwortet werden.

In diesem Abschnitt wurden verschiedene Aspekte von Positionsdaten aus heterogenen Positionierungsverfahren diskutiert. In einem einfachen Fall lassen sich diese Positionsdaten auf einer simplen Karte darstellen oder auch einer symbolischen Koordinate zuordnen. In dem Moment, in dem jedoch verschiedene Positionsdaten miteinander in Bezug gesetzt werden sollen (wie es unter anderem für die Indoor-Navigation nötig ist), werden zusätzliche Informationen über die Erreichbarkeit und topologische Entfernung benötigt. Der folgende Abschnitt befasst sich mit der Erzeugung von Wegenetzen und geht dabei auf verschiedene Möglichkeiten zur Beschreibung der Begehbarkeit von Räumen ein.

3.4. Erzeugung von Wegenetzen

Um die Positionsdaten verschiedener Zielobjekte und Orte von Interesse (POI) in einem Gebäude miteinander in räumliche Beziehung zu setzen, werden Informationen über die Begehbarkeit der Umgebung benötigt. Ein Akteur in der Rolle des Modell-Erzeugers (vgl. Abbildung 2.1) hat bei der Erzeugung eines Umgebungsmodells verschiedene Möglichkeiten, die Begehbarkeit des Gebäudes zu modellieren. Der Begriff Begehbarkeit bezeichnet dabei für zwei beliebige Orte die uni- oder auch bidirektionale Eigenschaft, untereinander durch eine natürliche Person zu Fuß erreichbar zu sein. Die Erreichbarkeit richtet sich sowohl nach physischen Barrieren (zwei durch eine Wand voneinander getrennte Orte sind nicht direkt untereinander erreichbar), lokalen Bestimmungen und Regelungen (für einen

3. Bereitstellung von Umgebungsmodellen für I-LBS

Besucher sind bestimmte Türen nicht passierbar), als auch anderen Eigenschaften eines Gebäudes (z.B. Laufrichtung einer Rolltreppe oder Breite eines Durchgangs).

Alle Informationen über die Begehbarkeit eines Gebäudes werden in einem *Wegenetz* gespeichert, mittels dessen der Modell-Provider die entsprechenden Anfragen des I-LBS Providers beantworten kann.

Ein Wegenetz kann entweder als reiner Graph oder durch eine Menge miteinander verbundener Flächen (auch als *Navigation Mesh* bezeichnet) ausgedrückt werden. Während ein Wegpunktgraph lediglich einzelne erreichbare Punkte im Raum und deren direkte Erreichbarkeit ausdrückt, enthält ein Navigation Mesh zusätzlich Informationen über die begehbaren Bereiche (Polygone) eines Ortes und die Erreichbarkeit zwischen den Bereichen. Der Vorteil eines reinen Graphen liegt in einer zumeist schnellen Berechnung kürzester Wege sowie einem vergleichsweise geringen Datenumfang für die Speicherung. Ein Navigation Mesh hingegen bietet zusätzliche Möglichkeiten. So kann beispielsweise die Größe der Polygone im Navigation Mesh dazu verwendet werden, um zusätzlich die maximale Breite von Durchgängen oder Passagen zu bestimmen. Ein weiteres populäres Anwendungsbeispiel sind Computerspiele, in denen z.B. zwei vom Computer gesteuerte Charaktere aufeinander treffen. Um sich gegenseitig auszuweichen, muss anhand der Ausdehnung des umliegenden Raumes ein alternativer Weg bestimmt werden.

Bevor im Folgenden die verschiedenen Ansätze im Detail diskutiert werden, gilt es zunächst, einige generelle Eigenschaften von Wegenetzen zu beachten. Das der Wegewahl in Gebäuden zugrunde liegende Problem des *Path Finding* tritt auch in vielen anderen Anwendungsbereichen auf, insbesondere im Bereich der Künstlichen Intelligenz zur autonomen Steuerung von Objekten in 2- oder 3-dimensionalen virtuellen Welten. Das Path Finding basiert letztendlich immer auf der Berechnung von Wegen auf Graphen, trotzdem muss zwischen zwei verschiedenen Teilproblemen unterschieden werden:

- Als Suchalgorithmen stehen heutzutage viele verschiedene Ansätze wie z.B. Breitensuche, Tiefensuche, Dijkstras Algorithmus [38] oder A* [55] zur Verfügung und müssen für den jeweiligen Anwendungsfall angepasst werden.
- Mindestens genau so entscheidend für eine effektive und effiziente Suche ist jedoch auch die Organisation des zugrunde liegenden *Suchraumes*. Knoten in den verwendeten Graph-Strukturen können einzelnen Wegpunkte, Übergänge zwischen Polygonen oder ganze Flächen repräsentieren.

Suchalgorithmen und Optimierungen existieren für viele verschiedene Anwendungsgebiete. Bei der Verwendung des selben Suchraumes unterscheidet sich die Berechnung von kürzesten Wegen letztendlich nur in der Laufzeit und der Speicherkomplexität des jeweiligen Algorithmus, die Länge eines berechneten Weges ist (abgesehen von Algorithmen, die auf reinen Schätzungen beruhen) identisch. Neben statischen existieren auch dynamische Verfahren, die die erneute Berechnung von Wegen aufgrund von lokalen Änderungen im Graphen effizient gestalten (siehe Stentz [135, 136]).

Die Beschaffenheit des Suchraumes entscheidet darüber, welche Anfragen grundsätzlich beantwortet werden können und welche nicht. Angenommen ein Wegenetz besteht z.B. nur

3. Bereitstellung von Umgebungsmodellen für I-LBS

aus einzelnen Punkten, zwischen denen die Entfernung und direkte Erreichbarkeit gespeichert wird. Dann können zwar kürzeste Wege berechnet werden, jedoch ist es nicht möglich, die maximale Breite eines Weges zu berechnen. Dazu wären zusätzliche Informationen über die genaue Lage und Form von Wänden, Türen etc. notwendig.

Bei der Erzeugung eines Wegenetzes stellt sich daher die Frage, nach welchen Kriterien Wegpunkte bzw. Bereiche gewählt werden sollen. Die Qualität eines einzelnen Wegpunktes oder Bereiches lässt sich in der Regel nicht direkt beurteilen, jedoch kann das Wegenetz als Ganzes bewertet werden und sollte dabei besonders in Bezug auf Abdeckung, Begehbarkeit, Weglänge und Komplexität optimal sein:

1. **Optimale Abdeckung:** Das Wegenetz sollte eine möglichst große Abdeckung haben, d.h. möglichst alle Orte im Gebäude einschließen und dadurch gewährleisten, dass jeder tatsächlich begehbare Ort im Gebäude auch mit dem Wegenetz verbunden ist.
2. **Natürliche Wege:** Die durch das Wegenetz ausgedrückten Wege sollten der natürlichen Bewegung eines Menschen möglichst gut entsprechen. Dazu zählen beispielsweise, dass Wege in der Mitte eines Flures verlaufen und nicht direkt an der Wand oder dass ein um eine Ecke führender Weg ebenfalls mit einem gewissen Abstand zur Wand verläuft. Diese Eigenschaft ist besonders dann wichtig, wenn die Wege unmittelbar in einem Navigationssystem, z.B. zur Visualisierung des Weges, verwendet werden sollen.
3. **Genauere Weglänge:** Die Länge eines mit Hilfe des Wegenetzes berechneten kürzesten Weges zwischen zwei Orten sollte möglichst genau der realen Weglänge entsprechen. Dieses Kriterium mag zunächst trivial erscheinen, ist diese Eigenschaft doch für viele bekannte Graphen (z.B. auf einem Straßennetz) gegeben. Aufgrund der zahlreichen verschiedenen möglichen Wege durch große Hallen und Räume kann die genaue Weglänge jedoch je nach verwendetem Wegenetz abweichen.
4. **Geringe Komplexität:** Für eine effiziente Nutzung des Wegenetzes sollte es eine möglichst geringe Komplexität aufweisen, d.h. aus möglichst wenigen Knoten und Kanten bzw. Polygonen bestehen. Dies wird besonders dann relevant, wenn ein Dienst nicht mehr nur einzelne Routenberechnungen anfragt, sondern beispielsweise die dynamischen Positionen einer Vielzahl von mobilen Zielobjekten überwacht und miteinander in Beziehung gesetzt werden sollen.

Dies sind vier teilweise gegenläufige Kriterien, die bei der Erzeugung eines Wegenetzes nicht alle zugleich vollständig erfüllt werden können. Geht man davon aus, dass Lage, Form und Verbundenheit von Räumen in einem Gebäude bekannt sind, so kann ein Wegenetz auf unterschiedliche Weise erzeugt werden und dabei unterschiedliche Formen annehmen. Bei einem Wegenetz für Straßen fällt die Auswahl von Wegpunkten vergleichsweise einfach aus, indem man beispielsweise an jeder Kreuzung einen Wegpunkt einfügt und die Länge

3. Bereitstellung von Umgebungsmodellen für I-LBS

der Kanten zwischen entsprechend miteinander verbundenen Wegpunkten berechnet (eine Ausnahme bilden große Kreuzungen mit mehrspurigen Straßen, hier sind unter Umständen mehr Wegpunkte notwendig). Da sich Personen innerhalb eines Gebäudes jedoch nicht auf vorgegebenen Straßen sondern auf freien Flächen bewegen können, müssen andere Kriterien für die Auswahl von Wegpunkten gefunden werden.

Im einfachsten Fall besteht ein Gebäude ausschließlich aus konvexen Räumen, d.h. dass die Verbindungsstrecke zwischen zwei beliebigen Punkten eines Raumes immer vollständig innerhalb dieses Raumes liegt. In diesem Fall könnte das Wegenetz sehr einfach erzeugt werden, indem an jeder Verbindungstür zwischen zwei Räumen ein Wegpunkt eingefügt wird und die Wegpunkte eines Raumes untereinander vollständig untereinander vermascht werden. Dadurch ist sichergestellt, dass von jedem Ort im Gebäude mindestens ein Wegpunkt auf geradlinigem Weg erreicht werden und von dort ein kürzester Weg berechnet werden kann. Ein noch einfacherer Ansatz besteht darin, nur die Mittelpunkte der Räume als Wegpunkte in einen Wegpunktgraphen einzufügen und direkt über eine Tür miteinander verbundene Räume in diesem Graph miteinander zu verbinden. Allerdings lässt sich anhand eines solchen Wegpunktgraphen der Weg nicht zufriedenstellend visualisieren.

Ein Wegpunktgraph lässt sich abbilden als gerichteter und gewichteter Graph $G = (V, E)$, die Menge der Knoten $V = \{v_1, \dots, v_n\}$ entspricht den Wegpunkten und eine Kante $e_{ij} = (v_i, v_j) \in E$ drückt aus, dass v_i und v_j direkt untereinander erreichbar (begehrbar) sind. Betrachtet man beispielsweise die in Abbildung 3.4 dargestellten beiden Räume, die durch einen breiten Durchgang miteinander verbunden sind, so könnte eine einzelne Kante (3.9(a)) oder auch mehrere Kanten (3.9(b)) in den Wegpunktgraph eingefügt werden. Für ein Navigation Mesh könnte die Verbindung zwischen den beiden Räumen durch durch ein zusätzliches Polygon in 2D oder auch 3D modelliert werden (3.9(c) und 3.9(d)).

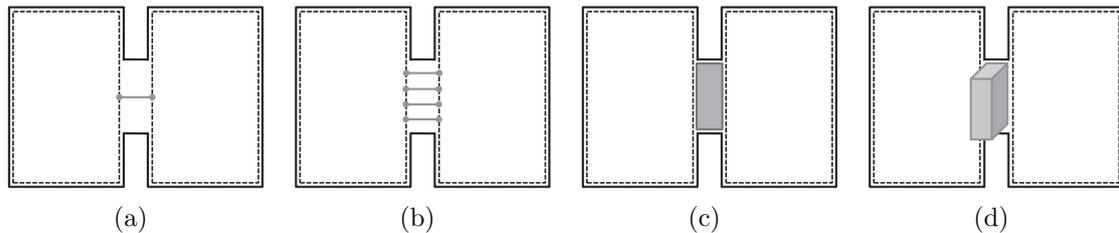


Abbildung 3.9.: Schematische Darstellung von zwei Raumpolygonen (gestrichelte Linie), die (a) über zwei Wegpunkte, (b) über mehrere Wegpunkte miteinander verbunden sind. (c) Verbindung zweier Räume über ein zweidimensionales oder (d) ein dreidimensionales Polygon.

Ein grundsätzliches Problem bei der Erzeugung von Wegenetzen stellt sich jedoch im Falle von konkaven (nach innen gewölbten) Räumen bzw. Polygonen. In einem konkaven Raum können die Ein- und Ausgänge nicht mehr einfach untereinander verbunden werden, sondern es müssen zusätzliche Wegpunkte innerhalb des Raumpolygons gewählt werden. Die Art und Weise, wie diese Wegpunkte gewählt werden beeinflusst maßgeblich die Qualität des erzeugten Wegenetzes. Abbildung 3.10 zeigt schematisch vier Räume, die über

3. Bereitstellung von Umgebungsmodellen für I-LBS

verschiedene Türen miteinander verbunden sind. Der Raum $R2$ ist konkav und wurde in zwei konvexe Teile $R2 - 1$ und $R2 - 2$ unterteilt. Eine gute Unterteilung mag für ausschließlich rechtwinklige Polygone mit wenigen Eckpunkten einfach zu finden sein, bei größeren Räumen mit vielen konkaven Einbuchtungen, beliebigen Winkeln oder auch Rundungen stellt dies jedoch kein triviales Problem dar.

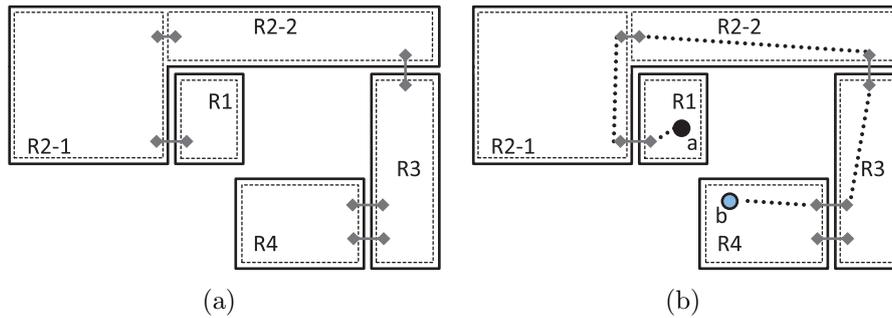


Abbildung 3.10.: (a) Schematische Darstellung von vier Räumen, die über Wegpunkte miteinander verbunden sind. Der konkave Raum $R2$ ist in zwei konvexe Teile $R2 - 1$ und $R2 - 2$ unterteilt. (b) Kürzester Weg zwischen den beiden Orten a und b .

Im Folgenden werden daher verschiedene Ansätze zur Erzeugung eines Wegenetzes für beliebige (d.h. sowohl konvexe als auch konkave) Polygone vorgestellt. Die betrachteten Algorithmen ermöglichen dabei zunächst nur die jeweilige Berechnung von Wegenetzen für einzelne Raumpolygone. Die resultierende Menge von Wegenetzen muss dann in einem zweiten Schritt zu einem Wegegraphen zusammengefasst werden, der das gesamte Gebäude abdeckt. Dies wird im Abschnitt 3.5 behandelt. Bei allen verwendeten Polygonen wird davon ausgegangen, dass es sich um einfache Polygone handelt.

Definition (Einfaches Polygon). Ein Polygon \mathcal{P} heißt *einfach*, wenn sich die Kanten des Polygons nicht schneiden. Eine *Diagonale* ist eine Linie, welche zwei Eckpunkte von \mathcal{P} verbindet und innerhalb von \mathcal{P} liegt.

Viele Datenquellen – insbesondere ältere CAD-Dateien oder andere elektronische Baupläne – enthalten oftmals keine einheitlichen und konsistenten Informationen über die Lage und Form von Räumen, beispielsweise wenn die Wände eines Raumes nicht über ein geschlossenes einfaches Polygon, sondern durch viele einzelne Linien dargestellt werden, die möglicherweise auch noch auf mehreren unterschiedlichen Ebenen im Dokument gespeichert sind. Zur Erzeugung von Wegenetzen müssen diese Daten zuerst in ein einheitliches Format gebracht werden, was in der Praxis mit einem nicht unerheblichen Aufwand verbunden sein kann.

Es gibt grundsätzlich zwei unterschiedliche Herangehensweisen, um Wegenetze anhand bekannter Raum-Polygone zu erzeugen: entweder betrachtet man ausschließlich den Rand

3. Bereitstellung von Umgebungsmodellen für I-LBS

eines Polygons (*randorientiert*), oder das Wegenetz wird anhand der vom Polygon umschlossenen Fläche generiert (*flächenorientiert*). Die randorientierten Algorithmen verwenden zur als Berechnungsgrundlage lediglich die Eckpunkte des Polygons und erzeugen das Wegenetz in Bezug auf diese Punkte. Beispiele dafür sind:

- Corner Graph
- Mittiger Wegpunktgraph
- Mediale Achse
- Straight Skeleton
- Konvexe Partitionierung
- Triangulierungen

Flächenorientierte Algorithmen unterscheiden sich dahin gehend, dass zunächst potenzielle Wegpunkte auf der Fläche des Polygon verteilt werden (z.B. in Form eines regelmäßigen Gitters) und dann ausgehend von einem Startpunkt sukzessive für benachbarte Punkte überprüft wird, ob tatsächlich eine Verbindung möglich ist. Beispiele dafür sind:

- Gitterförmige Wegenetze
- Quadrees
- Kreis-basierte Wegenetze

Diese Algorithmen können auf einfache Art und Weise sehr feinmaschige Wegpunktgraphen erzeugen. Ein Nachteil der flächenorientierten Ansätze ist jedoch, dass der resultierende Graph nicht unbedingt die gesamte Fläche des Raumes abdeckt.

Beinhaltet das resultierende Wegenetz neben einem Wegpunktgraphen auch begehbare Flächen, so erhält man ein Navigation Mesh. Wenngleich in der Datenhaltung komplexer, so unterstützen Navigation Meshes zusätzliche Anfragen, wie weiter unten vorgestellt wird.

Nachfolgend werden zunächst randorientierte Algorithmen vorgestellt, gefolgt von flächenorientierten Ansätzen und Navigation Meshes.

3.4.1. Randorientierte Algorithmen zur Erzeugung von Wegenetzen

Angenommen, es soll ein Wegenetz für eine Umgebung berechnet werden, in der sowohl sehr schmale und lang gezogene Gänge mit verschiedenen Abzweigungen und Kreuzungen, als auch viele große und weitläufige Räume mit konkaven Ecken existieren. Oder anders ausgedrückt: die Form der Räume in einem Gebäude ist a priori nicht bekannt und das zugehörige Polygon eines Raumes charakterisiert den begehbaren Bereich. Es sollen nun Wegpunkte auf oder innerhalb des Polygons bestimmt werden, die miteinander verbunden ein Wegenetz bilden und die Begehbarkeit der Umgebung widerspiegeln.

3. Bereitstellung von Umgebungsmodellen für I-LBS

Die im Folgenden vorgestellten *randorientierten Algorithmen* haben gemeinsam, dass sie entweder nur bestimmte Eckpunkte des bestehenden Polygons auswählen oder neue Punkte in Abhängigkeit eines oder mehrerer Eckpunkte erzeugen. Die resultierende Anzahl von Wegpunkten liegt dabei in der Größenordnung der Anzahl von Polygoneckpunkten.

Corner Graph: Ein vergleichsweise einfacher Ansatz zur Erzeugung eines Wegenetzes ist der Corner Graph [142]. Ein Corner Graph ist ein reiner Wegpunktgraph, der aus einer Teilmenge von Eckpunkten des Raumpolygons sowie den Ein- und Ausgängen besteht. Zur Berechnung wird zunächst an jedem konkaven Eckpunkt des Raumpolygons (d.h. der Innenwinkel ist größer als 180°) ein Wegpunkt gesetzt. Zusätzlich werden alle Ein- und Ausgänge des Raumes als Wegpunkte auf der entsprechenden Position des Raumpolygons gesetzt. Anschließend wird für jedes mögliche Paar von Wegpunkten überprüft, ob die Diagonale zwischen den beiden Wegpunkten das Raumpolygon schneidet. Ist dies nicht der Fall, so wird eine entsprechende Kante im Wegpunktgraph eingefügt. Der Corner Graph kann in $O(n)$ berechnet werden, wobei n die Anzahl der konkaven Ecken des Polygons ist.

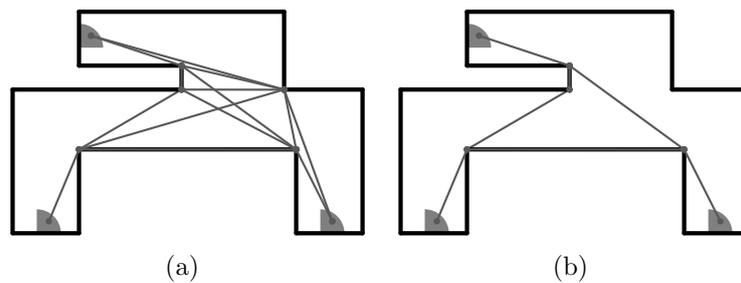


Abbildung 3.11.: Corner Graph eines konkaven Raumes mit drei Ein- und Ausgängen. (a) voll vermaschter Corner Graph. (b) vereinfachter Corner Graph, der nur die kürzesten Wege zwischen allen Paaren von Ein- und Ausgängen beinhaltet.

Abbildung 3.11 zeigt einen konkaven Raum mit drei Ein- und Ausgängen. Der resultierende Corner Graph hat den Vorteil, dass er eine sehr geringe Anzahl von Wegpunkten aufweist und die berechnete Pfadlänge immer minimal ist. Jedoch hat er gleichzeitig auch den Nachteil, dass Wege oftmals direkt an einer Wand entlang oder um eine spitze Ecke herum führen und dadurch weniger die natürliche Bewegung eines Menschen durch ein Gebäude widerspiegeln.

Da der fertige Corner Graph keine weiteren Informationen über die Lage von Wänden oder die Breite von Türen beinhaltet, ist es für einen berechneten Weg nicht möglich, die maximale Durchgangsbreite zu bestimmen. Des Weiteren kann es bei einer Routenberechnung vorkommen, dass ausgehend vom Startpunkt zunächst ein weiter entfernter Startknoten v_s gewählt wird bzw. dass ein längerer Pfad als der tatsächlich kürzeste berechnet wird. Dies lässt sich jedoch beheben, in dem ein bereits berechneter Weg $(v_s, \dots, v_z), v \in V$ nochmals geglättet wird, d.h. dass ein Wegpunkt $v_i, s < i < z$ aus dem Ergebnis entfernt werden kann, sofern zwischen v_{i-1} und v_{i+1} eine direkte Sichtverbindung besteht.

3. Bereitstellung von Umgebungsmodellen für I-LBS

Der Corner Graph lässt sich besser an die Bewegung eines Menschen anpassen, in dem man beispielsweise die Wegpunkte etwas von den Wänden in Richtung des Rauminnen versetzt.

Mittiger Wegpunktgraph: Für viele Anwendungen ist ein Wegpunktgraph, dessen Wegpunkte in der Mitte von Räumen und Korridoren liegt, von Vorteil. Dies reflektiert zum einen die Bewegung eines Menschen durch das Gebäude, zum anderen können die Wegpunkte besser verwendet werden, wenn dem Benutzer beispielsweise Navigationsanweisungen in Form von Pfeilen oder Linien auf einer Karte angezeigt werden sollen.

Ein einheitlicher und homogener mittiger Wegpunktgraph mit möglichst wenigen Wegpunkten ist in der Praxis schwer durch vollständige Automatisierung realisierbar und erfordert häufig einen großen manuellen Zusatzaufwand bei der Erstellung. In Kombination mit den nachfolgend vorgestellten Ansätzen lassen sich jedoch vergleichsweise gute mittige Wegpunktgraphen erzeugen.

Mediale Achse Die Medialachse eines Polygons besteht aus allen Punkten im Inneren des Polygons, bei denen der Abstand zu mindestens zwei Punkten auf dem Rand des Polygons minimal wird. Auch als *medial axis*, *symmetric axis* oder *skeleton* bezeichnet, kann sie für ein einfaches Polygon in $O(n)$ berechnet werden [31]. In einem konvexen Polygon besteht die Mediale Achse aus geraden Linien, bei konkaven Polygonen jedoch auch aus gekrümmten Linien. Dieser Umstand erschwert den Einsatz für Wegenetze, da die Abbildung in einem Wegpunktgraphen nur schwer zu realisieren ist.

Straight Skeleton Im Gegensatz zur Medialen Achse besteht das von Aichholzer et al. [4] vorgestellte *Straight Skeleton* aus geraden Linien. Dabei werden – ähnlich wie bei einem Buschfeuer – die Kanten des Polygons parallel in das Polygoninnere verschoben. Schneiden sich zwei Kanten, so entsteht ein Punkt des Skeletts. Abbildung 3.12 zeigt ein Beispiel. Felkel und Obdrzalek [44] beschreiben einen Algorithmus zur Berechnung des Straight Skeleton in $O(nm + n * \log(n))$, wobei n die Gesamtanzahl der Polygoneckpunkte und m die Anzahl der konkaven Ecken ist. Der beschriebene Algorithmus unterstützt auch Polygone mit Löchern, wie in Abbildung 3.13 gezeigt.

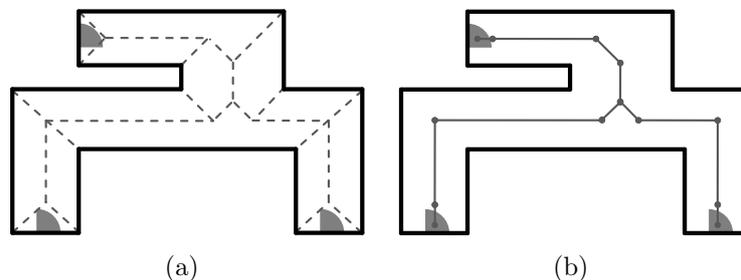


Abbildung 3.12.: Straight Skeleton eines Raumes mit drei Ein- und Ausgängen. (a) Straight Skeleton (b) resultierender Wegpunktgraph.

3. Bereitstellung von Umgebungsmodellen für I-LBS

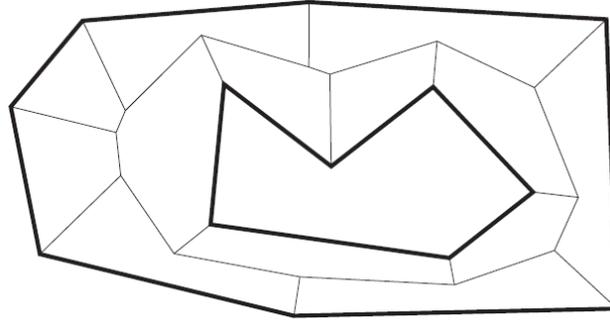


Abbildung 3.13.: Straight Skeleton eines Polygons mit Loch. Quelle: [44]

Vorteile des Straight Skeleton sind eine vergleichsweise geringe Anzahl an Kanten sowie ein geringer Verzweigungsgrad. Demgegenüber steht jedoch, dass viele Wege über die Mitte eines Polygons führen, was nicht immer den natürlichen Weg widerspiegelt.

Konvexe Partitionierung Die Konvexe Partitionierung unterteilt ein konkaves Polygon \mathcal{P} in eine Menge von x konvexen Teilpolygone $\mathcal{P}'_1, \mathcal{P}'_2, \dots, \mathcal{P}'_x$, wobei gilt: $\mathcal{P} = \mathcal{P}'_1 \cup \mathcal{P}'_2 \cup \dots \cup \mathcal{P}'_x$ und $\mathcal{P}'_1 \cap \mathcal{P}'_2 \cap \dots \cap \mathcal{P}'_x = \emptyset$. Dadurch kann jedes Teilpolygon eine beliebige Anzahl von Eckpunkten besitzen und ist nicht auf dreieckige Formen beschränkt. Ein Polygon $\mathcal{P} = \{p_1, p_2, \dots, p_n\}$ mit i konkaven Ecken lässt sich beispielsweise folgendermaßen partitionieren: man betrachtet nacheinander alle Ecken des Polygons, bis man eine konkave Ecke p_k findet, d.h. der Innenwinkel α zwischen den Ecken p_{k-1}, p_k und p_{k+1} ist größer als 180° . Die in Richtung des Polygoninneren verlängerte Winkelhalbierende von α schneidet \mathcal{P} im Punkt s . Nun unterteilt man \mathcal{P} in zwei Teilpolygone \mathcal{P}_1 und \mathcal{P}_2 . \mathcal{P}_1 besteht aus dem Schnittpunkt s und allen Punkten von \mathcal{P} , die gegen den Uhrzeigersinn auf s folgen bis einschließlich p_k . Das andere Teilpolygon \mathcal{P}_2 besteht aus p_k , den auf p_k im Uhrzeigersinn folgenden Punkten von \mathcal{P} bis einschließlich s . Beide Teilpolygone enthalten nun in der Summe weniger als i konkave Ecken, da der Innenwinkel der Ecke p_k durch die Halbierung kleiner als 180° geworden ist. Durch rekursive Teilung von \mathcal{P}_1 und \mathcal{P}_2 entstehen nach dem Theorem von Chazelle [29] letztendlich maximal $i + 1$ konvexe Teilpolygone. Abbildung 3.14 zeigt ein Beispiel.

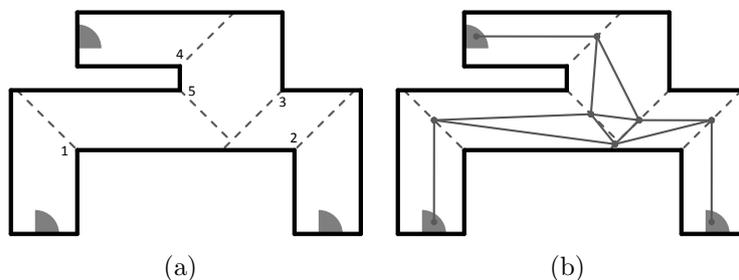


Abbildung 3.14.: Konvexe Partitionierung.

3. Bereitstellung von Umgebungsmodellen für I-LBS

Auch bei der Konvexen Partitionierung können Wegpunkte auf den Schwerpunkt jedes Teilpolygons gesetzt werden. Alternativ können ebenfalls Wegpunkte auf die Mittelpunkte der mit benachbarten Polygonen gemeinsamen Seiten positioniert werden. Vorteile der Konvexen Partitionierung sind eine geringe Knotenanzahl und ein geringer Verzweigungsgrad. Auch ist eine gute Begehbarkeit gewährleistet, und der Ansatz bietet eine optimale Abdeckung, jedoch nur bei Räumen die frei von Rundungen sind. Ein Nachteil ist jedoch, dass die Zerlegung in eine minimale Anzahl konvexer Teilpolygone sehr zeitaufwändig ist. Keil [72] beschreibt einen Algorithmus mit einer Komplexität von $O(n^3 \log(n))$. Hertel und Mehlhorn beschreiben einen Ansatz, der maximal viermal mehr Polygone als die optimale Zerlegung liefert und nur eine lineare Komplexität besitzt [114]. Eine ausführliche Diskussion findet sich auch in Tor und Middleditch [141].

Triangulierungen Eine weitere Möglichkeit zur Erzeugung eines Wegenetzes basiert auf der Triangulierung von Polygonen, insbesondere der sogenannten *Delaunay-Triangulierung*. Dabei werden die Polygone so in Dreiecke unterteilt, dass die Summe der Innenwinkel aller Dreiecke minimal ist. Dadurch ergibt sich insgesamt eine Unterteilung in besonders gleichmäßige Dreiecke und lange spitze Winkel werden vermieden. Basierend auf einer solchen Triangulierung können dann anhand der Dreiecke die Wegpunkte gesetzt werden. Im Folgenden wird zunächst eine formale Definition für die Delaunay-Triangulierung gegeben. Die verwendeten Bezeichner und Definitionen stammen dabei aus De Berg et al. [33]. Anschließend wird diskutiert, wie sich die allgemeine Delaunay-Triangulierung durch Hinzunahme von Stützpunkten auf konkave Raumpolygone anwenden lässt und wie die entsprechenden Wegpunkte gesetzt werden können.

Definition (Triangulierung eines Polygons). Sei \mathcal{P} ein einfaches Polygon. Eine *Triangulierung von \mathcal{P}* ist eine Unterteilung von \mathcal{P} in Dreiecke durch eine maximale Menge sich gegenseitig nicht überschneidender Diagonale.

Triangulierungen sind in der Regel nicht eindeutig, d.h. für ein gegebenes Polygon existieren zumeist viele verschiedene mögliche Triangulierungen. Jedes einfache Polygon lässt jedoch eine Triangulierung zu, und jede Triangulierung eines einfachen Polygons mit n Ecken besteht aus genau $n - 2$ Dreiecken. Betrachtet man allgemein eine Menge $P := \{p_1, p_2, \dots, p_n\}$ von Punkten in der Ebene, so lässt sich auch dafür eine Triangulierung definieren. Dazu sei \mathcal{S} die *maximale planare Unterteilung* definiert als eine Unterteilung in der keine Kante zwischen zwei Knoten hinzugefügt werden kann, ohne die Planarität von \mathcal{S} zu zerstören.

Definition (Triangulierung eine Menge von Punkten). Sei $P := \{p_1, p_2, \dots, p_n\}$ eine Menge von n Punkten in der Ebene. Eine Triangulierung \mathcal{T} von P ist definiert als eine maximale planare Unterteilung, deren Knotenmenge P ist.

Auch für eine Menge von Punkten existieren zumeist viele verschiedene Triangulierungen. Jede Triangulierung von P hat dieselbe Anzahl von Dreiecken. Die genaue Zahl hängt jedoch von der Anzahl k der Punkte ab, die auf der konvexen Hülle von P liegen. Jede

3. Bereitstellung von Umgebungsmodellen für I-LBS

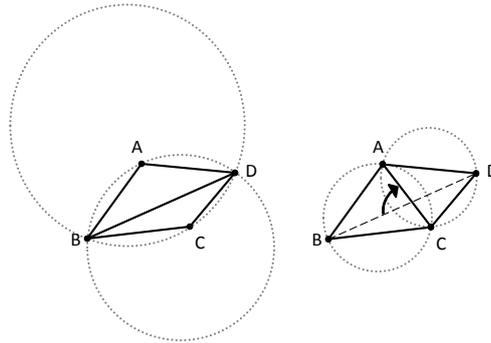


Abbildung 3.15.: Zwei verschiedene Triangulierungen der Punktemenge $P = \{A, B, C, D\}$. Rechte Seite: die Triangulierung bestehend aus den beiden Dreiecken $\triangle ABC$ und $\triangle ACD$ ist eine Delaunay-Triangulierung, da der Umkreis eines jeden Dreiecks keine weiteren Punkte enthält.

Triangulierung einer Menge P von n Punkten in der Ebene, die nicht alle kollinear sind, besteht aus $2n - 2 - k$ Dreiecken und $3n - 3 - k$ Kanten.

Zum Vergleich zweier Triangulierungen betrachtet man die Innenwinkel aller ihrer Dreiecke. Sei \mathcal{T} eine Triangulierung von P mit m Dreiecken. Sei $\alpha_1, \alpha_2, \dots, \alpha_{3m}$ die aufsteigend sortierte Liste der $3m$ Innenwinkel der Dreiecke in \mathcal{T} . Dabei wird $A(\mathcal{T}) := (\alpha_1, \alpha_2, \dots, \alpha_{3m})$ als der *Winkel-Vektor von \mathcal{T} bezeichnet*. Sei \mathcal{T}' eine andere Triangulierung von P und sei $A(\mathcal{T}') := (\alpha'_1, \alpha'_2, \dots, \alpha'_{3m})$ der zugehörige Winkel-Vektor. Der Winkel-Vektor von \mathcal{T} ist *größer* als der Winkel-Vektor von \mathcal{T}' , wenn $A(\mathcal{T})$ lexikographisch größer als $A(\mathcal{T}')$ ist, oder anders gesagt, wenn es einen Index i , $1 \leq i \leq 3m$ gibt, so dass gilt: $\alpha_j = \alpha'_j \forall j < i$ und $\alpha_i > \alpha'_i$. Dies wird mit $A(\mathcal{T}) > A(\mathcal{T}')$ bezeichnet. Eine Triangulierung wird *winkelo optimal* genannt, falls $A(\mathcal{T}) \geq A(\mathcal{T}')$ für alle Triangulierungen \mathcal{T}' von P ist.

Winkelo optimale Triangulierungen haben den Vorteil, dass längliche Dreiecke mit einem sehr spitzen Winkel eher vermieden werden und sich die Triangulierungen dadurch besonders gut zur Erzeugung von Wegenetzen eignen. Die Erzeugung einer winkelo optimalen Triangulierung kann auf verschiedene Arten geschehen. Dabei besteht unter anderem ein enger Zusammenhang mit Voronoi-Diagrammen, beispielsweise lässt sich der duale Graph eines Voronoi-Diagramms zur Erzeugung einer Delaunay-Triangulierung verwenden.

Definition (Delaunay-Triangulierung). Benannt nach dem russischen Mathematiker Boris Nikolaevich Delone bezeichnet die *Delaunay-Triangulierung* eine Triangulierung von P , so dass kein Punkt in P innerhalb des Umkreises eines Dreiecks der Triangulierung liegt.

Abbildung 3.15 zeigt ein Beispiel für zwei unterschiedliche Triangulierungen von vier Punkten A, B, C und D . Die linke Triangulierung besteht aus den beiden Dreiecken $\triangle ABD$ und $\triangle BCD$. Diese verletzt die Delaunay-Bedingung, da z.B. der Punkt A innerhalb des Umkreises des Dreiecks $\triangle BCD$ liegt. Durch einen sogenannten *flip* der Kante \overline{BD} entsteht eine Delaunay-Triangulierung (rechte Seite), bestehend aus den beiden Dreiecken $\triangle ABC$ und $\triangle ACD$.

3. Bereitstellung von Umgebungsmodellen für I-LBS

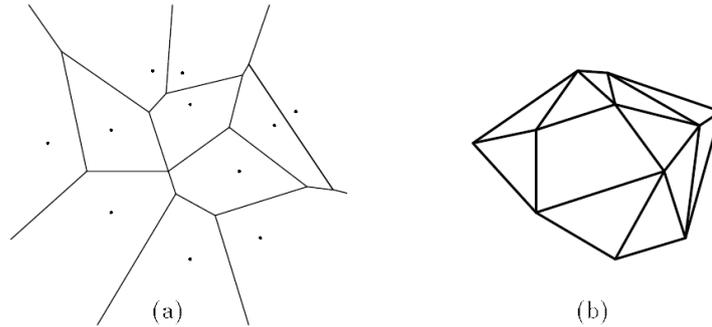


Abbildung 3.16.: (a) Ein Voronoi-Diagramm. (b) Die Triangulierung der Orte. Quelle: [49]

Definition (Voronoi-Diagramm). Sei $P := \{p_1, p_2, \dots, p_n\}$ eine Menge von n Punkten in der Ebene. Die Punkte in P werden als Orte (engl. sites) bezeichnet. Sei $\text{dist}(p, q) := \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$ die Euklidische Distanz zwischen zwei Punkten p und q . Das *Voronoi-Diagramm von P* ist definiert als die Unterteilung der Ebene in n Regionen (engl. cells), eine für jeden Ort in P , wobei gilt: ein Punkt q liegt in der Region eines zugehörigen Ortes p_i genau dann wenn $\text{dist}(q, p_i) < \text{dist}(q, p_j) \forall p_j \in P, j \neq i$. Das Voronoi-Diagramm von P wird durch $\text{Vor}(P)$ gekennzeichnet. Die Region eines Ortes p wird als *Voronoi-Region von p* bezeichnet und ist durch $\mathcal{V}(p)$ gekennzeichnet.

Das Voronoi-Diagramm kann z.B. mit Fortune's Algorithmus in $O(n \log n)$ berechnet werden. Der *duale Graph \mathcal{G}* von $\text{Vor}(P)$ besitzt einen Knoten für jede Voronoi-Region und zwischen zwei Knoten existiert eine Kante, wenn die entsprechenden beiden Regionen eine gemeinsame Kante besitzen. Sei \mathcal{G} der duale Graph von $\text{Vor}(P)$, wobei der Knoten einer Voronoi-Region $\mathcal{V}(p)$ der Punkt p ist. Die geradlinige Einbettung von \mathcal{G} , bei der alle benachbarten Voronoi-Regionen $\mathcal{V}(p)$ und $\mathcal{V}(q)$ durch die Strecke \overline{pq} verbunden sind, entspricht einer Delaunay-Triangulierung von P . Jedoch gilt diese Eigenschaft nicht im Allgemeinen: Nicht immer handelt es sich bei der resultierenden Triangulierung um eine echte Delaunay-Triangulierung. Abbildung 3.16 zeigt ein Beispiel für ein Voronoi-Diagramm, in dem vier Orte auf einem gemeinsamen Umkreis liegen. Die resultierende Triangulierung besteht dementsprechend nicht ausschließlich aus Simplexen und stellt somit keine Delaunay-Triangulierung dar. In diesem Fall kann eine zusätzliche Kante eingefügt werden, die das entsprechende Viereck wieder in zwei Dreiecke teilt, um wieder zu einer Triangulierung zu gelangen. In Fortune [49] findet sich eine ausführliche Diskussion über Voronoi-Diagramme und Delaunay-Triangulierungen.

Die Delaunay-Triangulierung eines Raumpolygons kann zur Erzeugung eines Wegenetzes verwendet werden, indem die Wegpunkte beispielsweise auf die Mittelpunkte der Dreiecke gesetzt werden und alle Paare von Wegpunkten, deren Dreiecke eine gemeinsame Kante besitzen, über eine Kante miteinander verbunden werden. Jedoch erzeugt die oben beschriebene Triangulierung im Falle von konkaven Polygonen auch Kanten, die außerhalb des Raumes liegen. In diesem Fall können entweder bestimmte Kanten als *constrained edges* deklariert werden, die in der Triangulierung enthalten sein müssen. Dann entsteht

3. Bereitstellung von Umgebungsmodellen für I-LBS

eine sogenannte *eingeschränkte (constrained) Delaunay-Triangulierung (CDT)*, bei der die Delaunay-Eigenschaft lokal verletzt werden kann, d.h. dass einzelne Dreiecke in ihrem Umkreis Punkte anderer Dreiecke enthalten. Die andere Möglichkeit ist, dass zusätzliche Stützpunkte (*Steiner Punkte*) an den Schnittpunkten von Kanten der Triangulierung und Kanten des Raumpolygons eingefügt werden. Dadurch werden letztendlich zusätzliche Dreiecke erzeugt, aber die Delaunay-Eigenschaft bleibt überall erhalten. In beiden Fällen kann die resultierende Triangulierung dann als Grundlage zur Erzeugung des Wegenetzes verwendet werden.

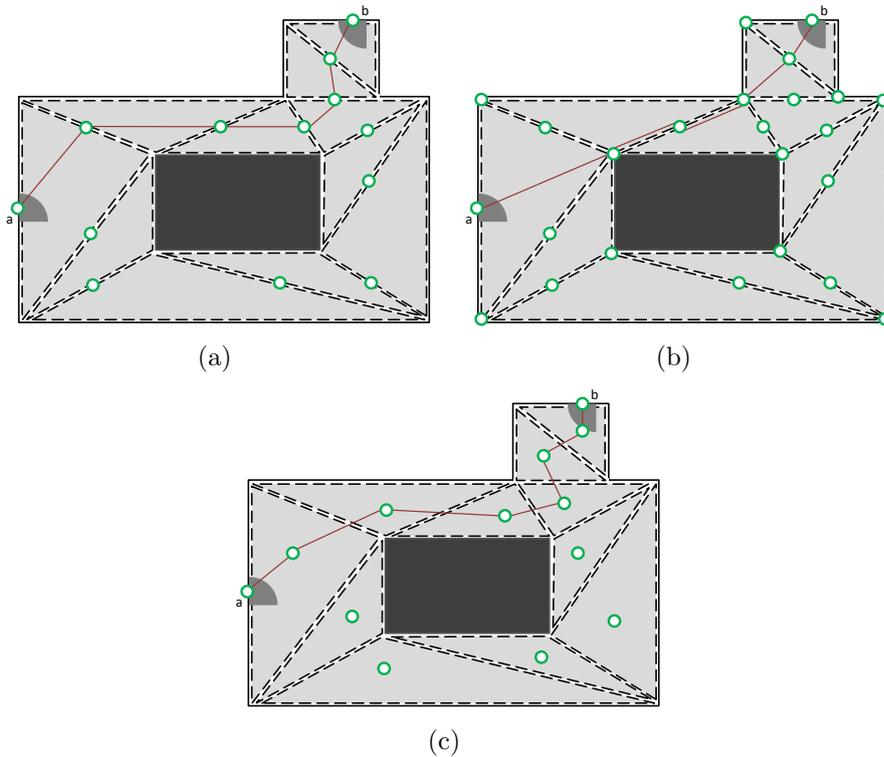


Abbildung 3.17.: Eine Triangulierung und drei verschiedene Möglichkeiten zur Wahl von Wegpunkten.

Die Wahl von Wegpunkten auf Triangulierungen ist ebenso wie bei n -eckigen Polygonen unterschiedlich möglich. Abbildung 3.17 zeigt eine Triangulierung (keine Delaunay-Triangulierung) mit drei verschiedenen Ansätzen. So können die Wegpunkte z.B. auf die Mittelpunkte von Seiten benachbarter Dreiecke (3.17(a)), zusätzlich auf die Eckpunkte benachbarter Dreiecke (3.17(b)) oder auch auf die Schwerpunkte der Dreiecke ((3.17(c)) gesetzt werden. Dies beeinflusst die Qualitäten des resultierenden Wegenetzes. So berechnet die zweite Variante einen sehr kurzen Weg, wohingegen die erste und dritte Variante einen etwas natürlichen Bewegungsverlauf widerspiegelt. Sind einzelne Dreiecke in einer Triangulierung sehr klein, so können sie unter Umständen mit einem benachbarten Drei-

3. Bereitstellung von Umgebungsmodellen für I-LBS

eck verschmolzen werden: zwei Dreiecke $\triangle ABC$ und $\triangle DEF$ können zusammengefasst werden, falls sie eine gemeinsame Kante besitzen und für $X, Y \in \{A, B, C\}$, $X \neq Y$ und $Z \in \{DEF\}$ gilt: die Gerade \overline{XY} schneidet Z . Oder anders ausgedrückt: falls die konvexe Hülle \mathcal{H} der sechs Punkte A, B, C, D, E, F ein Dreieck bildet und $\mathcal{H} \cap (\triangle ABC \cup \triangle DEF) = \mathcal{H}$ ist.

Triangulierungen bieten allgemein eine gute Möglichkeit zur Erzeugung sowohl von reinen Wegegraphen, als auch von Navigation Meshes, die weiter unten vorgestellt werden.

3.4.2. Flächenorientierte Algorithmen zur Erzeugung von Wegenetzen:

Die oben vorgestellten Ansätze betrachten bei der Erzeugung des Wegenetzes den Rand eines Polygons. Nachfolgend wird eine andere Klasse von Algorithmen vorgestellt: *Flächenorientierte Algorithmen* richten sich an der gesamten begehbaren Fläche eines Raumpolygons. Sie legen eine größere Anzahl von Punkten oder geometrischen Objekten auf das Polygon und versuchen anschließend, diese Objekte miteinander zu verbinden. So entstehen Wegenetze, die eine sehr große Anzahl von Wegpunkten enthalten können bzw. bei denen die Begehrbarkeit eines Raumes sehr fein aufgelöst wird. Gitterförmige Wegenetze oder Quadrees werden auch zum Pathfinding in Computerspielen eingesetzt (Tozour [142]), wenn beispielsweise ein *Non-Player-Character* automatisch durch eine 2D- oder 3D-Umgebung bewegt werden soll.

Gitterförmige Wegenetze: Ein gitterförmiges Wegenetz lässt sich vergleichsweise einfach erzeugen, indem man ein Gitter über das Raumpolygon legt und alle Gitterzellen entfernt, welche nicht vollständig innerhalb des Raumpolygons liegen. Besitzen zwei Gitterzellen eine gemeinsame Kante, so werden sie über eine Kante im Graphen miteinander verbunden. Dabei können die zugehörigen Wegpunkte des Graphen z.B. in die Mitte jeder Gitterzelle platziert werden. Je nachdem, welche Form die Gitterzellen haben, erhält man einen höheren bzw. niedrigeren Verzweigungsgrad. Bei einfachen quadratischen Zellen hat jede Zelle (mit Ausnahme der am Rand des Polygons liegenden Zellen) eine Verbindung zu vier anderen Zellen. Lässt man zusätzlich Verbindungen zu den diagonal über einen Eckpunkt erreichbaren Nachbarn zu, erhöht sich diese Zahl auf acht. Des Weiteren sind auch dreieckige oder hexagonale Zellen möglich, dabei besteht eine Verbindung zu bis zu sechs benachbarten Zellen. Ein so erzeugtes Wegenetz enthält je nach Größe des Gitters eine entsprechend hohe Anzahl an Knoten und Kanten. Abbildung 3.18 zeigt verschiedene Beispiele für gitterförmige Wegenetze. Je nach dem wie groß die Gitterzellen gewählt werden, kann es passieren, dass kein zusammenhängendes Wegenetz entsteht (Abbildung 3.18e) oder dass zwar ein zusammenhängendes Wegenetz entsteht, aber die Knotenzahl stark ansteigt (Abbildung 3.18f).

Vorteile von gitterförmigen Wegenetzen sind, dass sie vergleichsweise einfach zu implementieren sind und eine hohe Abdeckung erreicht werden kann, sowie dass für eine gegebene Koordinate die zugehörige Gitterzelle konstant in $O(1)$ bestimmt werden kann. Nachteile

3. Bereitstellung von Umgebungsmodellen für I-LBS

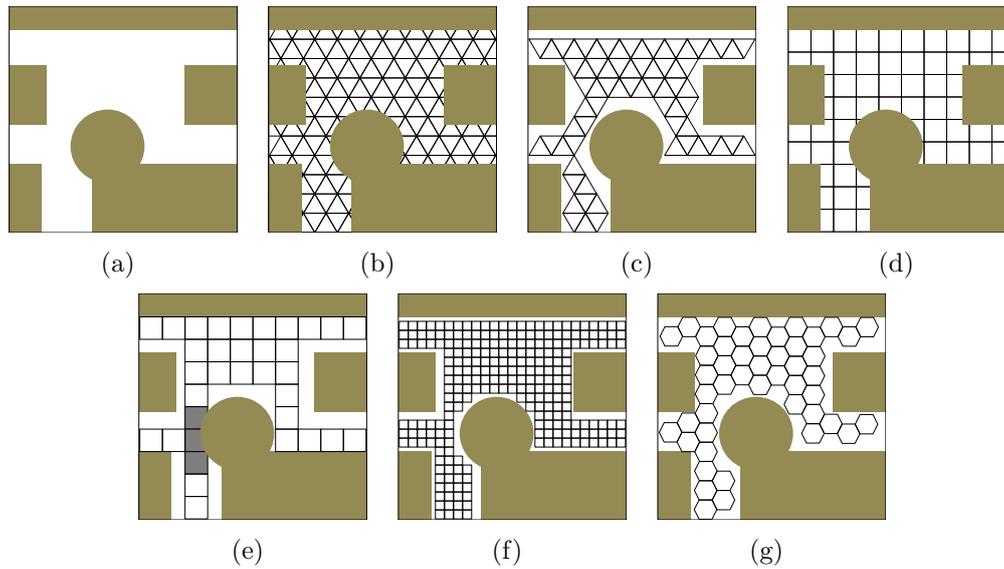


Abbildung 3.18.: Beispiele für gitterförmige Wegenetze: (a) ein Raum mit fünf Ein- und Ausgängen. (b) Dreieckbasiertes Gitter. (c) Dreieckiges Wegenetz. (d) Quadratbasiertes Gitter. (e) Großes quadratisches Wegenetz. (f) Kleines quadratisches Wegenetz. (g) Hexagonales Wegenetz.

sind jedoch die hohe Anzahl an Wegpunkten, der hohe Verzweigungsgrad und die Tatsache, dass ein nicht zusammenhängendes Wegenetz entstehen kann.

Quadrees Eine weitere Möglichkeit besteht darin, den von Finkel und Bentley [46] vorgeschlagenen Quadtree für die Erzeugung eines Wegenetzes zu nutzen. Dieses Verfahren unterteilt ein Polygon ebenfalls in Quadrate, allerdings kann die Größe der Quadrate variieren. Das minimal umgebende Quadrat eines Raumpolygons wird solange rekursiv in Zellen gevierteilt, bis jede Zelle vollständig innerhalb des Raumpolygons liegt oder bis eine minimale Zellengröße erreicht ist. Der Wegegraph über den Zellen entsteht hier beispielsweise, indem Knoten auf den Mittelpunkten gemeinsamer Kanten (oder auf einer gemeinsamen Ecke) gesetzt werden. Dadurch wird eine gute Abdeckung des Raumpolygons erreicht, welche auch bei Rundungen gute Ergebnisse produziert. Demgegenüber steht gerade in verwinkelten Räumen eine hohe Knotenanzahl und ein hoher Verzweigungsgrad.

Kreisbasierte Wegenetze Ein kreisbasiertes Wegenetz entsteht, indem Kreise gezeichnet werden, die vollständig innerhalb des Polygon liegen und anschließend die Mittelpunkte von sich gegenseitig schneidenden Kreisen miteinander verbunden werden. Abbildung 3.19 zeigt ein Beispiel.

In welchem Abstand die Kreise gezeichnet werden, hängt wiederum stark von der Umgebung ab. Ist der Abstand groß, so reduziert sich dementsprechend die Anzahl der Wegpunkte und in großen Räumen kommt es zu einer guten Abdeckung. Gleichzeitig kann dies

3. Bereitstellung von Umgebungsmodellen für I-LBS

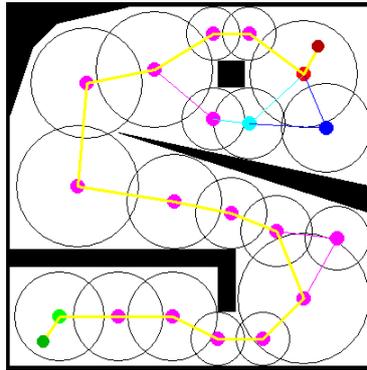


Abbildung 3.19.: Beispiel eines kreisbasierten Wegenetzes. Quelle: [144]

jedoch dazu führen, dass in schmalen Gängen nicht mehr alle Wege zusammenhängend abgebildet werden, da sich benachbarte Kreise nicht mehr schneiden. Wird andererseits ein kleiner Abstand gewählt, so mag dadurch zwar die Abdeckung in schmalen Gängen verbessert werden, aber in weitläufigen Bereichen erhöht sich sowohl die Knotenanzahl als auch der Verzweigungsgrad jedes einzelnen Knotens. Somit ist es für kreisbasierte Wegenetze im Allgemeinen schwer, einen optimalen Kompromiss zwischen der Anzahl der Knoten und der Abdeckung zu finden.

Bei allen flächenbasierten Ansätzen wird nur dann eine einhundertprozentige Abdeckung erreicht, wenn entweder das Gitter exakt das Polygon ausfüllt (z.B. beim Quadtree in einem quadratischen Polygon) oder wenn die Zellen des Gitters unendlich klein werden. Letzteres resultiert jedoch auch in einer unendlich großen Anzahl an Wegpunkten, und dies widerspricht der Forderung nach einer möglichst geringen Komplexität des Wegenetzes.

3.4.3. Navigation Meshes

Wenn man in einem Wegenetz nicht nur einzelne Wegpunkte und deren gegenseitige Erreichbarkeit speichert, sondern zusätzlich auch die geometrischen Formen von begehbaren Teilbereichen, so erhält man ein *Navigation Mesh*. Ein Navigation Mesh ist eine Menge von konvexen überlappungsfreien Polygonen, in der jedes Polygon mindestens eine gemeinsame Kante mit einem anderen Polygon hat. Die Vereinigung aller Polygone bildet im Idealfall die tatsächlich begehbare Fläche des betrachteten Bereiches ab. Der Vorteil eines Navigation Meshes ist, dass für unterschiedliche Arten von Anfragen der jeweils optimale Pfad bestimmt werden kann. Beispiele für mögliche Anfragen sind:

- Der kürzeste Weg zwischen zwei Orten A und B.
- Der kürzeste Weg zwischen A und B für ein Objekt bestimmter Größe.
- Der kürzeste Weg zwischen A und B für ein Objekt mit begrenztem Wendekreis.

3. Bereitstellung von Umgebungsmodellen für I-LBS

- Überprüfen, ob ein Ort B von einem Ort A aus sichtbar ist.

Ebenso wie bei reinen Wegpunktgraphen existieren auch für Navigation Meshes verschiedene Ansätze zur Erzeugung. Triangulierungen, insbesondere Delaunay-Triangulierungen, bieten sehr gute Eigenschaften für Navigation Meshes. Dazu werden nicht nur einzelne Wegpunkte auf den Dreiecken gesetzt (vgl. Abbildung 3.17), sondern gleichzeitig auch die Dreiecke selbst hinterlegt.

Eine Anwendung der eingeschränkten Delaunay-Triangulierung (CDT) zur Erzeugung eines Navigation Meshes zur schnellen Wegewahl wird z.B. von Demyen et. al. [37] beschrieben: ausgehend von einer Menge von Polygonen, die auch Löcher enthalten können, werden zunächst alle Linien die eine Wand repräsentieren, als *constrained edges* deklariert. Anschließend wird für die Menge aller Punkte eine CDT berechnet, wobei die *constrained edges* nicht entfernt oder verändert werden dürfen. Die Mittelpunkte von direkt aneinander grenzenden Dreiecken werden zu einem Graphen verbunden und die entsprechenden Knoten werden in vier verschiedene Klassen (Grad 0 bis 3) eingeteilt:

- Grad 0 Knoten sind alle Dreiecke, die aus drei *constrained edges* bestehen. Diese Dreiecke bilden jeweils eine Insel, und der Knoten ist mit keinem anderen Knoten im Graphen verbunden.
- Grad 1 Knoten: diese Knoten bilden einen Baum im Graphen und stellen Sackgassen im Wegenetz dar. Die Wurzel des Baumes verbindet diesen mit dem Rest des Graphen. Das Dreieck eines Knoten vom Grad 1 besteht aus zwei oder weniger *constrained edges* und enthält höchstens eine mit einem Grad 2 Knoten gemeinsame *unconstrained edge*.
- Grad 2 Knoten beschreiben Korridore in der Umgebung und sind über *unconstrained edges* mit zwei Knoten vom Grad 2 oder 3 verbunden. Eine Gruppe von zusammenhängenden Knoten vom Grad 2 bildet entweder einen Korridor zwischen zwei Knoten vom Grad 3 oder einen Rundweg, der an demselben Grad 3 Knoten startet und endet.
- Grad 3 Knoten entsprechen den Entscheidungspunkten bei der Wegewahl. Alle Knoten vom Grad 3 sind entweder direkt benachbart oder über Korridore miteinander verbunden.

Die Wegewahl in einem solchen Navigation Mesh erfolgt folgendermaßen: zunächst wird für den Start und Zielpunkt das zugehörige Dreieck ermittelt. Liegen Start und Ziel in unterschiedlichen Knoten, von denen mindestens einer vom Grad 0 ist, so existiert kein Weg. Liegen die Punkte beide im selben Baum, so findet eine Suche ausgehend vom Startknoten (Grad 1) über die Wurzel des Baumes bis zum Ziel statt. Ein Weg zwischen zwei Knoten vom Grad 2 liegt entweder vollständig innerhalb des selben Korridors, oder führt über einen oder mehrere Knoten vom Grad 3. Alle Knoten vom Grad 3 sind im Graphen direkt miteinander verbunden, sodass hier eine klassische Suche z.B. mittels A* durchgeführt werden kann. Ein solcher Ansatz lässt auch Anfragen mit vorgegebener Mindestbreite für

3. Bereitstellung von Umgebungsmodellen für I-LBS

Durchgänge zu. Dazu müssen in den Knoten vom Grad 3 lediglich Zusatzinformationen über die minimale Breite von Korridoren bzw. Rundwegen gespeichert und bei der Wegegwahl berücksichtigt werden.

Rand- und flächenorientierte Ansätze sowie Navigation Meshes sind unterschiedliche Möglichkeiten zur Organisation des Suchraumes für Wegenetze, und eine ausführliche Diskussion findet sich z.B. auch in Tozour [143], Obelleiro et al. [104], Hamm [54], Demyen und Buro [37] und Farnstrom [43]. Alle vorgestellten Ansätze zielen zunächst nur darauf ab, die Begehbarkeit *eines einzelnen Polygons* auszudrücken. Anders ausgedrückt: das resultierende Wegenetz gehört zu einem bestimmten Raum eines Gebäudes und steht zunächst nicht in Bezug zu anderen Wegenetzen, die möglicherweise für andere Räume des Gebäudes berechnet wurden.

Im folgenden Abschnitt wird nun diskutiert, wie das gesamte Modell eines Gebäudes erzeugt werden kann, d.h. auch, wie ein Gesamt-Wegenetz erzeugt werden kann, in dem alle Informationen über die Begehbarkeit des Gebäudes zusammengeführt sind.

3.5. Erzeugung von Umgebungsmodellen

In diesem Abschnitt sollen die für eine automatisierte Erzeugung von Umgebungsmodellen notwendigen Schritte näher betrachtet werden. Dazu zählen einerseits das Einlesen von Rohdaten, aus denen ein Modell-Erzeuger wichtige Basisinformationen wie beispielsweise die Lage und Form von Räumen sowie begehbare Bereiche identifizieren muss. Andererseits müssen diese Informationen zur Erstellung des fertigen Modells veredelt werden, d.h. es müssen zugehörige Wegenetze generiert werden und die Vollständigkeit des Modells muss überprüft werden. Eine ähnliche Vorgehensweise zum Erzeugen von Umgebungsmodellen wurde auch von Whiting et al. [151] vorgeschlagen. Dazu verwenden die Autoren ebenfalls digitale Gebäudepläne und erzeugen einen Wegpunktgraphen mittels einer eingeschränkten Delaunay-Triangulierung, um so Routen visualisieren zu können. Dort wird jedoch nicht weiter auf verschiedene Arten von Wegenetzen und deren Auswirkungen auf Dienste und Navigation eingegangen.

Im Folgenden wird ein allgemeiner Arbeitsablauf vorgestellt und diskutiert, der die generellen Schritte bei der automatisierten Erzeugung von Umgebungsmodellen beschreibt. Anschließend wird anhand von Beispielen gezeigt, wie sich die Auswahl von bestimmten Ansätzen auf die Eigenschaften eines fertigen Modells auswirkt.

Betrachtet man die zur automatisierten Erzeugung eines Umgebungsmodells notwendigen Schritte, so lassen sich fünf verschiedene Phasen identifizieren: das Einlesen von Rohdaten, die Bestimmung topologischer Zusammenhänge, die Generierung von Wegenetzen, die Überprüfung der Vollständigkeit und zuletzt die geeignete Speicherung des Modells:

1. **Einlesen der Rohdaten:** Die Pläne eines Gebäudes können in vielen verschiedenen Datei-Formaten vorliegen und dabei mehr oder weniger konsistent sein. Daher ist

3. Bereitstellung von Umgebungsmodellen für I-LBS

zunächst eine Aufbereitung notwendig, mittels derer aus den vorliegenden Daten mindestens folgende Informationen extrahiert werden:

- Lage und Form von Räumen
- Symbolische Bezeichner von Räumen
- Lage von Verbindungen zwischen Räumen (Türen, Durchgänge, Passagen)
- Zuordnung von Räumen und Verbindungen zu Stockwerken
- Zuordnung von Stockwerken zu Gebäuden
- Bestimmung von Stützpunkten für die Koordinatentransformation in andere Referenzsysteme

Eine Herausforderung beim Einlesen der Rohdaten ist oftmals, dass die zugrunde liegenden geometrischen Gebilde nicht immer einheitlich vorliegen. So kann beispielsweise die Form eines Raumes sowohl durch ein einzelnes Polygon (eine zusammenhängende Folge von Punkten), als auch durch eine nicht zusammenhängende Menge von einzelnen Linien gezeichnet worden sein. Ein anderes Problem stellt sich beim Vorliegen von rein rasterbasierten Bilddaten. Hier sind zusätzliche Algorithmen notwendig, die diese Informationen geeignet aufbereiten.

2. **Bestimmung der Gebäude-Topologie:** Sind die Lage und Form von Räumen bekannt, so müssen aus der Menge der gesammelten Informationen topologische Zusammenhänge abgeleitet werden, die später für die Berechnung von Wegen benötigt werden. Ein wichtiger Schritt ist hierbei auch das Matching zwischen verschiedenen Stockwerken. Da viele Rohdaten ein Gebäude als eine Menge unabhängiger Stockwerke darstellen, müssen in diesem Schritt z.B. Treppen, Aufzüge und Rolltreppen erkannt und miteinander in Beziehung gesetzt werden.
3. **Wegenetz-Generierung:** Wie im Abschnitt 3.4 beschrieben, existieren viele verschiedene Möglichkeiten zur Erzeugung von Wegenetzen. In diesem Schritt muss – eventuell auch in Abhängigkeit der späteren Anwendung – ein geeignetes Wegenetz erzeugt werden. Dazu zählt
 - a) die Ermittlung von Wegenetzen innerhalb einzelner Räume, sowie
 - b) die Zusammenführung der Wegenetze aller Räume, d.h. die Verknüpfung von Übergängen zwischen Räumen (z.B. Türen oder Treppen). Hierfür sind die Informationen aus Schritt 2 erforderlich.
4. **Überprüfung der Integrität:** Da die Rohdaten in der Regel nicht vollständig fehlerfrei sind, ist es notwendig, dass der Modell-Erzeuger entsprechende Tests durchführt, um das Modell auf Vollständigkeit und Korrektheit zu überprüfen. Ein einfaches Beispiel dafür ist, dass der Graph des erzeugten Wegenetzes zusammenhängend sein muss, damit alle Orte eines Gebäudes erreichbar sind.

3. Bereitstellung von Umgebungsmodellen für I-LBS

5. **Speicherung des Modells:** Am Ende der automatisierten Erzeugung steht die Speicherung des Modells in einem geeigneten Datenformat. Auf diesen Punkt wird im Kapitel 4 näher eingegangen.

Der dritte, vierte und fünfte Schritt kann bei Bedarf mehrmals und jeweils mit verschiedenen Parametern durchgeführt werden. Diese *Trennung zwischen der Hierarchie der einzelnen Räume (Schritt 2) und der Begehbarkeit innerhalb eines jeden Raumes (Schritt 3)* hat den großen Vorteil, dass sich dadurch für ein und dasselbe Gebäude verschiedene Wegenetze im Voraus berechnen lassen. So kann je nach Anforderung des ausführenden Dienstes das jeweils passende Wegenetz bereitgestellt werden:

- Beispielsweise kann auf diese Weise für ein Gebäude zum einen ein schlanker Corner Graph berechnet werden, anhand dessen sich effizient kürzeste Wege ermitteln lassen. Zum anderen kann parallel ein Wegenetz basierend auf einer Triangulierung angeboten werden, das z.B. in einer Navigationsanwendung verwendet werden kann.
- Alternativ könnten auch mehrere Wegenetze mit demselben Algorithmus, jedoch mit verschiedenen Parametern, erzeugt werden. Beispiele dafür sind die Delaunay-Triangulierung mit einer unterschiedlichen Anzahl an Steiner-Punkten oder gitterförmige Wegenetze mit einer unterschiedlich Größe der Zellen.

Das im Schritt 3 berechnete Wegenetz muss von einem Umgebungsmodell-Anbieter (vgl. Abbildung 2.1) später in einem geeigneten Suchraum organisiert werden, so dass eine effiziente Suche möglich wird. Hierfür bietet es sich an, die Räume und Stockwerke in einem hierarchischen Graphen zu organisieren. Dazu werden mehrere Graphen gebildet, die in verschiedenen Ebenen (*Layern*) angeordnet sind, wobei ein Knoten der Ebene n bestimmte Eigenschaften einer Menge von Knoten zusammenfasst, die sich auf einer darunter liegenden Ebene $n + 1$ befinden. Allgemein können hierarchische Graphen in den verschiedensten Bereichen zum Einsatz kommen: So beschreiben Fernandez und Gonzales [45], wie hierarchische Graphen für die Navigation mobiler Roboter eingesetzt werden kann. Bry et al. [23] geht auf den Aspekt des *Reasonings* für Geodaten ein und beschreibt dazu ebenfalls eine Hierarchie des Graphen. Von Stoffel et al. [138, 139] wird ein Ansatz beschrieben, der die Übergänge zwischen Räumen als Knoten in einem hierarchischen Graphen organisiert. Dabei kann die Hierarchie der Knoten entsprechend des Enthaltenseins von Räumen geschehen, oder auch, wie von Tsetsos [147] vorgestellt, indem jeweils ganze Stockwerke als Äste im hierarchischen Graphen stehen.

Da alle Algorithmen zur Erzeugung von Wegenetzen unterschiedliche Vor- und Nachteile haben, kann es vorteilhaft sein, wenn im Schritt 3a ein hybrider Ansatz verfolgt wird: anstatt alle Polygone mittels desselben Algorithmus zu berechnen, können einzelne Polygone zunächst klassifiziert und das jeweilige Wegenetz dann mittels eines geeigneten Algorithmus bestimmt werden.

3.5.1. Behandlung verschiedener Polygone

Um die Wahl eines Algorithmus für ein bestimmtes Polygon zu begünstigen, können die verschiedenen Polygone unterschiedlichen Klassen zugeordnet werden (siehe auch Abbildung 3.20). Das erste und wichtigste Kriterium ist die Unterteilung in *konvexe* und *konkave* Polygone. Erstere stellen die einfachste Form dar und bedürfen bei manchen der oben betrachteten Algorithmen keiner weiteren Bearbeitung. So entsteht in einem konvexen Polygon beispielsweise ein Corner Graph einfach durch die paarweise Verbindung zwischen allen Ein- und Ausgängen des Raumes. Auch bei der konvexen Partitionierung bedarf es keiner weiteren Unterteilung, das Polygon ist bereits konvex.

Im Vergleich zu geradlinigen Polygonen lassen sich Polygone *mit Rundungen* schwerer oder gar nicht zur Erzeugung eines Wegenetzes heranziehen. Eine Triangulierung schlägt bei Rundungen fehl, ebenso die konvexe Partitionierung, letztere allerdings nur bei konkaven Rundungen. Um dieses Problem zu umgehen, kann eine Rundung durch mehrere gerade Teilstrecken approximiert werden. Dadurch verringert sich bei einer zuvor konvexen Rundung die Fläche des Polygons, bei einer konkaven Rundung wird die resultierende Fläche größer.

Eine weitere Möglichkeit zur Klassifizierung ist die Unterscheidung zwischen *länglichen* bzw. *schmalen* und *großflächigen* bzw. *breiten* Polygonen. Sei A_{Poly} die Fläche eines Polygons und A_{MBR} die Fläche des minimal umgebenden Rechtecks des Polygons. Ferner seien MBR_x und MBR_y die Seitenlängen des minimal umgebenden Rechtecks des Polygons. Ein längliches Polygon liegt vor, wenn entweder das Verhältnis

$$\frac{A_{Poly}}{A_{MBR}}$$

oder das Verhältnis

$$\frac{\min(MBR_x, MBR_y)}{\max(MBR_x, MBR_y)}$$

klein wird, d.h. einen Wert nahe Null annimmt (beispielsweise 0,2).

Entscheidend für die Wahl eines Algorithmus kann auch sein, ob ein Polygon *Löcher* enthält. Die meisten betrachteten Ansätze lassen sich ohne Weiteres auch auf Polygone mit Löchern anwenden, wenngleich bei den gitterförmigen Wegenetzen die Auflösung des Gitters unter Umständen erhöht werden muss.

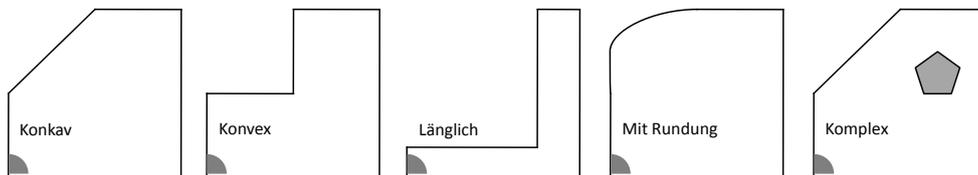


Abbildung 3.20.: Beispiele für unterschiedliche Klassen von Polygonen.

3.5.2. Empirischer Vergleich für die Erzeugung von Wegenetzen

Im Folgenden sollen die zuvor diskutierten Algorithmen zur Erzeugung von Wegenetzen anhand eines empirischen Vergleichs näher untersucht werden. Der Fokus liegt dabei auf der Klasse der randorientierten Algorithmen, da diese garantieren, dass alle Bereiche eines Raumes immer berücksichtigt werden. Die Ergebnisse wurden in [75] vorgestellt, eine ausführliche Behandlung findet sich in [73]. Als Datengrundlage dienen die CAD-Baupläne eines Universitätsgebäudes, bestehend aus fünf Stockwerken mit insgesamt 874 Räumen, 1015 Türen und 82 weiteren Verbindungselementen wie Treppen und Aufzüge. Aus DXF-Dateien werden die Polygone der Räume sowie die Positionen von Türen und Treppen ausgelesen. Diese Informationen bilden die Grundlage für die Erzeugung der verschiedenen Wegenetze. Abbildung 3.21 zeigt ein einzelnes Stockwerk des Gebäudes. Bei DXF handelt es sich um ein textbasiertes Format, das aus einer Abfolge von Schlüsseln und Werten aufgebaut ist. Die Schlüssel kennzeichnen den Typ des nachfolgenden Wertes, wobei ein Objekt wie z.B. ein Polygon durch eine Abfolge von mehreren Schlüssel-Wert-Paaren beschrieben wird. Beim Einlesen von DXF-Plänen müssen in der Regel mehrere verschiedene Ebenen verarbeitet und am Ende zusammengeführt werden. Für die Speicherung und weitere Verarbeitung des Modells wird ein XML-basiertes Format verwendet, das im nachfolgenden Kapitel im Abschnitt 4.3 ausführlich erläutert wird.

Die wichtigsten der nachfolgend untersuchten Eigenschaften sind:

- die Länge von auf dem jeweiligen Wegenetz berechneten Wegen
- die Komplexität (Anzahl der Knoten und Kanten) des resultierenden Graphen
- die Begehbarkeit des jeweiligen Wegenetzes

Unter den Begriff der Begehbarkeit fallen hier ein Mindestabstand zu Wänden, eine möglichst geringe Anzahl von Knoten auf einem Weg, wenige Richtungswechsel und die Vermeidung von Umwegen. Die Komplexität des Graphen beeinflusst den Speicherplatzbedarf und sollte daher möglichst gering sein. Die Länge der Wege sollte ebenfalls minimal sein und geht einher mit dem Kriterium der Vermeidung von Umwegen.



Abbildung 3.21.: Einzelnes Stockwerk des zur Simulation verwendeten Gebäudes.

3. Bereitstellung von Umgebungsmodellen für I-LBS

Algorithmus	Bemerkungen
Corner-Graph	Referenzalgorithmus, da immer kürzeste Weglänge
Waypoint-Graph	Variante des Corner-Graphs mit in das Rauminnere verschobenen Wegpunkten
Convex-Partition	Der ursprüngliche Ansatz der Konvexen Partitionierung
Convex-Partition-HM	Konvexe Partitionierung nach dem Algorithmus von Hertel und Mehlhorn
Straight-Skeleton	Das Straight-Skeleton
Triangulation	Delaunay-Triangulierung mit Wegpunkten auf dem Mittelpunkt der Dreieckskanten
Triangulation-C	Delaunay-Triangulierung mit Wegpunkten auf den Schwerpunkten der Dreiecke

Tabelle 3.2.: Übersicht der untersuchten Algorithmen für die Erzeugung von Wegenetzen.

Tabelle 3.2 zeigt die betrachteten Algorithmen. Der *Corner-Graph* dient dabei als Referenz für die Weglänge, denn er liefert immer den kürzesten möglichen Weg durch das Gebäude. Der *Waypoint-Graph* stellt eine Variante des *Corner-Graph* dar, bei dem die Eckpunkte in das Rauminnere verschoben sind. *Convex-Partition* bezeichnet die konvexe Partitionierung und *Convex-Partition-HM* die effiziente Implementierung nach Hertel und Mehlhorn [114]. Letztere entfernt aus einer Delaunay-Triangulierung sukzessive alle Dreieckskanten, die sich entnehmen lassen, ohne dass dadurch ein konkaves Polygon entsteht. Bei den Triangulierungen werden zwei verschiedene Varianten betrachtet: zum einen eine Delaunay-Triangulierung, bei der die Wegpunkte auf die Mittelpunkte der Dreieckskanten gesetzt werden (im Folgenden mit *Triangulation* bezeichnet), zum anderen eine Delaunay-Triangulierung, bei der die Wegpunkte jeweils auf die Schwerpunkte jedes Dreiecks gesetzt werden (*Triangulation-C*).

Als Beispiel für die verschiedenen Algorithmen ist der *Corner-Graph* eines einzelnen Stockwerks in Abbildung 3.22 dargestellt.

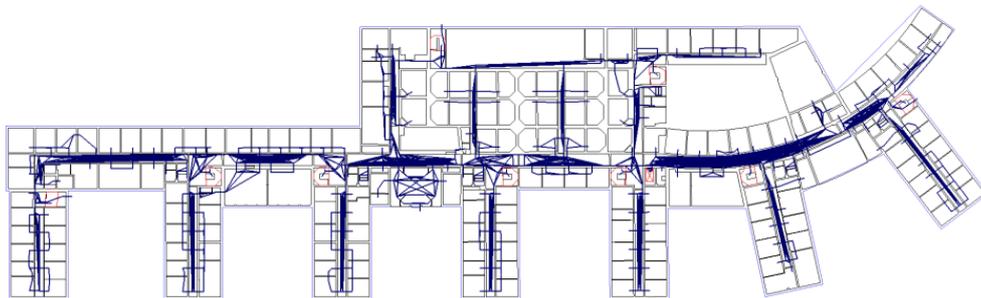


Abbildung 3.22.: Instanz des *CornerGraph* für ein einzelnes Stockwerk.

Für den Vergleich der Wegenetze werden jeweils 20 Routen zwischen 13 verschiedenen Orten berechnet, die manuell über das gesamte Gebäude verteilt liegen. Diese 13 Punkte

3. Bereitstellung von Umgebungsmodellen für I-LBS

befinden sich jeweils auf den Übergängen zwischen benachbarten Raumpolygonen, da die Position der restlichen Wegpunkte vom jeweils angewandten Algorithmus abhängt. Jeder Übergang wird durch zwei miteinander verbundene Wegpunkte gespeichert, so dass sich ein Wegpunkt innerhalb des einen Polygons befindet und der andere Wegpunkt innerhalb des benachbarten Polygons liegt. Des Weiteren werden für jeden Algorithmus gleichzeitig weitere Parameter untersucht:

- das Entfernen nicht benötigter Ecken (*Polishing*)
- die Genauigkeit der Gleitkommazahlen der einzelnen Koordinaten
- das Rendern von gekrümmten Liniensegmenten in mehrere einzelne Strecken

Unterschiedliche Genauigkeiten von Gleitkommazahlen stellen bei der Verarbeitung von Koordinaten ein generelles Problem dar. Ungenauigkeiten, Rundungsfehler und Überläufe erfordern, dass jede Koordinate bei der Verarbeitung mit einer gewissen Toleranz verarbeitet wird. Dadurch wird festgelegt, bis zu welchem Grad Koordinaten bei ähnlicher Nachkommastelle als gleich angesehen werden. Gekrümmte Ausbuchtungen in einem Raumpolygon (*bulges*) können durch das Einfügen von geradlinigen Verbindungen zwischen zusätzlichen Stützpunkten approximiert werden. Die Entfernung zwischen den eingefügten Punkten wird durch die so genannte *Render-Bulge-Distance* festgelegt. Eng benachbarte Eckpunkte können zusammengefasst werden. Dies reduziert die Komplexität der Polygone. Bei manchen Ansätzen, wie zum Beispiel bei der Erzeugung einer Delaunay-Triangulierung, sollten mehrere benachbarte Punkte nicht auf einer gemeinsamen Linie liegen, da der Umkreis von drei benachbarten Punkten dann unendlich groß werden kann und der Algorithmus nicht terminiert. Das so genannte *Polishing* entfernt entsprechende Punkte mit einem Winkel von 180° zwischen ein- und ausgehender Kante sowie kleine Ein- oder Ausbuchtungen im Polygon. Letzteres geschieht, wenn die entsprechende Linie kürzer als die so genannte *Skipping-Distance* ist. Abbildung 3.23 zeigt ein Beispiel für das *Polishing* und die Approximation einer gekrümmten Ausbuchtung durch gerade Teilstrecken. Die *Render-Bulge-Distance* muss immer größer sein als die *Skipping-Distance*, da ansonsten die beim Rendern einer gekrümmten Linie eingefügten Punkte durch das *Polishing* wieder entfernt werden würden.

Tabelle 3.3 zeigt fünf verschiedene Szenarien mit jeweils unterschiedlichen Werten für die Genauigkeit der Gleitkommazahlen, die Größe der *Skipping-Distance* und der *Render-Bulge-Distance*. Im ersten Szenario (*Standard*) wird eine *Skipping-Distance* von einem halben Meter angesetzt, um zu verhindern, dass fälschlicherweise Türen aus dem Modell entfernt werden. Zur Approximation gekrümmter Linien wurde ein Wert von 1m für die *Render-Bulge-Distance* gewählt. Dieser muss wie oben erwähnt größer als die *Skipping-Distance* sein, darf jedoch auch nicht zu groß sein, um das Modell nicht zu sehr zu verfälschen. Die Genauigkeit gibt an, ab welcher Auflösung zwei Koordinaten als identisch angesehen werden.

Für jeden der sieben Algorithmen werden gemäß der Szenarien in Tabelle 3.3 jeweils fünf Wegpunktgraphen erzeugt. Auf jedem Graphen werden wie oben erwähnt 20 Routen

3. Bereitstellung von Umgebungsmodellen für I-LBS

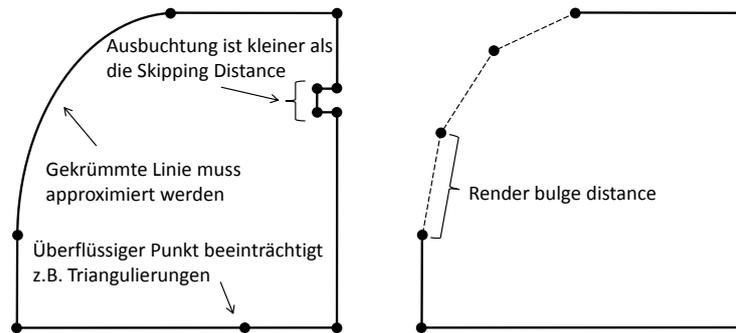


Abbildung 3.23.: Nachbearbeitung von Polygonen: Entfernung von kleinen Ausbuchtungen und Umwandlung von gekrümmten Linien in gerade Teilstrecken.

Szenario	Genauigkeit	Skipping-Distance	Render-Bulge-Distance
Standard	0,000001m	0,5m	1,0m
lowAccuracy	0,00001m	0,5m	1,0m
highAccuracy	0,0000001m	0,5m	1,0m
lowSkippingDist	0,000001m	0,25m	1,0m
lowRenderDist	0,000001m	0,5m	0,51m

Tabelle 3.3.: Parameter für unterschiedliche Szenarien.

zwischen 13 Orten berechnet. Die Ergebnisse sind in Abbildung 3.24 dargestellt. Es handelt sich dabei, wie bereits erwähnt, um einen Vergleich von Algorithmen, die auf ein bestimmtes Gebäude angewendet wurden. Die Zahlen stellen somit eine abschließende Bewertung für das konkrete Gebäude dar. Gleichzeitig fungieren sie auch als Anhaltspunkt für andere Gebäude und bieten eine Einschätzung für ähnliche Umgebungen. Nachfolgend werden die Eigenschaften der verschiedenen Ansätze diskutiert.

Unterschiede in den Szenarien Insgesamt lassen sich in dem betrachteten Modell aufgrund einer unterschiedlichen Genauigkeit der Koordinaten keine signifikanten Unterschiede beobachten. Die bei den beiden Triangulierungen marginale Erhöhung der Knotenanzahl im Szenario *lowAccuracy* lässt sich dadurch erklären, dass bei einer geringeren Genauigkeit einige wenige zusätzliche Steinerpunkte eingefügt werden und dadurch einige wenige zusätzliche Dreiecke entstehen.

Erwartungsgemäß erhöht sich die Anzahl der Knoten und Kanten bei einer Reduzierung der *Render-Bulge-Distance*. Besonders stark davon betroffen ist das *Straight-Skeleton* und der *Waypoint-Graph*. Die Zunahme bei der Anzahl der Kanten lässt sich bei beiden dadurch erklären, dass mehr Punkte entstehen, die untereinander sichtbar sind.

Verringert man die *Skipping-Distance*, so werden weniger Punkte beim *Polishing* entfernt, und die Gesamtanzahl der Knoten steigt mit Ausnahme des *Corner-Graph* leicht an. Allerdings erhöht sich beim *Corner-Graph* der Berechnungsaufwand für die Optimierung der Wege.

3. Bereitstellung von Umgebungsmodellen für I-LBS

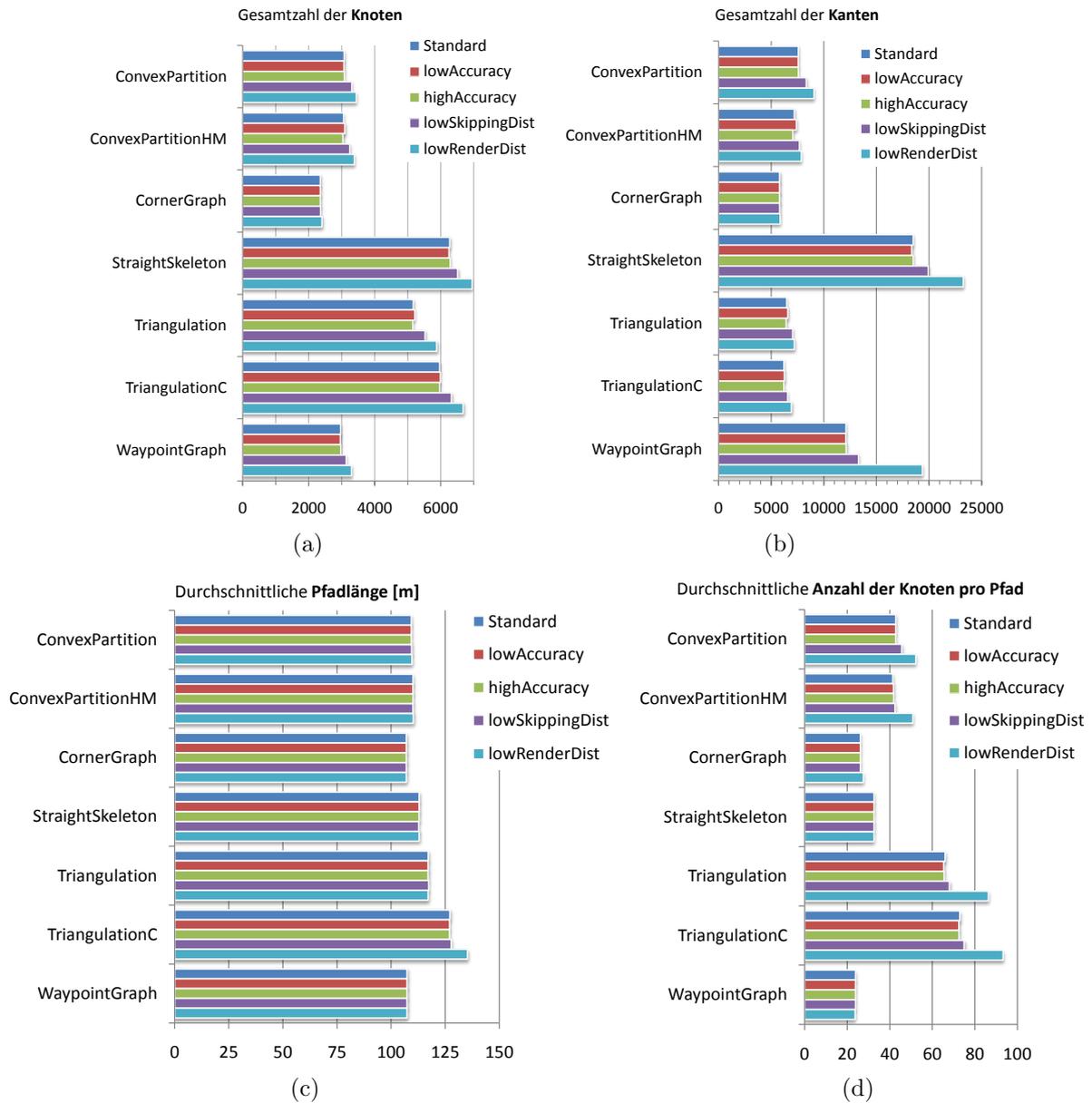


Abbildung 3.24.: Empirischer Vergleich von Algorithmen zur Erzeugung von Wegenetzen. (a) Anzahl der erzeugten Knoten. (b) Anzahl der erzeugten Kanten. (c) Durchschnittliche Länge der Pfade [m]. (d) Durchschnittliche Anzahl der Knoten pro Pfad.

Anzahl der Knoten und Kanten Bei der Erzeugung des *Corner-Graph* werden alle nicht benötigten Knoten und Kanten auf den Wegen zwischen Ein- und Ausgängen entfernt. Dementsprechend enthält der *Corner-Graph* immer eine minimale Anzahl von Knoten und Kanten (vgl. Abbildung 3.24(a) und 3.24(b)). Bei der Variante des *Corner-Graph* mit nach innen verschobenen Wegpunkten (*Waypoint-Graph*) erhöht sich die Anzahl der Kanten

3. Bereitstellung von Umgebungsmodellen für I-LBS

aufgrund der Tatsache, dass durch die Verschiebung der Punkte jetzt mehr Wegpunkte gegenseitig sichtbar sind und miteinander verbunden werden müssen. Ein Ausschnitt aus dem *Waypoint-Graph* ist in Abbildung 3.25(a) dargestellt.

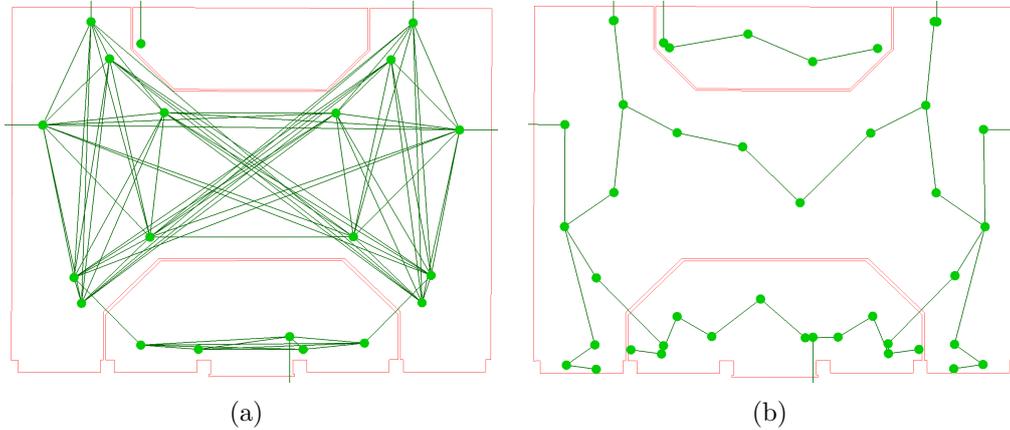


Abbildung 3.25.: Beispiele der Algorithmen: a) *Waypoint-Graph* b) *Triangulation-C*.

Die beiden Triangulierungen weisen zwar eine hohe Anzahl von Knoten auf, jedoch gleichzeitig eine geringe Anzahl an Kanten. Letzteres erklärt sich dadurch, dass bei einer Triangulierung immer nur Kanten zu unmittelbar benachbarten Dreiecken bestehen und nicht wie beim *Corner-* oder *Waypoint-Graph* eine vollständige Vermaschung innerhalb eines Raumes vorliegt. So weist *Triangulation* einen durchschnittlichen Verzweigungsgrad von 2,51 auf, für *Triangulation-C* liegt er bei 2,08. Der durchschnittliche Verzweigungsgrad beim *Corner-Graph* liegt hingegen bei 4,93, beim *Waypoint-Graph* beträgt er 8,17. Letzterer Wert erklärt sich durch die Vermaschung der Ein- und Ausgänge in lang gezogenen Korridoren, an denen viele einzelne Räume liegen. In solchen Situationen bieten die Triangulierungen wesentlich günstigere Eigenschaften. Abbildung 3.25(b) zeigt einen Ausschnitt der Triangulierung, bei der die Knoten auf die Schwerpunkte der Dreiecke gesetzt wurden (*Triangulation-C*).

Die Konvexen Partitionierungen erzeugen zwar mehr Knoten und Kanten als der *Corner-Graph*, dafür führen die Wege jedoch auch besser um Hindernisse herum und entsprechen einem natürlicheren Verlauf. Abbildung 3.26 zeigt ein Beispiel für *Convex-Partition* und *Convex-Partition-HM*. Das *Straight-Skeleton* weist die höchste Anzahl von Knoten und Kanten auf, sowie einen Verzweigungsgrad von 5,9. Die hohe Knotenanzahl resultiert vor allem daraus, dass die skelettbildenden Algorithmen gerade in Raumecken viele Wegpunkte erzeugen. Die große Kantenzahl entsteht hier, da die Übergangsknoten in diesem Versuch mit dem sichtbaren Knoten des Skeletts verbunden wurden.

Länge der Wege und Anzahl der Knoten pro Pfad Die in Abbildung 3.24(c) betrachtete durchschnittliche Pfadlänge ist erwartungsgemäß beim *Corner-Graph* minimal. Die ca. 5% geringere Anzahl der Knoten pro Pfad beim *WaypointGraph* resultiert daraus, dass bei

3. Bereitstellung von Umgebungsmodellen für I-LBS

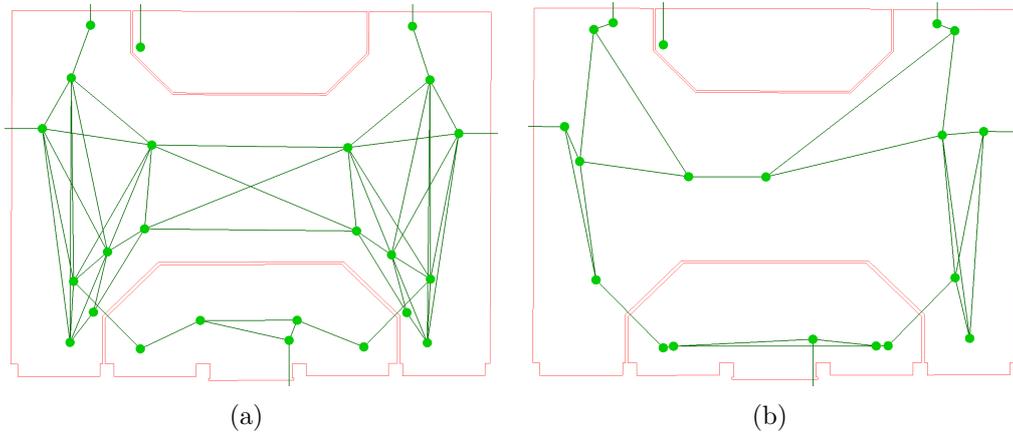


Abbildung 3.26.: Beispiele der Algorithmen: a) *Convex-Partition* b) *Convex-Partition-HM*.

längeren Wegen entlang einer konkaven Wand weniger Wegpunkte benötigt werden. Triangulierungen liefern gegenüber dem kürzest möglichen Weg einen um bis zu 16% längeren Pfad, bedingt durch die Zick-Zack-Form der Wege. Sie beinhalten gleichzeitig auch die höchste Anzahl an Knoten pro Pfad (vgl. Abbildung 3.24(d)).

Die restlichen Algorithmen liefern bei der Pfadlänge ähnliche Ergebnisse. Bei der Anzahl der Knoten pro Pfad zeigt sich eine Besonderheit des *Straight-Skeleton*: es werden vergleichsweise wenig Knoten besucht, obwohl das *Straight-Skeleton* aus vergleichsweise vielen Knoten besteht. Dies spricht dafür, dass bei einer zukünftigen Optimierung des *Straight-Skeleton* beispielsweise in den Ecken von Räumen auftretende Häufungen von Wegpunkten im Nachhinein bereinigt werden könnten. Die Konvexen Partitionierungen enthalten zwar mehr Knoten pro Pfad als *Corner-Graph*, *Waypoint-Graph* und *Straight-Skeleton*, jedoch spiegeln sie ebenso wie der *Waypoint-Graph* die natürliche Bewegung durch ein Gebäude relativ gut wieder.

Abdeckung und Begehbarkeit Alle vorgestellten Ansätze ermöglichen es, dass von einer beliebigen Startposition aus im gesamten Gebäude immer mindestens ein Knoten des Wegenetzes in Sichtweite ist, d.h. auf einem geraden Weg erreicht werden kann. Dadurch kann die Länge eines Weges von einer beliebigen Startposition zu einer beliebigen Zielposition korrekt bestimmt werden. Es kann jedoch passieren, dass innerhalb des betreffenden Polygons unter Umständen mehrere Wegpunkte liegen und dass der vom Startpunkt aus gesehen nächstgelegene Wegpunkt nicht unbedingt auf dem kürzesten Pfad zum Ziel liegt. Dann muss für jeden Knoten auf der Teilstrecke innerhalb des Startpolygons überprüft werden, ob durch eine direkte Verbindung des Startpunktes mit einem dieser Knoten ein noch kürzerer Weg erreicht werden kann.

Neben der Pfadlänge und der Anzahl der Knoten pro Pfad ist, wie bereits oben erwähnt, für viele I-LBS auch der Verlauf eines Weges entscheidend. Der *Waypoint-Graph* erweist sich bei näherer Betrachtung als vorteilhaft, da seine Wege nicht eng an Wänden entlang bzw. nicht direkt um eine Ecke herum verlaufen. Auch *Convex-Partition* liefert gute Er-

3. Bereitstellung von Umgebungsmodellen für I-LBS

gebnisse, da die Wege einen vergleichsweise direkten Verlauf zwischen Türen nehmen und tendenziell nicht direkt an Wänden entlang verlaufen. *Convex-Partition-HM* besteht aus der Teilmenge einer Triangulierung und ist den Triangulierungen nicht nur in Bezug auf die Anzahl der Knoten vorzuziehen: auch die resultierenden Wege sind kürzer und haben einen natürlicheren Verlauf. Das *Straight-Skeleton* ist aufgrund seiner hohen Komplexität eher weniger gut geeignet. Der *Corner-Graph* bietet sich für eine schnelle Berechnung kürzester Wege an, für andere Anwendungsszenarios wie beispielsweise die Erzeugung von Navigationsanweisungen ist jedoch der *Waypoint-Graph* oder *Convex-Partition* vorzuziehen.

Kombination der Algorithmen Bei dem in Abbildung 3.24 betrachteten Vergleich wurde jeweils ein bestimmter Ansatz auf ein ganzes Gebäude angewendet. Zusätzlich wurde auch eine Kombination der Algorithmen untersucht, bei der für jedes einzelne Polygon anhand seiner Form entschieden wird, welcher Ansatz verwendet wird. Das Ziel besteht dabei weniger darin, die Pfadlänge oder die Anzahl der Knoten pro Pfad zu reduzieren, da diese beim *Corner-Graph* bereits minimal sind. Vielmehr bietet eine solche Kombination die Möglichkeit, dass sich der Verlauf der Wege besser gestaltet, d.h. zum Beispiel besser für die Indoor-Navigation verwendet werden kann. Der *Waypoint-Graph* bietet sich für konkave längliche Räume sowie für Räume mit wenigen konkaven Ecken an, da er in diesen Fällen Wege mit vergleichsweise wenigen Knoten und einem Verlauf durch die Mitte der Räume ermöglicht. Sind ein großer Teil, z.B. mehr als ein Drittel, der Ecken in einem Polygon konkav, so erzielt die konvexe Partitionierung ein gutes Ergebnis. Wendet man diese beiden Ansätze in Kombination an, so bewegt sich die Anzahl von Knoten und Kanten knapp unterhalb des reinen *Waypoint-Graph*. Dies erklärt sich vor allem dadurch, dass beide Ansätze innerhalb eines konvexen Polygons die selben Knoten und Kanten erzeugen, nämlich die Vermaschung zwischen den Ein- und Ausgängen.

In dem vorgestellten Ansatz wurde die Erzeugung von Umgebungsmodellen beschrieben. Der Vergleich der Algorithmen demonstriert gleichzeitig auch die Umsetzbarkeit des Ansatzes. Wie genau und in welcher Form ein Umgebungsmodell-Anbieter diese Daten intern vorhält und weiterverarbeitet, ist jedoch dadurch noch nicht festgelegt. Ein einheitliches Format zur Speicherung des Modells, welches hier auch bereits für den Vergleich der Wegenetze eingesetzt wurde, wird im nächsten Kapitel vorgestellt.

Zuvor soll noch auf eine weitere Problematik eingegangen werden: die Bereitstellung von Umgebungsmodellen für mobile Anwendungen. Dies wird genau dann relevant, wenn nicht mehr sämtliche Anfragen an das Modell durch einen zentralen Server beantwortet werden, sondern auch anhand von Umgebungsmodellldaten, die direkt auf dem Endgerät vorliegen.

3.6. Umgebungsmodelle für mobile Anwendungen

Mit der Verbreitung von leistungsfähigen Endgeräten wie z.B. Smartphones und Tablet-Computern sind in den letzten Jahren auch zahlreiche neue Technologien für diese Klasse von Geräten entstanden. Zum einen existieren mittlerweile verschiedenste Sensoren z.B. zur

3. Bereitstellung von Umgebungsmodellen für I-LBS

Messung von Lage, Beschleunigung, Orientierung, Licht, Temperatur, oder zur Benutzer-Eingabe per Sprache, Geste und in Zukunft vielleicht auch Hirnströme. Zum anderen haben sich auf der Ebene der Software-Plattformen und -Frameworks Ansätze entwickelt, um ähnlich geartete Funktionalitäten vor dem Anwendungsentwickler zu verschatten und somit die Entwicklung zu vereinfachen. Ein Beispiel dafür sind die auf vielen Plattformen verbreiteten Methoden für den Zugriff auf Ortsinformationen. So wurde z.B. bereits 2003 mit dem *Java Specification Request (JSR) 179* [64] ein Standard verabschiedet, der für Entwickler in der *Java 2 Micro Edition* die Möglichkeit schuf, im Quellcode eines Programms auf Positionsdaten des jeweiligen Gerätes zuzugreifen, ohne das zu verwendende Positionierungssystem genauer zu spezifizieren. Die Aktivierung des Positionierungssystems (z.B. GPS oder Cell-ID) erfolgt zur Laufzeit anhand zuvor festgelegter Kriterien wie beispielsweise der benötigten Genauigkeit oder dem maximal zulässigen Energieverbrauch. Obwohl inzwischen von anderen Plattformen überholt, ist JSR 179 nach wie vor ein gutes Beispiel für die Idee des darunter liegenden Konzeptes. Eine zweite wichtige Entwicklung ist die lose Kopplung zwischen verschiedenen mobilen Anwendungen, wie sie beispielsweise auf der *Android-Plattform* mittels sogenannter *Intents* realisiert wird. Dieses Konzept ermöglicht es dem Entwickler, aus seiner Anwendung heraus Aktionen zu initiieren, deren zuständiges Programm vom Betriebssystem dynamisch gewählt wird. Ein Beispiel dafür ist der Befehl zum Öffnen und Anzeigen einer Webseite. Welches Programm genau beim Auftreten eines solchen *Intents* gestartet wird, hängt von der laufenden Konfiguration anderer Anwendungen ab. Ein dritter Aspekt ist, dass mit der steigenden Anzahl mobiler Anwendungen auch zunehmend auf einem einzelnen Endgerät mehrere Anwendungen parallel im Hintergrund laufen, die dabei immer wieder auf dieselben Basisinformationen zugreifen. Dazu zählen beispielsweise Kartendaten für die aktuelle Umgebung des Endgerätes, Statusinformationen über umliegende Orte und Objekte von Interesse oder auch die Registrierung von *Event-Handlern* für bestimmte ortsbezogene Ereignisse. Sofern diese Informationen nicht auf dem Endgerät selbst vorliegen, sondern von einem entfernten Server abgerufen werden, ist es sinnvoll, diese Anfragen und Daten nach Möglichkeit auf dem Endgerät zwischen zu speichern bzw. zu akkumulieren. Dadurch lässt sich zum einen die Anzahl der übertragenen Nachrichten reduzieren, zum anderen kann dadurch der Energieverbrauch des Endgerätes reduziert werden. Ein weiterer Aspekt ist, dass durch eine verminderte Anzahl von ortsbezogenen Anfragen bei einem Server die *Location Privacy* besser geschützt bleibt.

Auch bei Ausführung von ortsbezogenen Diensten in Gebäuden stellen diese Konzepte eine wichtige Grundlage dar. Die Informationen über ein Gebäude werden nicht von jedem I-LBS vollständig und zu jeder Zeit benötigt. Vielmehr können z.B. nur Informationen über die aktuelle unmittelbare Umgebung oder auch entlang einer bestimmten Route von Relevanz sein. Oder eine Anwendung benötigt zusätzlich Informationen über die ungefähre Distanz zu weiter entfernten Orten oder Landmarken, um dem Benutzer eine grobe Orientierung geben zu können. Angenommen, mehrere Anwendungen müssen gleichzeitig Distanzen oder Routen zu Orten in einem Gebäude berechnen, so bietet die Bearbeitung durch eine einzelne gemeinsame Komponente einige Vorteile gegenüber einem Ansatz, bei dem jede Anwendung für sich alle Berechnungen durchführt und die notwendige Datenbasis sowie Algorithmen selbst verwaltet.

3. Bereitstellung von Umgebungsmodellen für I-LBS

In diesem Abschnitt wird eine Klasse von I-LBS betrachtet, die unter Verwendung eines beliebigen Positionierungsverfahrens auf einem mobilen Endgerät ausgeführt werden und bei denen eine Verarbeitung und Darstellung von Ortsinformationen auf dem Endgerät erfolgt. Besonders wenn mehrere gleichzeitig ausgeführte Anwendungen und Dienste auf Umgebungsmodelldaten zugreifen möchten, bietet sich eine Bündelung dieser Anfragen an, um so gegenüber einem entfernten Umgebungsmodell-Anbieter für eine effiziente Kommunikation zu sorgen. In einem möglichen Szenario für einen mobilen I-LBS betritt ein Benutzer zum ersten Mal ein fremdes Gebäude und will sich durch sein mobiles Endgerät zu einem bestimmten Raum navigieren lassen. Dabei kann er zur Positionsbestimmung auf ein lokal verfügbares Positionierungssystem zurückgreifen, ferner besteht eine Internetverbindung über das Mobilfunkmodul des Endgerätes. Gleichzeitig möchte der Benutzer seine Position anderen Personen freigeben, jedoch nur, solange er sich innerhalb des besagten Raumes oder der unmittelbaren Umgebung befindet. In einem anderen Szenario befindet sich ein Benutzer an einem Flughafen und möchte auf seinem Endgerät eine Karte mit der Route zu seinem Gate sowie alle Restaurants entlang der Route angezeigt bekommen. Dabei sind für ihn ausschließlich die Informationen entlang der Route von Belang, alle weiteren Informationen über andere Terminals oder Gates sind nicht von Interesse. Ebenfalls möglich wäre eine mobile Anwendung, die einem Benutzer hilft, sich in einem Einkaufszentrum zurecht zu finden. Dabei könnte der Benutzer auf seinem Endgerät bereits Anwendungen von Drittanbietern installiert haben, die dem Benutzer innerhalb des Einkaufszentrums anhand seines Aufenthaltsortes Angebote und Informationen bereit stellen.

3.6.1. Bündelung von Anfragen

Aus der Sicht des Entwicklers einer mobilen Anwendung wird es in Zukunft von Vorteil sein, wenn Anfragen an ein Umgebungsmodell einfach und ohne spezielles Wissen über das jeweilige Gebäude gestellt werden können. Dafür sind entsprechende Schnittstellen notwendig, die einen einheitlichen Zugriff ermöglichen. Aus der Sicht eines mobilen Endgerätes ist es darüber hinaus erstrebenswert, dass die Anfragen aller auf dem Endgerät ausgeführten Anwendungen mit möglichst wenig Rechenaufwand sowie einem möglichst geringen Datenaufkommen auf der Luftschnittstelle beantwortet werden. Für manche Szenarien kann es zwar sinnvoll sein, sämtliche potenziell zur Ausführung benötigten Daten auf dem Endgerät zu speichern (wie beispielsweise viele Outdoor-Navigationssysteme Kartenmaterial über externe Speicherkarten nutzen), jedoch stellt der Indoor-Bereich im Allgemeinen gänzlich andere Anforderungen und eine vollständige Speicherung z.B. aller Gebäude einer Stadt ist in dieser Form (zumindest heutzutage) nicht möglich. Die Umgebungsmodelle für Gebäude ändern sich häufiger als das Straßennetz einer Stadt und noch wichtiger: der Zugriff auf Umgebungsmodelle ist zumeist nur speziellen Benutzergruppen gewährt, und längst nicht alle Umgebungsmodelle werden öffentlich verfügbar sein.

Übertragen auf die Nutzung von Umgebungsmodellen bedeutet dies, dass eine mobile Anwendung möglichst nur die Informationen anfordern sollte, die gerade benötigt werden. Einmal bei einem Server angefragt, sollten die Informationen nach Möglichkeit zwischengespeichert und weiter verwendet werden. Dabei stellt sich zunächst die Frage, in welcher

3. Bereitstellung von Umgebungsmodellen für I-LBS

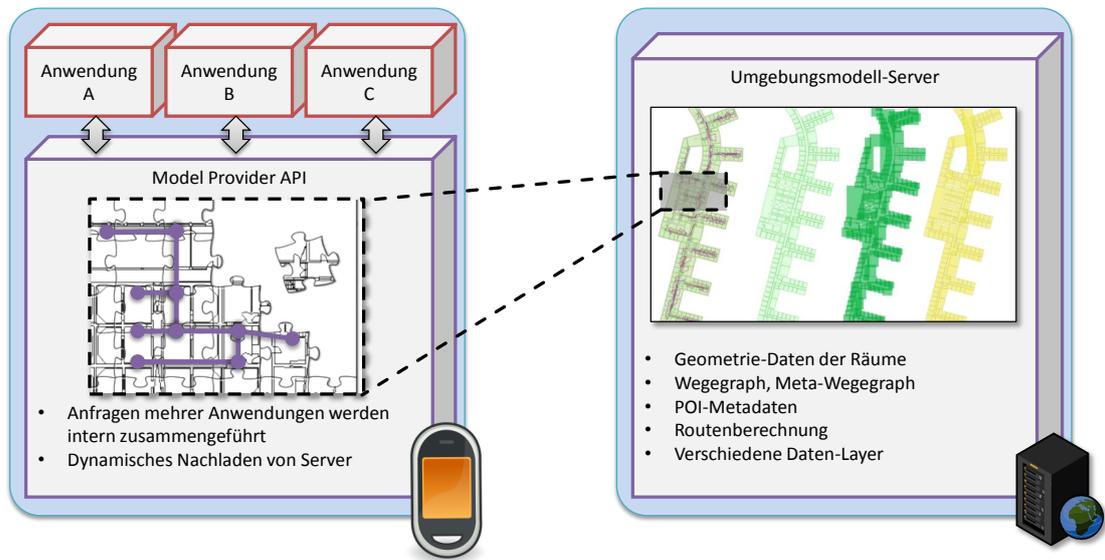


Abbildung 3.27.: Dynamisches Nachladen von Teilinformationen aus einem Umgebungsmodell auf ein mobiles Endgerät.

Form die umgebungsmodellspezifischen Daten auf einem mobilen Endgerät verwaltet und verwendet werden sollten und auf welche Art und Weise fehlende Informationen von einem entfernten Server angefragt werden sollten.

Entsprechend dem eingangs vorgestellten Rollenmodell betrifft das nachfolgend beschriebene Konzept eine Klasse von I-LBS, bei denen die Rolle des Model-Providers teilweise auf dem Zielobjekt ausgeführt wird. Teilweise deshalb, weil gleichzeitig auf einem entfernten Server ebenfalls Informationen vorgehalten werden und beide Komponenten an der Erbringung der Funktionen beteiligt sind.

Abbildung 3.27 skizziert die generelle Idee: ein zentraler Umgebungsmodell-Server hält sämtliche Informationen bereit, die die Geometrie des Gebäudes, Wegegraphen, Orte von Interesse und weitere Daten-Schichten betreffen. Gleichzeitig ist der Umgebungsmodell-Server in der Lage, Routen durch das Gebäude zu berechnen sowie bestimmte Informationen entlang von Routen zu übermitteln. Auf der Seite des mobilen Endgerätes steht den verschiedenen Anwendungen eine einheitlich Schnittstelle für den Zugriff auf das Umgebungsmodell zur Verfügung („Model Provider API“). Diese Schicht führt die Anfragen der verschiedenen Anwendungen zusammen und lädt fehlende Teilinformationen des Umgebungsmodells vom Umgebungsmodell-Server herunter. Wie im Abschnitt 3.2 vorgestellt, betreffen die Anfragen vor allem Orte von Interesse, die Berechnung von kürzesten Wegen und Navigationsanweisungen sowie Geometriedaten für die Visualisierung. Wenn ein Umgebungsmodell nicht vollständig auf dem mobilen Endgerät vorliegt, sondern Informationen von einem Server dynamisch nachgeladen werden müssen, so gibt es grundsätzlich drei verschiedene Möglichkeiten, diesen Informationsfluss zu optimieren:

1. In einer Antwort werden nur jene Daten übertragen, die für die momentane Situation

3. Bereitstellung von Umgebungsmodellen für I-LBS

gerade ausreichend sind.

2. Jede Anfrage wird genau und vollständig beantwortet. Für eine längere Route oder eine große Menge von angefragten POIs werden alle Daten übermittelt.
3. Zusätzlich zu den angefragten Informationen werden bereits weitere, potentiell relevante, Daten übermittelt.

Ein Beispiel für den ersten Fall ist, dass für eine längere angefragte Route durch ein Gebäude nur die jeweils nächsten 30 Meter übermittelt werden, erst nach Erreichen bestimmter Positionen werden weitere Daten nachgeliefert. Auch kann zunächst ein vereinfachter Graph übermittelt werden, der lediglich Übergänge zwischen Räumen (Portale) und deren Verbindungen enthält. In Bezug auf die geometrischen Daten kann das Modell auf unterschiedliche Weise aufgeteilt werden, so dass immer nur bestimmte Ausschnitte übertragen werden:

- Aufteilung in ein regelmäßiges Raster: Über das gesamte Gebäude bzw. über jedes Stockwerk wird ein Raster gelegt, und für eine bestimmte Anfrage werden nur jene Objekte zurück geliefert, die sich innerhalb der Zelle der angefragten Position befinden oder die diese Zelle schneiden. Dabei müssen die Zellen eine passende Größe aufweisen: sind sie zu klein, so werden unter Umständen nicht alle gewünschten Objekte geliefert, sind sie zu groß, so steigt der Overhead an nicht benötigten Informationen.
- Gezielte Anfrage einer Region: Beim Server wird ein bestimmter Ausschnitt des Modells angefordert, wobei die Größe des angefragten Ausschnittes vom Client vorgegeben wird.
- Aufteilung anhand der Struktur des Gebäudes: ähnlich zu dem rasterbasierten Ansatz wird das Gebäude in Zellen aufgeteilt, diese weisen jedoch keine regelmäßige Form auf, sondern entsprechen der Form der Räume.

Der zweite Fall (jede Anfrage wird vollständig beantwortet) stellt besonders dann eine gute Möglichkeit dar, wenn der Client die angefragten Daten vollständig und ausschließlich benötigt. In der Kombination mit mehreren hintereinander folgenden Anfragen wird dabei jedoch keine merkliche Verbesserung durch eine Zwischenspeicherung von Antworten erreicht: Abbildung 3.28 zeigt dazu die Ergebnisse einer Simulation, in der untersucht wird, wie sich die Speicherung von Informationen aus vorherigen Anfragen an einen Umgebungsmodell-Server auf die lokale Berechenbarkeit zukünftiger Anfragen auswirkt. Indem auf dem mobilen Endgerät die in der Vergangenheit von einem Umgebungsmodell-Server erhaltenen Ausschnitte und Routen gespeichert werden, sollen dadurch weitere Anfragen einer Anwendung möglichst lokal, d.h. ohne eine erneute Kommunikation mit dem Umgebungsmodell-Server beantwortet werden können. Es werden dazu drei verschiedene Anfragemuster verwendet (das Tupel (S_i, Z_i) bezeichnet dabei die i -te Routenanfrage von einem Startpunkt S_i zu einem Zielpunkt Z_i , und die Anzahl der insgesamt generierten Anfragen pro Anfragemuster ist n):

3. Bereitstellung von Umgebungsmodellen für I-LBS

- *Random*: Bei jeder Anfrage werden S_i und Z_i zufällig gewählt. Die Wahrscheinlichkeit der Punktauswahl ist gleichverteilt über die Fläche aller Räume.
- *Random Waypoint*: Dieses Muster entspricht einem Zielobjekt, das sich immer zu einem zufälligen nächsten Ort bewegt. Der Startpunkt einer Anfrage entspricht dem Zielpunkt aus der unmittelbar vorherigen Anfrage, d.h. $S_i := S_{i-1}$. Der Zielpunkt Z_i sowie der erste Startpunkt S_1 werden gleichverteilt zufällig auf der Fläche aller Räume gewählt.
- *Entrance*: Dieses Muster beschreibt einen wiederkehrenden Benutzer, dessen Routenanfrage immer vom Eingang eines Gebäudes startet, d.h. $S_1 = S_2 = \dots = S_n$. Z_i wird jeweils gleichverteilt zufällig auf der Fläche aller Räume gewählt.

Als Simulationsgrundlage dient das im Abschnitt 3.5 vorgestellte Gebäude. Für die Berechnung von kürzesten Wegen werden vier verschiedene Wegenetze betrachtet: ein CornerGraph, ein modifizierter CornerGraph mit nach innen verschobenen Wegpunkten, ein Wegpunktgraph basierend auf der konvexen Partitionierung sowie eine Delaunay-Triangulierung. Gemessen wird zum einen der Anteil der Knoten des Wegenetzes, die durch die akkumulierte Speicherung der Antworten vom Server auf dem Client bekannt sind. Aus den erhaltenen Routen kennt der Client so nach und nach einen immer größeren Teil des Wegenetzes. Nach ausreichend vielen Anfragen beim Server kann der Client mit diesem Wissen manche Routen lokal berechnen. So zeigt sich beispielsweise in Abbildung 3.28(a) für den CornerGraph, dass der Client bei 500 nach dem *Entrance-Muster* angefragten Routen in der Lage war, 22% dieser Anfragen lokal zu beantworten, dabei waren 62% der Knoten im Wegenetz bekannt. Dieses Beispiel und die Ergebnisse für die anderen Wegenetze zeigen insgesamt, dass zwar grundsätzlich eine Einsparung von Anfragen möglich ist, jedoch tritt sie erst nach einer vergleichsweise hohen Anzahl von Routenanfragen auf. Für die Praxis bedeutet dies, dass es in diesem Fall besser ist, den Graphen gleich zu Beginn vollständig auf den Client zu laden.

Bei der dritten Variante zur Beantwortung von Routenanfragen kann der Server zusätzliche, potentiell relevante Informationen mitschicken. Ein Beispiel dafür ist, dass zusätzlich zu einer berechneten Folge von Wegpunkten und zugehörigen Raumpolygonen auch weitere Daten entlang des Weges zurückgeliefert werden. Sollte der Benutzer vom geplanten Weg abweichen, so sind in diesem Fall bereits Daten zur Korrektur bzw. Unterstützung vorhanden. Dazu muss eine geeignete Auswahl der zusätzlichen Daten erfolgen. Betrachtet man die geometrischen Elemente eines Umgebungsmodells, so lassen sich hier Ansätze verwenden, die auch für browserbasierte Kartendienste eingesetzt werden:

- Für einen anzuzeigenden rechteckigen Kartenausschnitt mit der linken unteren Ecke (x_{lu}, y_{lu}) sowie rechten oberen Ecke (x_{ro}, y_{ro}) wird im Hintergrund bereits immer ein etwas größerer Ausschnitt $(x_{lu} - \delta x, y_{lu} - \delta y)$, $(x_{ro} + \delta x, y_{ro} + \delta y)$ geladen. Die Streifen mit der Breite δx bzw. δy bewirken beispielsweise, dass bei einem Verschieben der Karte durch einen Benutzer der neue Ausschnitt schneller angezeigt werden kann.

3. Bereitstellung von Umgebungsmodellen für I-LBS

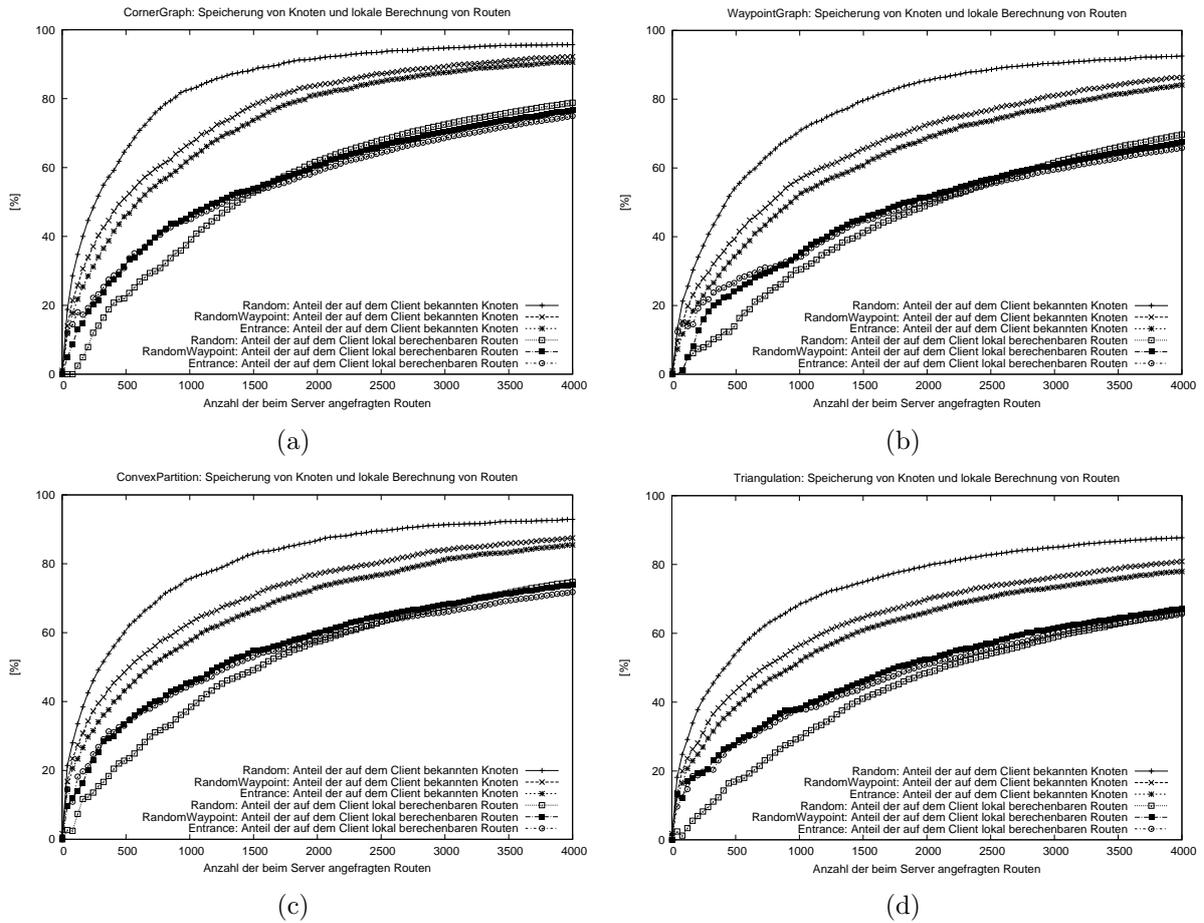


Abbildung 3.28.: Simulationsergebnisse zur Speicherung bereits bekannter Routen: (a) CornerGraph, 2323 Knoten. (b) Modifizierter CornerGraph mit nach innen verschobenen Wegpunkten (WaypointGraph), 2937 Knoten. (c) Konvexe Partitionierung, 3045 Knoten. (d) Delaunay-Triangulierung, 5138 Knoten.

- Die geometrischen Objekte können in unterschiedlichen Auflösungen gespeichert werden. Ein Raumpolygon wird dann zunächst z.B. durch einen minimal umgebenden Kreis oder eine minimal umgebendes Rechteck dargestellt. Erst in einer höheren Auflösung sind alle Eckpunkte eines Polygons enthalten.

Zusammengefasst ergibt sich für alle der drei diskutierten Möglichkeiten, dass der Umgebungsmodell-Server die Daten des Umgebungsmodells in Schichten unterteilt. Auf diese Weise können die jeweils benötigten Informationen durch den Client abgerufen werden:

- In Bezug auf das Wegenetz werden zwei unterschiedliche Auflösungen vom Server angeboten: Ein einfaches Wegenetz enthält lediglich die Verbindungen zwischen Räumen und ermöglicht dem Client eine Abschätzung der voraussichtlichen Route. In einer

3. Bereitstellung von Umgebungsmodellen für I-LBS

höheren Auflösung werden auch alle Wegpunkte innerhalb der Räume berücksichtigt, so dass eine genaue Wegewahl möglich wird.

- Für geometrische Objekte steht auf unterster Ebene eine grobe Approximation der Räume und Stockwerke durch minimal umgebende Rechtecke. Damit kann ein Client bei bekannter Aufenthaltskoordinate bereits seinen Standort eingrenzen. Für bestimmte Räume bzw. Polygone entlang eines Weges können dann die vollständigen geometrischen Objekte abgefragt werden. Orte von Interesse müssen dabei gesondert angefordert werden können, ebenso Bereiche von Interesse.

Auf der Seite des Clients können diese Daten an einer einzelnen Stelle verarbeitet und mehreren Anwendungen zur Verfügung gestellt werden. Für den Server ergibt sich dadurch, dass ein Umgebungsmodell auf eine Weise vorgehalten werden muss, die alle dieser Anfragen zulässt. Auf das Problem der Speicherung des Umgebungsmodells wird im nachfolgenden Kapitel genauer eingegangen.

3.7. Zusammenfassung und Diskussion

Eine wichtige Anforderung an Umgebungsmodelle für zukünftige Indoor LBS besteht, dass die Begehbarkeit der Umgebung durch das Umgebungsmodell erfasst werden kann. Dadurch können kürzeste Wege berechnet werden, und es kann die Generierung von Navigationsdaten unterstützt werden. Eine weitere Anforderung ist die Austauschbarkeit des Wegenetzes, dadurch können für unterschiedliche Indoor LBS auch unterschiedliche Wegenetze bereit gestellt werden. Zusammen mit den anderen genannten Anforderungen ergibt sich zunächst, dass ein entsprechendes Modell sowohl geometrische Koordinaten, als auch symbolische Bezeichner beinhalten muss. Betrachtet man die verwandten Arbeiten, so reicht eine rein hierarchische oder mengenbasierte Darstellung der Orte jedoch nicht aus, denn damit ist beispielsweise noch keine Wegewahl möglich. Die Eigenschaften der verschiedenen Indoor-Positionierungsverfahren müssen ebenfalls berücksichtigt werden. Der vorgestellte Ansatz zur Kombination von GPS- und GSM-Cell-Id-Positionierung zeigt, wie in einem Outdoor-Szenario Energie auf den mobilen Endgeräten eingespart werden kann. Durch einen dynamischen Wechsel zwischen Positionierungsverfahren wird so erreicht, dass das genauere Verfahren nur dann eingesetzt wird, wenn es tatsächlich zur Erkennung der räumlichen Beziehungen zwischen den Zielobjekten benötigt wird. Überträgt man dieses Prinzip auf Indoor-Umgebungen, so wird deutlich, dass auch innerhalb von Gebäuden die Positionierungsverfahren von einem Umgebungsmodell profitieren können. Ein Beispiel hierfür ist die Änderung einer Position in Bezug auf die gehbare Entfernung, die als zusätzlicher Filter zur Verbesserung eines Positionierungsverfahrens eingesetzt werden kann.

Die Begehbarkeit eines Gebäudes wird durch ein Wegenetz beschrieben, das verschiedene Formen annehmen kann. Ausgehend von der Lage und der Form der Räume, die über mehrere Stockwerke und (Teil-)Gebäude verteilt sind, kann die Begehbarkeit durch eine Menge untereinander verbundener Wegpunkte oder auch gegenseitig angrenzender Flächen ausgedrückt werden. Die Herausforderung besteht dabei in der Berechnung eines Wegenetzes für

3. Bereitstellung von Umgebungsmodellen für I-LBS

konkave Raumpolygone, da in einem konkaven Raum nicht einfach alle Ein- und Ausgänge über geradlinige Verbindungen erreichbar sind. Zur Ermittlung von kürzesten Wegen in konkaven Räumen bieten die diskutierten Ansätze unterschiedliche Vor- und Nachteile. Ein Corner Graph besteht zwar aus einer minimalen Anzahl an Knoten und Kanten, jedoch bietet er nicht unbedingt die beste Alternative für die Visualisierung eines Weges oder für die Erzeugung von Navigationsanweisungen. Andere Ansätze wie beispielsweise eine konvexe Partitionierung oder eine Triangulierung unterteilen den Raum in eine Menge von konvexen Polygonen bzw. Dreiecken. Dies hat einerseits den Vorteil, dass dadurch auch die begehbaren Flächen erhalten bleiben, andererseits steigt jedoch auch die Komplexität des Wegenetzes.

Anhand eines realen Gebäudes wurden die randorientierten Algorithmen einem empirischen Vergleich unterzogen. Das Einlesen von CAD-Bauplänen stellt dabei insofern eine Herausforderung dar, als das digitale Baupläne Details und gekrümmte Linien enthalten, die für die Erzeugung von Wegenetzen hinderlich sind. Daher müssen diese Daten zunächst in einem Zwischenschritt automatisiert überarbeitet werden, d.h. dass kleine Ausbuchtungen entfernt und gekrümmte Wände durch mehrere gerade Teilabschnitte approximiert werden. Bei der anschließenden automatisierten Erzeugung der Wegenetze zeigt sich, dass beispielsweise ein optimierter Corner Graph mit in das Polygoninnere verschobenen Eckpunkten gut geeignet ist. Auch eine konvexe Partitionierung liefert für bestimmte Arten von Polygonen gute Ergebnisse. Weniger Aussicht auf Erfolg bieten skelettbildende Algorithmen: sie erzeugen eine sehr große Anzahl an Knoten und Kanten, und sind ohne weitere Schritte zur Vereinfachung eher nicht geeignet.

Bei dem in Abschnitt 3.5 vorgestellten Konzept zur Erzeugung von Umgebungsmodellen wurde davon ausgegangen, dass beim Einlesen der Rohdaten bereits gewisse Informationen in den Bauplänen vorliegen. Im Falle von rein rasterbasierten Bilddaten sind wie bereits erwähnt zusätzliche Schritte notwendig, um die Lage und Form der Räume zu extrahieren. Eine solche Mustererkennung lässt sich gleichermaßen auch verwenden, um die Verbundenheit und Erreichbarkeit zwischen verschiedenen Räumen zu erkennen, indem beispielsweise Tür- oder Treppensymbole auf dem Bauplan erkannt und zugeordnet werden. Aufgrund von Fehlern bei der Erkennung kann jedoch auch eine mehr oder minder aufwändige manuelle Nachbearbeitung erforderlich sein. Ändert sich der Plan anschließend, so bedarf es eines zusätzlich geregelten Arbeitsablaufs, um den Mehraufwand bei Änderungen möglichst gering zu halten.

Für mobile Anwendungen ist eine Bündelung von Anfragen an ein Umgebungsmodell sinnvoll, wenn mehrere Anwendungen auf diese Daten zugreifen. Dazu können die Antworten eines Umgebungsmodell-Servers unterschiedlich umfangreiche Antworten enthalten, um so nur den aktuell benötigten Umfang an Informationen zu übermitteln.

In diesem Kapitel wurde vorrangig die Frage diskutiert, wie sich die für einen I-LBS benötigten Informationen über ein Gebäude möglichst *explizit* erfassen lassen. Eine implizite Darstellung der Raumtopologie wäre beispielsweise durch eine einfache Karte oder die Menge aller Räume und Durchgänge gegeben. Dass der Mensch relativ schnell aus diesen impliziten Informationen auf die Zusammenhänge in der Umgebung schließen kann, ist jedoch nur für einen Teil von Diensten ausreichend. In dem Moment, in dem ein Dienst

3. Bereitstellung von Umgebungsmodellen für I-LBS

selbst den kürzesten Weg bestimmen muss, werden explizite Zusammenhänge im Modell benötigt. Grundsätzlich besteht für einen Teil von Diensten auch die Möglichkeit, dass die Umgebungsinformationen zwar explizit, aber nicht innerhalb eines bestimmten Referenzsystems gespeichert werden. Dies ist gegeben, wenn die messbaren Parameter eines Positionierungssystems direkt als symbolische Koordinaten verwendbar sind und keiner weiteren Verarbeitung innerhalb eines geometrischen Koordinatensystems bedürfen. Darunter fallen z.B. eine direkte Abbildung von IDs umliegender Funkstationen, oder eine direkte Abbildung auf einen zuvor aufgezeichneten Verlauf von gemessenen Sensorwerten in einem inertialen Navigationssystem.

Nach der erfolgreichen Erzeugung eines Umgebungsmodells steht die Speicherung in einem geeigneten Format. Im nächsten Kapitel wird dazu ein neues Modell vorgestellt, das auf der Geography Markup Language aufbaut, und das einen einfachen Austausch des Umgebungsmodells zwischen den beteiligten Akteuren ermöglicht.

4. Ein GML-basiertes Umgebungsmodell für I-LBS

In diesem Kapitel wird ein neues Modell zur Speicherung von Umgebungsmodellen für ortsbezogene Dienste in Gebäuden vorgestellt. Die Beschreibung des Modells erfolgt in Form eines XML-Schemas, das auf der Geography Markup Language (GML) [107] basiert und als Anwendungsschema für GML 3.2 umgesetzt wird. Gemäß den Definitionen in Abschnitt 3.1 handelt es sich um ein hybrides Modell, welches die Eigenschaften von Orten sowohl anhand geometrischer Gebilde, als auch durch symbolische Bezeichner beschreibt. Dabei sind die symbolischen Bezeichner hierarchisch organisiert und die Erreichbarkeit zwischen Orten wird über einen Graphen ausgedrückt. Um wie bei mengenbasierten Modellen auch mehrere Orte bzw. Bereiche auf einer Hierarchieebene zusammenfassen zu können, wird zusätzlich das Konzept der Bereiche von Interesse eingeführt. Ziel des Modells ist es, die in Kapitel 3 diskutierten Informationen über Lage und Form von Räumen, Wegegraphen, Orte und Bereiche von Interesse etc. in einem einheitlichen Format so zu speichern, dass ein Akteur in der Rolle des Umgebungsmodell-Anbieters (vgl. Abbildung 2.1) in der Lage ist, alle von anderen Rollen entsprechend benötigten Informationen bereit zu stellen.

Im Folgenden wird zunächst auf die grundlegenden Konzepte von GML eingegangen, und es werden verwandte Arbeiten zur Speicherung von Umgebungsinformationen vorgestellt. Anschließend wird das neue Modell im Abschnitt 4.3 vorgestellt.

4.1. Geography Markup Language

Die *Geography Markup Language (GML)* [107] ist ein vom *Open Geospatial Consortium* entwickelter Standard für den Austausch und die Speicherung von geographischen Informationen. Genauer gesagt ist GML eine XML-Grammatik, die in XML Schema [153, 154] geschrieben wurde und eine Menge von vordefinierten Datentypen und Elementen zur Erstellung von so genannten Anwendungsschemas (application schemas) enthält. Der Verabschiedung des GML-Standards gingen viele andere Arbeiten und Standards voraus, insbesondere eine Reihe von Standards der *International Organisation for Standardization (ISO)*. So beschreibt beispielsweise ISO-19103 eine *Conceptual Schema Language* zur Modellierung von Maßeinheiten und Basistypen, ISO-19107 behandelt räumliche Schemas und ISO-19111 beschreibt Koordinatenreferenzsysteme. Zeitliche Eigenschaften von geographischen Informationen werden in ISO-19108 beschrieben.

GML (die auch als ISO-19136 veröffentlicht wurde) behandelt die Umsetzung dieser bestehenden Modelle und Konzepte in eine XML-Grammatik. Dies wird auch anhand des

4. Ein GML-basiertes Umgebungsmodell für I-LBS

Titels des GML-Standards deutlich, er lautet *Geography Markup Language Encoding Standard*. Das *Encoding* deutet an, dass GML die bestehenden Modelle, die in den anderen Standards der ISO 191-Familie beschrieben werden, in eine XML-Grammatik umsetzt. Insgesamt besteht GML in der aktuellen Version 3.2 aus 29 Teilschemas. Dabei ist GML selbst keine abgeschlossene Sprache, die eins zu eins in einer Anwendung verwendet werden kann. Die von GML vordefinierten Datentypen können von einem Entwickler innerhalb einer spezifischen Anwendungsdomäne zur Erstellung eines Anwendungsschemas verwendet werden. Ein grundlegender Begriff ist dabei das so genannte *feature*:

„A feature is an 'abstraction of real world phenomena.' (ISO 19101); it is a geographic feature if it is associated with a location relative to the Earth. So a digital representation of the real world may be thought of as a set of features. The state of a feature is defined by a set of properties, where each property may be thought of as a name, type, value triple.“ [107]

Dieser zunächst sehr allgemeine Oberbegriff ermöglicht es, dass sich unter einem *feature* quasi jedes beliebige Phänomen der realen Welt abbilden lässt. Der Bezug zu einem geographischen Ort wird hergestellt, indem einem *feature* als Eigenschaft (*property*) entsprechende Koordinaten zugeordnet werden. Dem *feature* kann eine beliebige Anzahl von *properties* zugeordnet werden, die wiederum bestimmte Werte (*values*) besitzen. Zur Definition des *property*-Begriffs verweist GML auf die *RDF Model and Syntax Specification* des W3C:

„A property is a specific aspect, characteristic, attribute, or relation used to describe a resource. Each property has a specific meaning, defines its permitted values, the types of resources it can describe, and its relationship with other properties.“ [152]

Der einem *property* zugeordnete Wert muss nicht unbedingt ein einfacher Datentyp sein, es kann sich dabei auch um komplexe Datentypen oder sogar andere *features* handeln:

„A property element may contain its value as content encoded inline, or reference its value with a simple XLink. The value of a property may be simple, or it may be a feature or other complex object. When recorded inline, the value of a simple property is recorded as a literal value with no embedded markup (text), while if the value is complex it appears as a subtree using XML markup (i.e. an XML element with sub-structure).“ [107]

Die Verlinkung von *values* hat den großen Vorteil, dass verschiedenste Objekte innerhalb eines Modells nur an einer Stelle deklariert werden müssen und dann über einen entsprechenden *XLinks* verknüpft werden können.

Zur Erstellung eines *application schemas* definiert der Entwickler zunächst eine Menge von benötigten *features* und fasst diese in einem *feature catalog* zusammen. Innerhalb eines bestimmten Anwendung-Schemas könnte ein *feature* beispielsweise ein Auto, eine Straße,

4. Ein GML-basiertes Umgebungsmodell für I-LBS

ein Gebäude, ein Baum oder auch eine Veranstaltung, ein Grundstück, eine Menge von Wetterstationen oder eine Wanderweg sein. Anschließend wird jedem *feature* eine Menge von *property*-Elementen zugeordnet, die die Eigenschaften des *features* näher beschreiben. Neben drei räumlichen Dimensionen können *features* auch zeitliche Eigenschaften zugeordnet werden, z.B. das Datum der Erzeugung, Dauer einer Veranstaltung oder Alter eines Baumes. Die Definition der *property*-Elemente erfolgt durch Zuordnung zu bestehenden *property types*. Der Entwickler des Anwendungs-Schemas kann dabei entweder auf die bereits im W3C XML Schema beschriebenen *simpleTypes* zurückgreifen oder komplexe Datentypen verwenden. Komplexe Datentypen können sowohl abstrakte als auch konkrete GML-Datentypen enthalten. Ein konkreter Datentyp ist beispielsweise *PointPropertyType* zur Beschreibung eines Punktes. Ein Beispiel für einen abstrakten Datentyp ist *AbstractGMLType*, das folgendermaßen definiert ist¹:

```
1 <element name="AbstractGML" type="gml:AbstractGMLType" abstract="true"
  substitutionGroup="gml:AbstractObject">
3   <annotation>
      <documentation>The abstract element gml:AbstractGML is any GML object
5       having identity. It acts as the head of an XML Schema substitution
      group, which may include any element which is a GML feature, or other
7       object, with identity. This is used as a variable in content models
      in GML core and application schemas. It is effectively an abstract
9       superclass for all GML objects.</documentation>
  </annotation>
11 </element>
  <complexType name="AbstractGMLType" abstract="true">
13   <sequence>
      <group ref="gml:StandardObjectProperties" />
15   </sequence>
      <attribute ref="gml:id" use="required" />
17 </complexType>
  <group name="StandardObjectProperties">
19   <sequence>
      <element ref="gml:metaDataProperty" minOccurs="0" maxOccurs="unbounded" />
21   <element ref="gml:description" minOccurs="0" />
      <element ref="gml:descriptionReference" minOccurs="0" />
23   <element ref="gml:identifier" minOccurs="0" />
      <element ref="gml:name" minOccurs="0" maxOccurs="unbounded" />
25   </sequence>
  </group>
```

GML bietet drei unterschiedliche Klassen von vordefinierten *property types* an:

- *Geometry Property Types* zur Beschreibung von Punkten, Linien, Polygonen, Oberflächen, Körpern sowie Sammlungen dieser Typen.

¹Nach einer Konvention im GML-Standard werden alle Objekte als XML-Elemente mit einem Namen in *UpperCamelCase* instantiiert. Die Namen von *property elements* werden in *lowerCamelCase* geschrieben und allen abstrakten Elementen wird das Präfix *Abstract* bzw. *abstract* vorangestellt. Komplexe Datentypen werden in *UpperCamelCase* und mit dem Suffix *Type* geschrieben.

4. Ein GML-basiertes Umgebungsmodell für I-LBS

- *Formal Topology Property Types* beinhalten Elemente zur Beschreibung von topologischen Beziehungen.
- *Formal Temporal Property Types* drücken zeitliche Eigenschaften und Beziehungen von Elementen aus.

Diese *property types* können vom Entwickler nach Belieben in dem jeweiligen *application schema* eingesetzt werden. Beispielsweise könnte ein sehr einfaches (sowie wohlgeformtes und gültiges) Anwendungs-Schema zur Darstellung von Parkanlagen folgendermaßen aussehen:

```
<?xml version="1.0" encoding="UTF-8"?>
2 <schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml
  /3.2" xmlns:park="http://www.example.org/park" targetNamespace="http://www.example.
  org/park" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <import namespace="http://www.opengis.net/gml/3.2" schemaLocation="http://schemas.
    opengis.net/gml/3.2.1/gml.xsd"/>
4 <element name="AbstractParkInformation" type="park:AbstractParkInformationType"
  abstract="true" substitutionGroup="gml:AbstractFeature"/>
  <complexType name="AbstractParkInformationType" abstract="true">
6 <complexContent>
  <extension base="gml:AbstractFeatureType"/>
8 </complexContent>
</complexType>
10 <element name="Park" type="park:ParkType" substitutionGroup="
  park:AbstractParkInformation"/>
  <complexType name="ParkType">
12 <complexContent>
  <extension base="park:AbstractParkInformationType">
14 <sequence>
  <element name="Flaeche" type="gml:SurfacePropertyType" minOccurs="1"
    maxOccurs="1"/>
16 <element name="Eingang" type="gml:PointPropertyType" minOccurs="1"
    maxOccurs="unbounded"/>
  </sequence>
18 </extension>
</complexContent>
</complexType>
20 <complexType name="ParkPropertyType">
  <sequence minOccurs="0">
22 <element ref="park:Park"/>
24 </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
26 </complexType>
</schema>
```

In diesem Beispiel existiert ein *feature*-Element mit dem Namen *Park*. *Park* ist vom Typ *ParkType*, der zwei *property*-Elemente *Flaeche* und *Eingang* enthält. Das *property*-Element *Flaeche* ist vom Typ *gml:SurfacePropertyType*, der durch GML vordefiniert ist. Das *property*-Element *Eingang* ist vom Typ *gml:PointPropertyType*. Zur Einbindung in die

4. Ein GML-basiertes Umgebungsmodell für I-LBS

Klassenhierarchie von GML wird zusätzlich ein abstrakter Typ *AbstractParkInformationType* eingeführt, über den zusätzlich die Eigenschaften des *gml:AbstractFeatureType* geerbt werden. Dazu zählt insbesondere das Attribut *gml:id* zur eindeutigen Kennzeichnung von Elementen. Das Konzept der *property types* bildet in GML eine elegante Möglichkeit, Eigenschaften zu vererben und dadurch wieder zu verwenden. Gleichzeitig können vom Entwickler auch neue, bislang nicht in GML enthaltene *property types* definiert werden, wenn dies das jeweilige Szenario erfordert.

Nach obiger Schema-Definition besteht jeder Park aus genau einer Fläche sowie einem oder mehreren Eingängen. Eine konkrete Instanz eines Parks wäre dann z.B. folgendermaßen aufgebaut:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <park:Park gml:id="park1" xmlns:park="http://www.example.org/park" xmlns:gml="http://
  www.opengis.net/gml/3.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.example.org/park http://www.example.org/park">
3   <park:Flaeche>
4     <gml:Polygon gml:id="flaeche1">
5       <gml:exterior>
6         <gml:LinearRing>
7           <gml:pos>0.0 0.0</gml:pos>
8           <gml:pos>50.0 0.0</gml:pos>
9           <gml:pos>50.0 50.0</gml:pos>
10          <gml:pos>0.0 50.0</gml:pos>
11          <gml:pos>0.0 0.0</gml:pos>
12        </gml:LinearRing>
13      </gml:exterior>
14    </gml:Polygon>
15  </park:Flaeche>
16  <park:Eingang>
17    <gml:Point gml:id="eingang1">
18      <gml:pos>25.0 0.0</gml:pos>
19    </gml:Point>
20  </park:Eingang>
21 </park:Park>
```

Die *property*-Elemente können in GML entweder „inline“, d.h. direkt an der jeweiligen Stelle im XML-Dokument oder per XLink-Attribut als Referenz eingefügt werden. So kann ein und dasselbe Element gleichzeitig an mehreren Stellen in einem Dokument verwendet werden. In dem Beispiel ist sowohl dem Park, als auch der Fläche und dem Eingang jeweils eine ID zugeordnet. Auf diese Weise können die Objekte an anderen Stellen innerhalb des Dokuments referenziert werden. So kann in diesem oder auch in einem anderen Dokument ein neues Feature definiert werden, das z.B. Parks mit einem Biergarten beschreibt. Darin müssten dann keine neuen *Park*-Objekte gespeichert werden, sondern können einfach per XLink eingebunden werden. Werden mehrere *features* als *properties* eines anderen *features* deklariert, so wird dies im GML-Standard als *feature collection* bezeichnet.

GML bietet wie bereits erwähnt eine Vielzahl von vordefinierten Objekten. Darunter sind auch bereits Elemente zur Beschreibung von verschiedensten Koordinatensystemen

4. Ein GML-basiertes Umgebungsmodell für I-LBS

enthalten, und auch ein geodätisches Datum kann mittels GML zugeordnet werden. Auf diese Weise können auch *features*, deren Georeferenzen in unterschiedlichen Koordinatensystemen angegeben sind, miteinander in Bezug gesetzt werden. Ein Beispiel dafür ist (wie auch weiter unten gezeigt) die Speicherung eines bestimmten Modells in einem lokalen Kartesischen Koordinatensystem, um so z.B. einfach und schnell Koordinaten zu verarbeiten. Indem für einige wenige Punkte (Stützpunkte) auch die Koordinaten in einem anderen Referenzsystem (z.B. WGS84) bekannt sind, können die entsprechenden Parameter für eine Koordinatentransformation bestimmt werden. Als vordefinierte Koordinatensysteme bietet GML:

- *gml:EllipsoidalCS*: zwei- oder dreidimensionales Koordinatensystem mit Längengrad, Breitengrad und optionaler Höhe.
- *gml:CartesianCS*: Kartesisches Koordinatensystem mit einer, zwei oder drei Achsen.
- *gml:VerticalCS*: ein eindimensionales Koordinatensystem zur Angabe von Höhen oder Tiefen
- *gml:TimeCS*: ein eindimensionales Koordinatensystem zur Speicherung einer zeitlichen Position auf einer Zeitachse.
- *gml:LinearCS*: ein eindimensionales Koordinatensystem, das die räumliche Entfernung eines Punktes ausdrückt. So kann beispielsweise eine Position entlang einer Pipeline gespeichert werden.
- *gml:SphericalCS*: ein Koordinatensystem zur Angabe von Punkten auf einer Kugeloberfläche.
- *gml:PolarCS*: ein zweidimensionales Polarkoordinatensystem mit Angabe durch Entfernung und Winkel.
- *gml:CylindricalCS*: ein dreidimensionales Koordinatensystem zur Speicherung einer Position auf einem Zylinder.
- *gml:AffineCS*: ein zwei- oder dreidimensionales Koordinatensystem, dessen Achsen nicht notwendigerweise orthogonal sein müssen.

Zur Umrechnung zwischen verschiedenen Referenzsystemen sieht GML eine *AbstractGeneralTransformation* vor, darin lassen sich die entsprechenden Eigenschaften der Transformation angeben. Jedoch wird dieser Abschnitt im Standard sehr allgemein gehalten. Dies könnte in einer zukünftigen Version von GML noch verbessert werden, indem z.B. eine *Spatial Reference System ID (SRID)* hinzugenommen wird, so wie dies auch schon in einem neueren OpenGIS-Standard zur Implementierung von SQL-Schemas [112] umgesetzt wurde.

4.2. Verwandte Arbeiten

GML enthält, wie bereits erwähnt, eine Sammlung vordefinierter *features*, die innerhalb eines konkreten Anwendungsschemas verwendet werden können, d.h. es handelt sich bei GML nicht um eine fertige Sprache, die als solche in ein System übernommen werden kann. Nachfolgend sollen einige verwandte Arbeiten zur Beschreibung von Umgebungsmodellen betrachtet werden. Einige dieser Ansätze bauen auf GML auf, einige verwenden ähnliche Konzepte zur Beschreibung von *features* bzw. Geo-Objekten.

CityGML Die *City Geography Markup Language* ist ein GML Anwendungsschema und wird seit 2002 von der Special Interest Group 3D entwickelt. Seit 2008 liegt CityGML als OGC-Standard vor und definiert ein semantisches Informationsmodell zur Beschreibung von städtischen 3D-Objekten [109]. Es ist vorrangig zur Darstellung von Städten, Stadtteilen und Bauwerken konzipiert, dabei werden fünf unterschiedliche so genannte *Level of Detail (LOD)* definiert, wobei die detaillierteste Stufe LOD4 das Innere eines Gebäudes beschreibt. Die Basisklasse aller Klassen ist das *CityObject* als Unterklasse von GML *Feature*. Im Kern-Modul des Standards werden abstrakte Basisklassen definiert, die von entsprechenden Erweiterungs-Modulen implementiert werden. CityGML unterscheidet folgende Erweiterungs-Module:

- *Appearance*: beschreibt Eigenschaften der Oberfläche eines *features*. Diese sind nicht auf sichtbare Eigenschaften begrenzt, sondern können auch beispielsweise Infrarot-Abstrahlung, Lärm oder durch Erdbeben verursachte Belastungen des Gebäudes ausdrücken.
- *Building*: beschreibt die räumlichen Aspekte von Gebäuden, Gebäudeteilen und Installationen im Gebäude.
- *CityFurniture*: dient zur Repräsentation von unbeweglichen Objekten in einer Stadt, wie z.B. Laternen, Ampeln, Parkbänken oder Bushaltestellen.
- *CityObjectGroup*: ermöglicht eine Zusammenfassung beliebiger *CityObjects* zu einer Gruppe
- *Generics*: erlaubt eine Erweiterung des CityGML Datenmodells durch Hinzufügen beliebiger zusätzlicher Attribute und Features.
- *LandUse*: beschreibt Landflächen für die eine bestimmte Nutzung vorgesehen ist.
- *Relief*: modelliert das Terrain in einem Stadtmodell
- *Transportation*: beschreibt *features*, die für den Transport in einer Stadt verwendet werden können. Dazu zählen beispielsweise Straßen, Wege, Schienen oder auch Plätze.
- *Vegetation*: modelliert die Vegetation, z.B. Bäume, Biotope, Wälder oder andere Pflanzenansammlungen.

4. Ein GML-basiertes Umgebungsmodell für I-LBS

- *Water-Body*: beschreibt die Ausmaße von Flüssen, Kanälen, Bächen, Becken oder Seen.
- *TexturedSurface (deprecated)*: weist einem feature Eigenschaften wie z.B. Farbe, Glanz oder Transparenz zu. Dieses Modul wird in zukünftigen Versionen nicht mehr verfügbar sein und wird durch das Appearance-Modul ersetzt.

Die Modellierung von Gebäuden wird in dem *Building-Modul* von CityGML beschrieben. Bis zum LOD3 werden nur von außen sichtbare Merkmale des Gebäudes abgebildet, so dass sich dessen äußere Form in verschiedenen Genauigkeiten darstellen lässt. Im LOD4 können auch einzelne Räume modelliert werden, einschließlich Fenster, Türen und Möbel. Fenster, Türen und Räume werden durch *gml:MultiSurface* beschrieben, Letztere alternativ auch durch *gml:solid*. Damit deckt CityGML zahlreiche Informationen ab, die für eine Visualisierung von Gebäuden benötigt werden.

Zwar ist durch die Hierarchie zwischen *Building*, *BuildingPart* und *Room* eine gewisse Enthaltensein-Beziehung vorgegeben, aber CityGML sieht weder eine explizite Speicherung der Verbindungen zwischen Räumen vor, noch ist ein Graph zur Speicherung von Wegenetzen vorhanden. Hinzu kommt, dass alle Assoziationen in CityGML nur unidirektional sind. Diese für eine Visualisierung sicherlich sinnvolle Abbildung verhindert jedoch, dass für ein gegebenes Objekt direkte Rückschlüsse auf dessen Referenzierungen möglich sind. Dadurch sind selbst die Verbindungen zwischen Räumen nur implizit jeweils durch zwei unidirektionale Beziehungen zu der jeweiligen Tür gegeben.

Generell bietet CityGML ein umfangreiches Konzept zur Modellierung von Städten in unterschiedlichen Stufen der Genauigkeit. In Bezug auf die Verwendung als Indoor-Umgebungsmodell fehlt jedoch die Möglichkeit, ein Wegenetz und Orte und Bereiche von Interesse explizit abzubilden.

OpenLS Im Jahr 2003 hat das Open Geospatial Consortium mit der *Open GIS Location Service (OpenLS) Implementation Specification* [110] eine Architektur für LBS vorgestellt. Auch als GeoMobility-Server bekannt werden darin verschiedene Funktionalitäten, Schnittstellen und Formate für den Betrieb von LBS beschreiben. Abbildung 4.1 zeigt einen Überblick über die Architektur. OpenLS unterscheidet zwischen fünf verschiedenen Kern-Diensten: Der *Directory Service* ist ein Dienst zum Auffinden des nächstgelegenen oder eines spezifischen Ortes, Produktes oder Dienstes. Ein *Gateway Service* stellt eine Schnittstelle zwischen dem GeoMobility-Server und einem Location-Provider her und entspricht den Vorgaben des *Mobile Location Protocol* [113]. Darunter fallen die Anfragen eines Clients nach der einmaligen, periodischen oder getriggerten Position eines Zielobjekts. Der *Location Utility Service* ist ein Dienst zur Berechnung einer geographischen Position für einen gegebenen symbolischen Bezeichner wie z.B. Straßename oder Postleitzahl (Geocoding), bzw. zur Berechnung der postalischen Adresse zu einer gegebenen geographischen Koordinate (Reverse geocoding). Durch den *Presentation Service* werden geographische Informationen zur Anzeige auf einem mobilen Endgerät gerendert. Der *Route Service* wurde in [111] zum *Navigation Service* erweitert. Letzterer sieht vor, dass ein Benutzer anhand einer Start- und Zielposition die Berechnung einer Route beim GeoMobility-Server anfordern

4. Ein GML-basiertes Umgebungsmodell für I-LBS

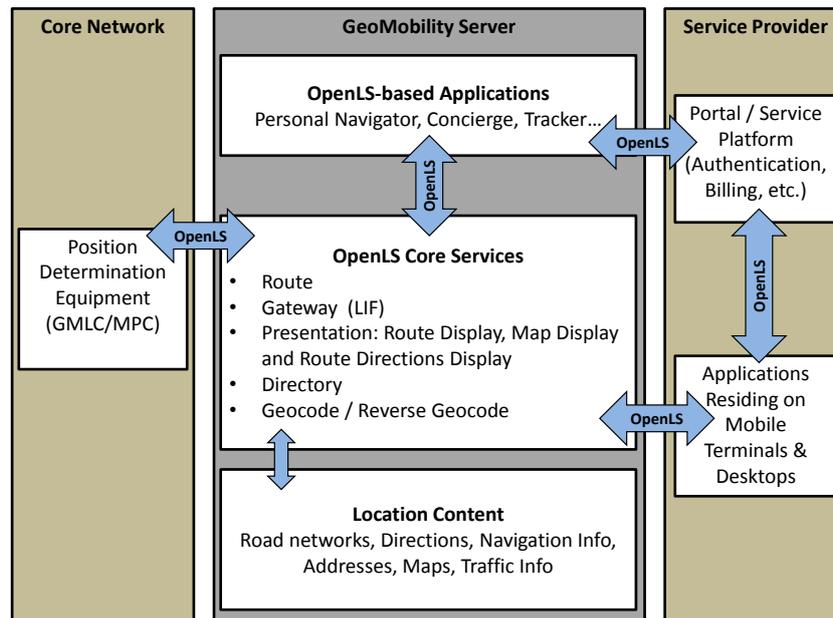


Abbildung 4.1.: GeoMobility-Architektur des OpenGIS Location Service. Quelle: [110]

kann und daraufhin eine Liste von Punkten entsprechend dem Weg zum Ziel zurückgeliefert wird. Für OpenLS spielt auch die OGC-Spezifikation für den *Simple Feature Access* [106] eine wichtige Rolle. Darin werden grundlegende geometrische Objekte definiert und insbesondere auch Formate zur Darstellung der Objekte als *Well-Known-Text* und *Well-Known-Binary* vorgestellt. Damit lassen sich jedoch nur geometrische Objekte serialisieren, eine Beschreibung der topologischen Zusammenhänge ist nicht möglich.

Industry Foundation Classes IFC [94] ist ein etablierter Standard für den Austausch von digitalen Gebäude-Bauplänen. In einem IFC-Modell werden zahlreiche Informationen über ein Gebäude gespeichert, wie beispielsweise Bauteile, verwendete Materialien, Kosten, geometrische Abmessungen und viele andere Details. Dabei wird ein objektorientierter Ansatz verfolgt, der (ähnlich wie bei GML) eine gute Erweiterbarkeit ermöglicht. Bei IFC handelt es sich um ein komplexes Modell, das für den Einsatz im Bauwesen konzipiert ist und das die Anforderungen von I-LBS bei weitem übersteigt. Jedoch sieht auch IFC keine Beschreibung von Wegenetzen vor, und die zahlreichen Details des Modells bieten sich nach Fertigstellung des Gebäudes eher zur Verwendung in der Gebäude-Verwaltung an. Ein Raum ist zwar nur implizit durch die umgebenden Wände beschrieben, aber der Standard sieht unter anderem ein *IfcSpace*-Element vor, mit dessen Hilfe sich ein oder mehrere Bereiche innerhalb desselben Stockwerks abbilden lassen.

KML Die *Keyhole Markup Language (KML)* [108] ist eine Sprache zur Speicherung von geographischen Informationen mit dem Schwerpunkt auf der Visualisierung der Daten.

4. Ein GML-basiertes Umgebungsmodell für I-LBS

KML zeichnet sich durch ein einfaches, auch von Menschen leicht zu lesendes Format aus und auch hier wird ein objektorientiertes Modell zugrunde gelegt. Zahlreiche vordefinierte Elemente wie z.B. eine Landmarke oder ein Overlay-Bild bilden ein Grundgerüst, mit dessen Hilfe festgelegt werden kann, was von der Umgebung angezeigt werden soll und wie die Darstellung zu erfolgen hat. Die geometrische Form von Objekten lässt sich anhand von Punkten, Linien, Polygonen etc. beschreiben, die mittels WGS84-Koordinaten gespeichert werden. Auch bei KML fehlt jedoch die Möglichkeit, eine Topologie für Orte abzubilden.

Nexus Nicklas [103] beschreibt mit der *Augmented World Modeling Language (AWML)* eine Sprache zum Austausch von Umgebungsmodelldaten für die *Nexus-Plattform* (vgl. Rothermel et al. [122] und Becker et al. [16]). AWML sieht umfangreiche Möglichkeiten zur Modellierung der Umgebung vor und ordnet alle Objekte in einer flachen Hierarchie an, in der die Objekte über Verweise miteinander verbunden sind. Zur Beschreibung von Gebäuden werden Räume, Stockwerke und Wohnungen unterschieden. Dabei ist die Enthaltensein-Beziehung jedoch nur implizit über die geometrische Ausdehnung eines Objektes gegeben. Für die Topologie von Straßen bietet AWML eine Beschreibung anhand von *NavigationNodes* und *NavigationEdges*. Ein Vorteil ist dabei, dass eine Trennung zwischen den geometrischen Objekten und den *NavigationalObjects* erfolgt. Innerhalb von Gebäuden ist jedoch keine Beschreibung von Wegen möglich, da auch keine Verbindungen zwischen Räumen modelliert werden.

OpenStreetMap Die OpenStreetMap² (OSM) wurde 2004 gegründet und ist ein freies Projekt für die Sammlung, Bearbeitung und Bereitstellung von Geodaten. Darin werden drei verschiedene Basistypen verwendet: *Node*, *Way* und *Relation*. Diese können jeweils mit einer beliebigen Anzahl von Schlüssel-Wert-Paaren (*Tags*) versehen sein. Ein *Node* enthält einen Längen- und Breitengrad (optional eine Höhenangabe) sowie eine eindeutige ID. Ein *Way* besteht aus einer geordneten Liste von Nodes, er kennzeichnet zum Beispiel einen Straßenabschnitt oder einen Wanderweg. Eine *Relation* dient der Gruppierung von zwei oder mehr *Way*- oder *Node*-Elementen. Jeder *Relation* ist ein Schlüssel mit dem Namen *type* zugeordnet, dieser kennzeichnet die Art der Relation. Die genaue Beschreibung von Geo-Objekten basiert in der OSM auf Konventionen für die Verwendung von bestimmten *tags* bzw. für die Reihenfolge der Objekte. Eine Fläche (*Area*) kann beispielsweise dadurch beschrieben werden, dass der erste und letzte *Node* eines *Way* identisch sind. Durch die Freiheit in der Wahl der *tag*-Bezeichner kann das Modell schnell und einfach erweitert werden. Gleichzeitig kann dies aber auch ein Problem darstellen, wenn eine Namenskonvention nicht allen bekannt ist, nicht von jedem eingehalten wird oder durch ein Tool noch nicht unterstützt wird.

Das OSM-Modell unterstützt somit gleichermaßen geometrische Eigenschaften (in Form von WGS84-Koordinaten) und symbolische Bezeichner (in Form von *tags* bzw. *Relationen*). Das Enthaltensein von Punkten in einer Fläche wird jedoch nur implizit abgebildet,

²<http://www.openstreetmap.org>

4. Ein GML-basiertes Umgebungsmodell für I-LBS

da ein *Node* zwar einem *Way*, nicht aber einer Fläche zugeordnet wird. Auch existieren bislang keine Ansätze zur Unterscheidung von Stockwerken oder einer Topologie von Räumen.

In den vorgestellten verwandten Arbeiten werden zum Teil unterschiedliche Herangehensweisen zur Modellierung der Umgebung bzw. Umwelt verfolgt. Auf der einen Seite stehen Ansätze mit exakt vorgegebenen Hierarchien und genauen Regeln für die Modellierung, auf der anderen Seite existieren sehr freie Formate mit wenig genauen Vorgaben und vielen Konventionen.

In einem weiteren Ansatz von Becker et al. [17] wird ein vielversprechendes Modell vorgeschlagen, bei dem der aktuelle Kontext (z.B. die Art der Bewegung oder das verwendete Positionierungsverfahren) auf unterschiedlichen Schichten abgebildet wird. Die Umgebung in einem Gebäude kann dabei sowohl in einem Euklidischen, als auch in einem topologischen Raum beschrieben werden. Durch Kombination der Schichten kann so beispielsweise der Empfangsbereich eines WLANs auf verschiedene Räume abgebildet werden.

4.3. Ein Umgebungsmodell für Indoor LBS

In diesem Abschnitt wird ein neues GML-Anwendungsschema zur Speicherung von Umgebungsmodellen für Indoor LBS vorgestellt: die *Building Information Geography Markup Language (BIGML)*. Das Ziel von BIGML ist, für die Beschreibung von Gebäuden ein einheitliches Modell bereit zu stellen, das alle Informationen beinhaltet, die von einem Akteur in der Rolle des Modell-Providers (vgl. Abbildung 2.1) benötigt werden. Dadurch können einerseits die Modelle von verschiedenen Gebäuden bei einem Modell-Provider einheitlich verarbeitet werden, andererseits können Modelle einfach zwischen verschiedenen Modell-Providern ausgetauscht werden. BIGML wurde in [74] vorgestellt und eine ausführliche Diskussion findet sich in [73]. Die Grundlage für BIGML bildet GML in der Version 3.2.1 [107].

Die grundlegenden Elemente zur Beschreibung von realen Objekten in BIGML sind das Grundstück (*Site*), das Gebäude (*Building*), das Stockwerk (*Floor*) und der Raum (*Room*). Daneben existieren Elemente zur Beschreibung von Orten von Interesse (*POI*), Bereichen von Interesse (*AOI*) und Übergängen (*Portal*) zwischen verschiedenen Orten. Allen Elementen können durch *property elements* verschiedene Eigenschaften wie beispielsweise eine Position, ein Polygon, ein Identifikator oder auch frei definierbare Tags zugeordnet werden.

Neben Lage und Form von Orten und Objekten ist auch ein Wegenetz (*Graph*) vorgesehen, das aus einer Menge von Knoten (*Node*) und Kanten (*Edge*) besteht. Die Art des Wegenetzes ist nicht vorgegeben und je nach Anwendungsfall können die Anzahl der Knoten sowie der Verzweigungsgrad angepasst werden. Dadurch ist es möglich, auf unterschiedliche Arten von Diensten einzugehen. So kann ein einfacher Corner Graph verwendet werden, bis hin zu beispielsweise einem sehr feinmaschigen rasterbasierten Wegenetz.

Gemäß der Klassifikation in Abschnitt 3.1 handelt es sich um ein hybrides Modell, das sowohl geometrische als auch symbolische Koordinaten unterstützt. Gleichzeitig sind die realen Objekte hierarchisch organisiert sowie über einen Graphen miteinander verbun-

4. Ein GML-basiertes Umgebungsmodell für I-LBS

den. Um zusätzlich auch Orte auf derselben Hierarchiestufe zusammenfassen zu können (mengenbasiertes Modell), wird das weiter unten beschriebene Konzept der Bereiche von Interesse verwendet.

Das UML-Diagramm in Abbildung 4.2 gibt einen Überblick der Klassen in BIGML. Das vollständige Schema findet sich im Anhang A und ist online³ abrufbar.

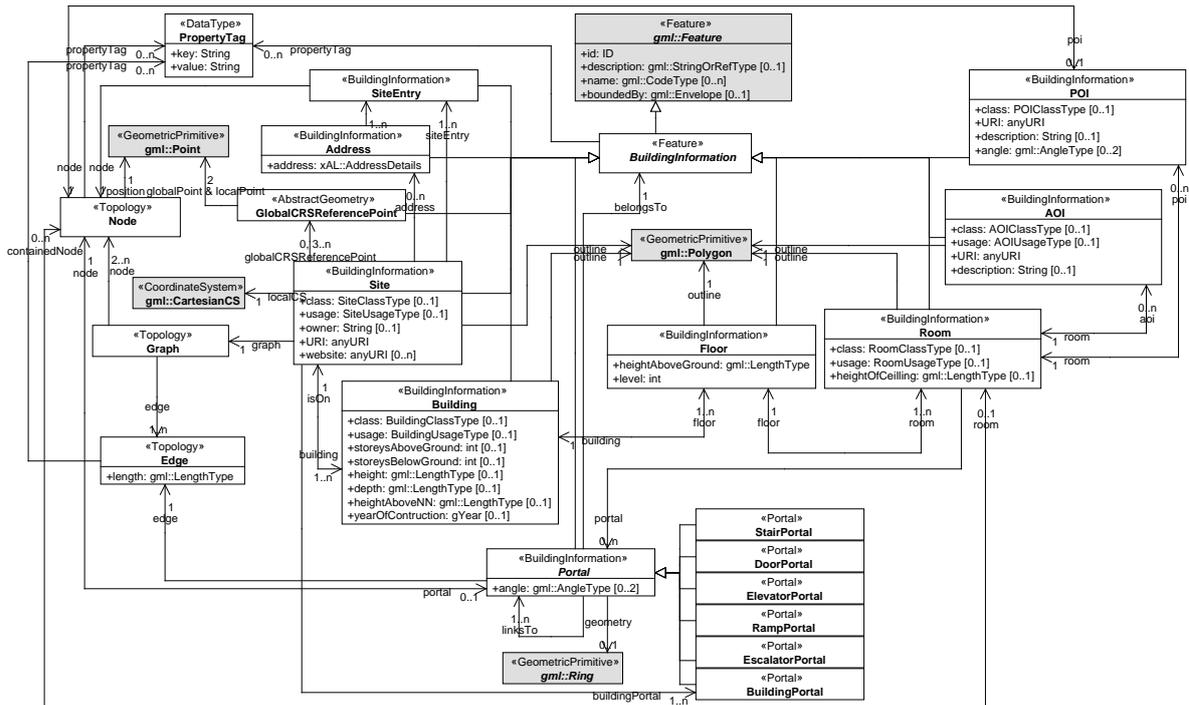


Abbildung 4.2.: Übersicht der Klassen in BIGML [74].

BuildingInformation: Der *AbstractBuildingInformationType* bildet die Grundlage für alle realen Objekte des Gebäudes. Er erweitert den *AbstractFeatureType* von GML und erbt dadurch insbesondere *gml:id*. Der *AbstractBuildingInformationType* enthält zusätzlich eine beliebige Anzahl von *PropertyTags*, dadurch kann jedes *BuildingInformation*-Element bei Bedarf durch diese Schlüssel-Wert-Paare angereichert werden. Gemäß den Vorgaben des GML-Standards erfolgt dazu die Definition eines globalen Elements *AbstractBuildingInformation*, des komplexen Typs *AbstractBuildingInformationType*, sowie eines zugehörigen *AbstractBuildingInformationPropertyType*. Der PropertyType beinhaltet lediglich eine Referenz auf ein *AbstractBuildingInformation*-Element und dient dazu, in einem konkreten Dokument eine Beziehung zu einem Objekt dieses Typs herzustellen. Diese Tripel (*Element*, *Type*, *PropertyType*) finden sich bei allen der folgenden Objekte wieder.

³Das BIGML-Schema ist unter <http://www.mobile.ifi.lmu.de/schemas/bigml> abrufbar.

4. Ein GML-basiertes Umgebungsmodell für I-LBS

Site: Abbildung 4.3 zeigt im Überblick die Eigenschaften und zugehörigen Elemente einer *Site*. Die *Site* repräsentiert ein zusammenhängendes Grundstück, das über einen oder mehrere Zugänge (*SiteEntry*) mit der Außenwelt verbunden ist. Einer *Site* können eine oder mehrere Adressen (*Address*) zugeordnet sein. Innerhalb einer *Site* befinden sich ein oder mehrere Gebäude (*Building*), deren Zugänge auf dem Grundstück durch das *BuildingPortal* beschrieben werden. Zusätzlich besitzt eine *Site* Informationen über das verwendete lokale Koordinatensystem (*localCS*) sowie eine Liste von mindestens drei verschiedenen Referenzpunkten (*GlobalCRSReferencePoint*) innerhalb der *Site*, zu denen entsprechende Koordinaten innerhalb eines globalen Referenzsystems (z.B. WGS84 oder UTM) bekannt sind. Dies ermöglicht die spätere Berechnung der für die Transformation in andere Koordinatensysteme benötigten Transformationsparameter. Die Attribute *usage* und *class* wurden von CityGML übernommen und beschreiben die Nutzung bzw. Art des Grundstücks näher. Jede *Site* enthält zusätzlich ein Wegenetz (*Graph*), das weiter unten vorgestellt wird. Der obligatorische URI identifiziert ein Grundstück eindeutig.

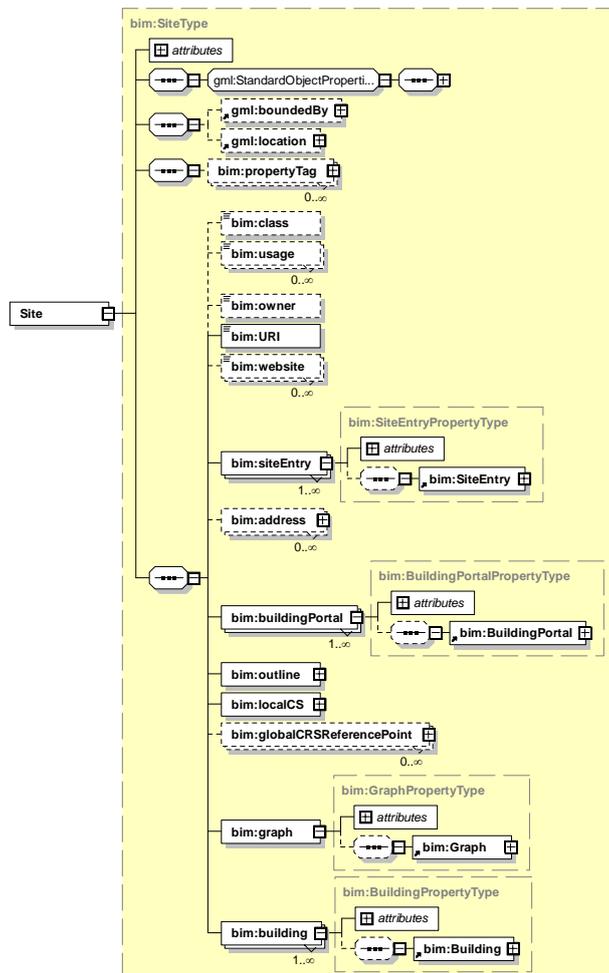


Abbildung 4.3.: Aufbau eines Grundstücks (*Site*).

4. Ein GML-basiertes Umgebungsmodell für I-LBS

Building: Ein Gebäude (*Building*) ist genau einer *Site* zugeordnet und enthält ein oder mehrere Stockwerke (*Floor*). Zusätzlich können jedem Gebäude weitere Eigenschaften wie z.B. Höhe, Tiefe, Anzahl der Stockwerke und Kellergeschosse, Baujahr, das umgebende Polygon (Umriss), sowie beliebige weitere *PropertyTags* zugeordnet werden (siehe Abbildung 4.4). Die Werte der Attribute *class* und *usage* richten sich innerhalb eines *Building* nach dem ALKIS-Objektartenkatalog der *Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland (AdV)*, siehe auch [7, S. 190]. Beispiele für die Klasse eines Gebäudes sind *Einkaufszentrum* (Code 2052), *Hochschulgebäude* (Code 3023) oder *Krankenhaus* (Code 3051). Im ATKIS-Objektartenkatalog sind insgesamt mehrere hundert verschiedene Möglichkeiten zur Beschreibung vorgesehen.

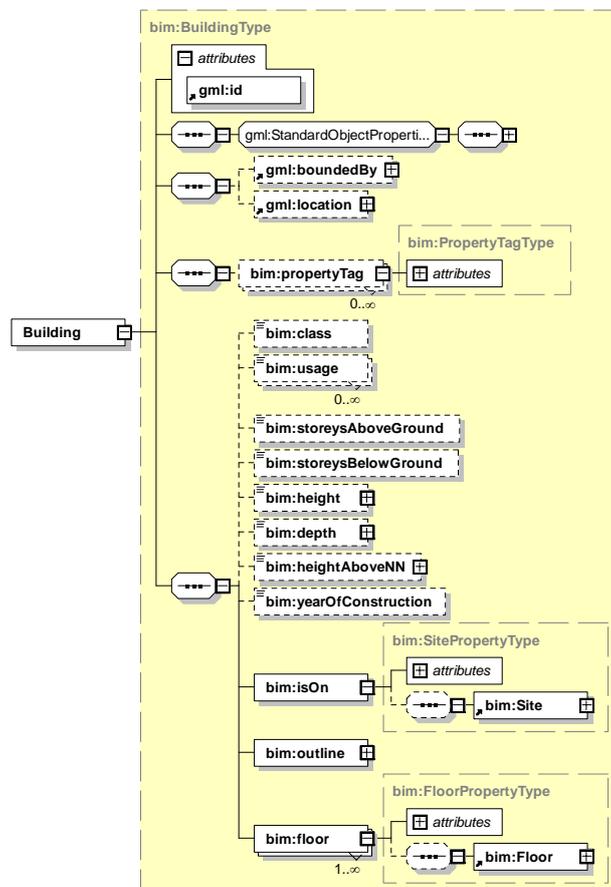


Abbildung 4.4.: Aufbau eines Gebäudes (*Building*).

Floor: Ein Stockwerk ist genau einem *Building* zugeordnet und umfasst eine Menge von einem oder mehreren Räumen (*Room*) (siehe Abbildung 4.5). Das Integer-Attribut *level* gibt das Stockwerk an, wobei 0 das Erdgeschoss bezeichnet. Das Attribut *heightAboveGround* bezeichnet die Höhe des Bodens des Stockwerks über dem Boden von *level 0*. Jedes

4. Ein GML-basiertes Umgebungsmodell für I-LBS

Stockwerk beinhaltet ebenso einen Umriss (*outline*), dieser entspricht der Außenseite des Gebäudes auf dem jeweiligen Stockwerk.

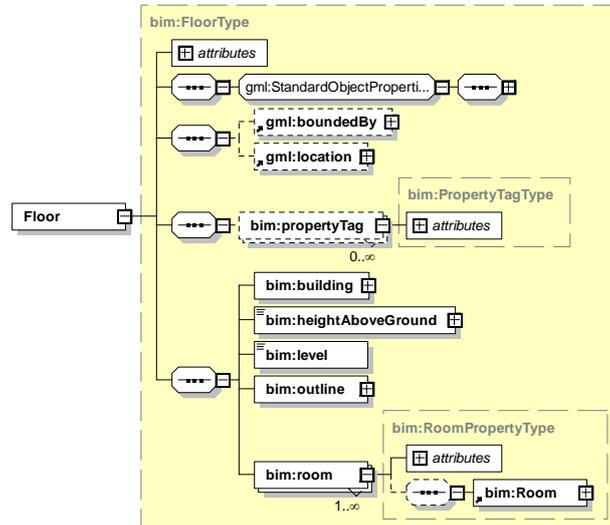


Abbildung 4.5.: Aufbau eines Stockwerks (*Floor*).

Room: Abbildung 4.6 zeigt die Elemente eines Raumes (*Room*). Jedem Raum ist ein umgebendes Polygon (*outline*) zugeordnet, das die grundsätzliche Form des Raumes beschreibt. Die Angabe dieses umgebenden Polygons erfolgt durch ein *gml:Polygon*. Die Verbindungen zwischen Räumen (wie z.B. Türen, Durchgänge, etc.) werden durch Portale (*PortalPropertyType*) modelliert. Einem Raum können zusätzlich Orte von Interesse (*POIPropertyType*) und Bereiche von Interesse (*AOIPropertyType*) zugeordnet werden. Jeder Raum enthält des Weiteren Verweise auf alle Knoten (*NodePropertyType*) des Wegenetzes, die innerhalb des umgebenden Polygons liegen (*containedNode*). Dadurch ist es z.B. möglich, für eine Routenberechnung den im Raum am besten geeigneten Start- bzw. End-Knoten zu finden.

AbstractPortal: Portale charakterisieren Verbindungen zwischen verschiedenen Elementen einer *Site*. Der *AbstractPortalType* (siehe Abbildung 4.7) legt die Eigenschaften für verschiedene konkrete Portale fest. Jedes *AbstractPortal* ist über ein *belongsTo*-Element genau einem *AbstractBuildingInformation*-Element zugeordnet. Über ein oder mehrere *linksTo*-Elemente ist das *AbstractPortal* mit einem oder mehreren anderen Portalen verbunden (mehrere Verbindungen treten zum Beispiel bei einem Fahrstuhl auf). Das *node*-Element verweist auf den zugehörigen Knoten im Wegenetz und damit auch gleichzeitig auf die Position des Portals. Der zusätzliche Verweis auf das *edge*-Element dient zur Optimierung des Graphen für die Wegfindung, dadurch können im Vorhinein bestimmte Arten von Kanten (z.B. Treppen oder Gebäude-Ausgänge) heraus gefiltert werden. Optional kann

4. Ein GML-basiertes Umgebungsmodell für I-LBS

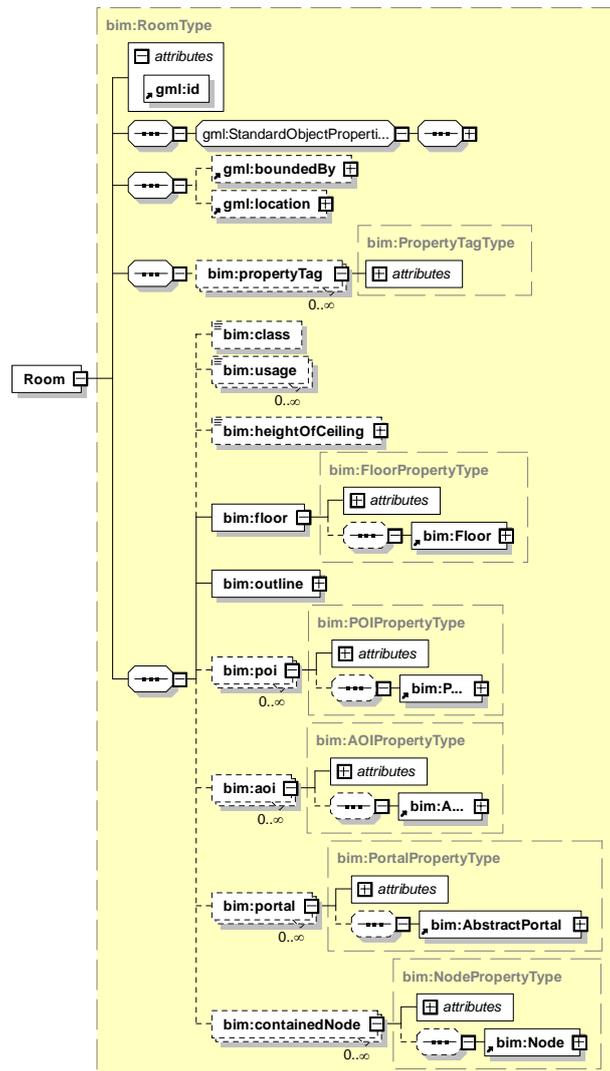


Abbildung 4.6.: Aufbau eines Raumes (*Room*).

die Richtung des Portals durch das *angle*-Element anhand eines *gml:AngleType* angegeben werden und das *geometry*-Element beschreibt die Form des Portals anhand eines *gml:AbstractRingPropertyType*.

Das *AbstractPortal* kann durch verschiedene Typen von konkreten Portalen instantiiert werden. Beispielsweise beschreibt ein *BuildingPortal* einen Ein- bzw. Ausgang eines Gebäudes. Mittels eines *DoorPortal* werden Übergänge zwischen Räumen ausgedrückt, analog existieren *ElevatorPortal*, *StairPortal* oder auch *RampPortal*. Je nachdem, ob es sich um einen Übergang zwischen zwei Räumen oder zwischen einem Raum und dem Außenbereich der *Site* handelt, verweist das *belongsTo*-Element auf einen *Room* oder auf die *Site*.

Mit Hilfe der Portale wird erreicht, dass eine flexible Trennung zwischen der topologi-

4. Ein GML-basiertes Umgebungsmodell für I-LBS

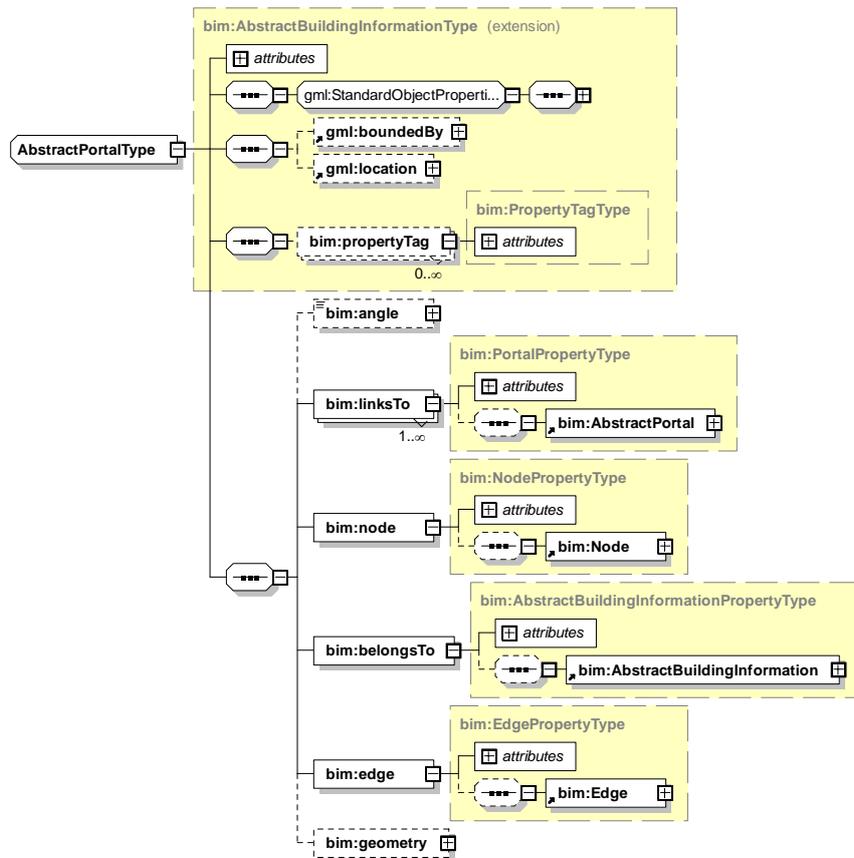


Abbildung 4.7.: Aufbau des abstrakten Portals (*AbstractPortalType*).

schen Verbindung von Räumen einerseits und andererseits der Begehbarkeit jedes einzelnen Raumes möglich ist. Unabhängig davon, ob der Raum eine konvexe oder konkave Form besitzt, legen Portale die statischen Verbindungen zu anderen Räumen fest und definieren dadurch bereits ein grobes Wegenetz. Sofern noch kein feinmaschiges Wegenetz für die Bereiche innerhalb der konkaven Räume existiert, kann dieses auch immer aus den *outlines* und Portalen der Räume generiert werden.

POI: Ein Ort von Interesse (*POI*) charakterisiert einen bestimmten Ort innerhalb der *Site* (vgl. Abbildung 4.8). Dabei handelt es sich um einen beliebigen Ort, dessen Position sich durch eine einzelne Koordinate angeben lässt, d.h. das der *POI* entweder eine zu vernachlässigende Ausdehnung besitzt oder aber dass er eindeutig anhand der Koordinate zu finden ist. Dazu ist jedem *POI* genau ein Knoten (*Node*) im Wegenetz zugeordnet, sowie ein eindeutiger URI und weitere Attribute wie z.B. die zugehörige Klasse (*class*) oder eine freie Beschreibung (*description*). Durch die optionale Angabe eines Winkels (*angle*) kann, sofern zutreffend, die Ausrichtung des *POIs* angegeben werden. Der Verweis auf den zugehörigen Raum erfolgt anhand des *room*-Elements des *POI*. Wie auch bei den anderen Elementen können dem *POI* über *PropertyTags* weitere Eigenschaften zugeordnet

4. Ein GML-basiertes Umgebungsmodell für I-LBS

werden, beispielsweise ein Verweis auf ein Foto, eine Webseite, einen Ansprechpartner, eine Bedienungsanleitung oder Öffnungszeiten.

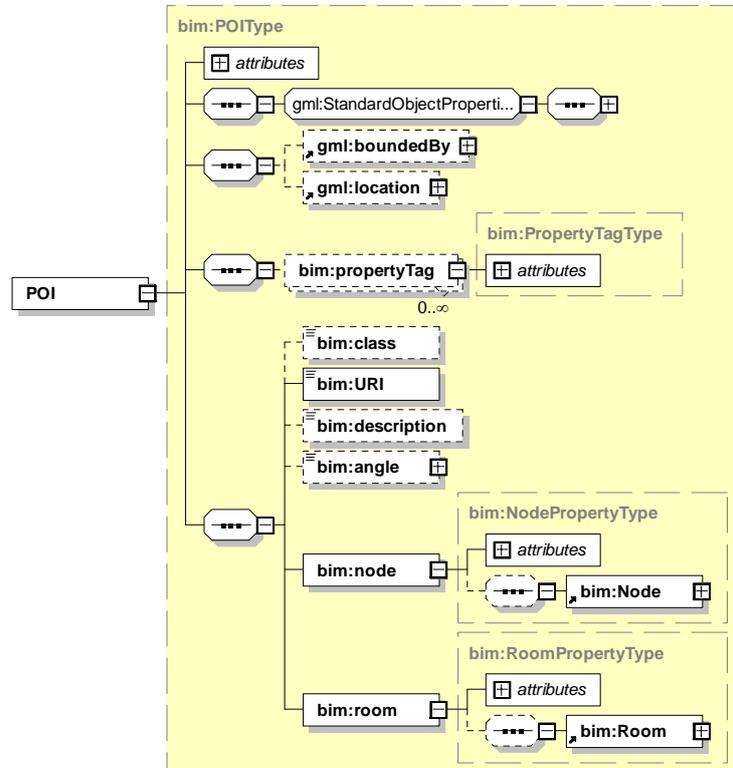


Abbildung 4.8.: Aufbau eines Ortes von Interesse (*POI*).

AOI: Mit der *Area of Interest (AOI)* wird in BIGML die Möglichkeit geschaffen, einen Bereich von Interesse zu modellieren (siehe Abbildung 4.9). Der von einer AOI umfasste Bereich wird durch ein Polygon (*gml:Polygon*) beschrieben, das nicht notwendigerweise deckungsgleich mit der *outline* des beinhaltenden Raumes sein muss. Mehrere nicht zusammenhängende Bereiche können zu einem größeren Bereich von Interesse zusammengefasst werden, indem sie sich jeweils als *AOI*-Elemente einen URI teilen. Auf diese Art und Weise können auch *AOIs* miteinander in Bezug gesetzt werden, die sich auf unterschiedlichen Stockwerken oder in unterschiedlichen Gebäuden befinden. Abbildung 4.10 zeigt ein Beispiel, in dem Räume verschiedenen *AOIs* zugeordnet sind.

Wegenetz: Der *Site* ist ein *Graph* zugeordnet, der aus einer Menge von Knoten (*Node*) und Kanten (*Edge*) besteht (siehe Abbildung 4.11). Dieser Graph bildet ein Wegenetz und charakterisiert die möglichen Wege durch das Gebäude. Jeder Knoten enthält Verweise auf die ihm zugehörigen Elemente wie z.B. den Raum, das Portal oder den *POI*. Zusätzlich ist jedem Knoten eine Position zugeordnet. Jede Kante besitzt eine Länge, sowie genau zwei Verweise auf die der Kante zugehörigen Knoten. Die Oberklasse von *Node* und *Edge*

4. Ein GML-basiertes Umgebungsmodell für I-LBS

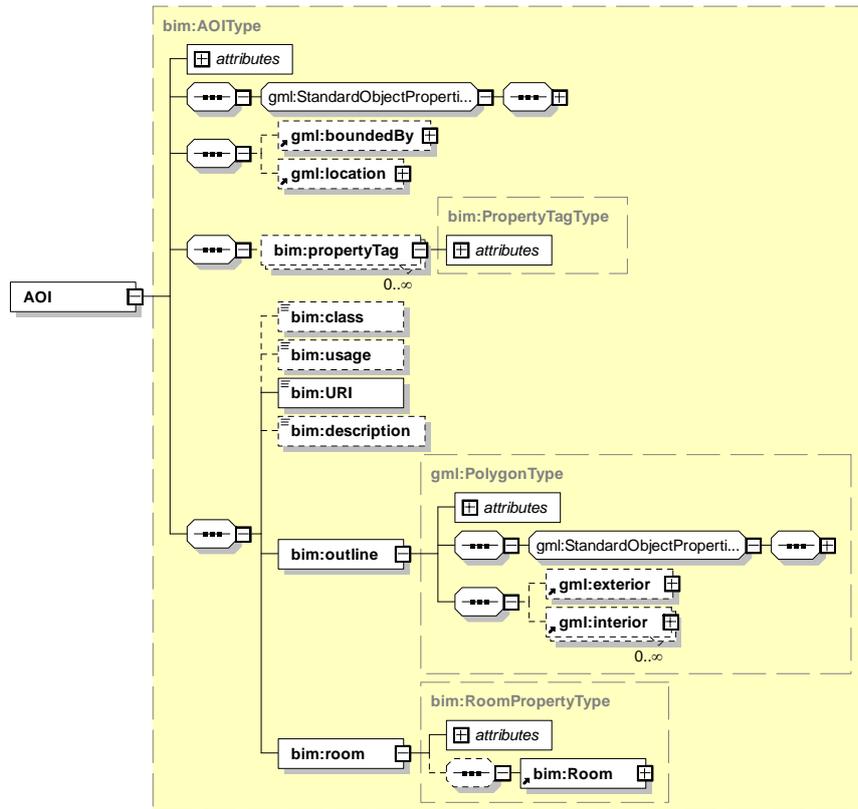


Abbildung 4.9.: Aufbau eines Bereiches von Interesse (AOI).

ist nicht *AbstractBuildingInformation*, sondern *gml:AbstractTopology*. Daher enthalten *Node* und *Edge* separate *PropertyTag*-Definitionen. Zusammen mit den zuvor vorgestellten Elementen ergibt sich für den *Graph*, dass dieser mindestens die entsprechenden Knoten und Kanten für Portale, Eingänge zum Gelände (*SiteEntry*) sowie für die POIs enthalten muss. Weitere zusätzliche Knoten entstehen abhängig von dem gewählten Ansatz für die Berechnung des Wegenetzes. Auf diese Art und Weise lassen sich nun für ein Gebäude (bzw. für eine *Site*) verschiedene Arten von Wegenetzen erzeugen. Die einzige Anforderung ist dabei, dass das Wegenetz innerhalb eines jeden Raumes (vgl. Abschnitt 3.4) an die Portal-Knoten angebunden ist.

Da alle Koordinaten, Kantengewichte und Eigenschaften der Kanten explizit im Graphen gespeichert sind, kann dieser direkt, d.h. nach Einlesen des XML-Dokuments, von einem beliebigen Algorithmus zur Suche in Graphen verarbeitet werden. Anhand der Kanten zwischen Portalknoten von Räumen bzw. Stockwerken und Gebäuden ist gleichzeitig auch eine Unterteilung vorhanden, die die Bildung eines hierarchischen Graphen vereinfacht.

Referenzkoordinaten: Abbildung 4.12 zeigt den Aufbau des *GlobalCRSReferencePointType*. Dieser beinhaltet zwei Elemente *localPoint* und *globalPoint* vom Typ *gml:Point*. Beide entsprechen derselben realen Position, jedoch in unterschiedlichen Koordinatensy-

4. Ein GML-basiertes Umgebungsmodell für I-LBS

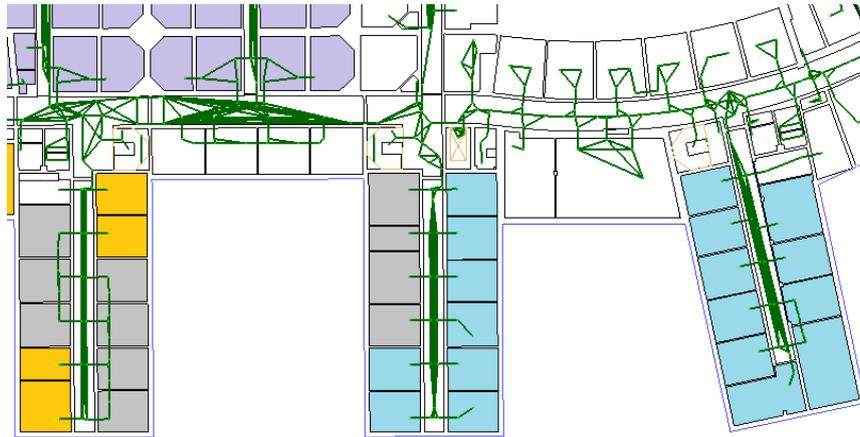


Abbildung 4.10.: Zusammenfassung von Räumen zu Bereichen von Interesse: unterschiedliche Farben (gelb, grau, lila, blau) repräsentieren jeweils eine Area of Interest (AOI). Grüne Linien und Punkte zeigen eine konkrete Instanz eines Wegpunktgraphen.

stemem. Anhand von drei verschiedenen Instanzen eines *GlobalCRDReferencePoint* können so die Transformationsparameter zur Umrechnung zwischen den beiden verwendeten Koordinatensystemen bestimmt werden. Das für den *localPoint* verwendete Koordinatensystem entspricht dabei dem innerhalb einer *Site* verwendeten Koordinatensystem. Mögliche Koordinatensysteme für die Speicherung des *globalPoint* sind beispielsweise Gauß-Krüger, GRS80 oder auch WGS84. Bei der Wahl der verschiedenen Referenzpunkte sollte darauf geachtet werden, dass diese innerhalb der *Site* möglichst weit voneinander entfernt liegen und nicht kollinear sein, um so die Genauigkeit bei der Bestimmung der Transformationsparameter zu erhöhen.

4.4. Bewertung und Diskussion

Mit *BIGML* wurde ein neues Umgebungsmodell für ortsbezogene Dienste in Gebäuden vorgestellt. Es basiert auf der *Geography Markup Language* und ist als GML-Anwendungsschema umgesetzt. Die Objekte der realen Welt sind hierarchisch organisiert, und die geometrischen Merkmale werden in einem lokalen Koordinatensystem abgebildet. Gleichzeitig beschreiben Portale die Übergänge zwischen Bereichen und kennzeichnen deren Verbundenheit. Das Wegenetz repräsentiert die Begehbarkeit des Gebäudes und kann je nach Anforderung eines Dienstes unterschiedliche Formen annehmen. Des Weiteren können Orte und Bereiche von Interesse in dem Modell gespeichert werden.

Im Folgenden werden die Eigenschaften des Modells genauer mit den im Abschnitt 3.2 vorgestellten Anforderungen verglichen:

- *Abbildung von Positionsdaten:* Die Positionsdaten aus unterschiedlichen Positionierungsverfahren können integriert werden. Handelt es sich um geometrische Koordina-

4. Ein GML-basiertes Umgebungsmodell für I-LBS

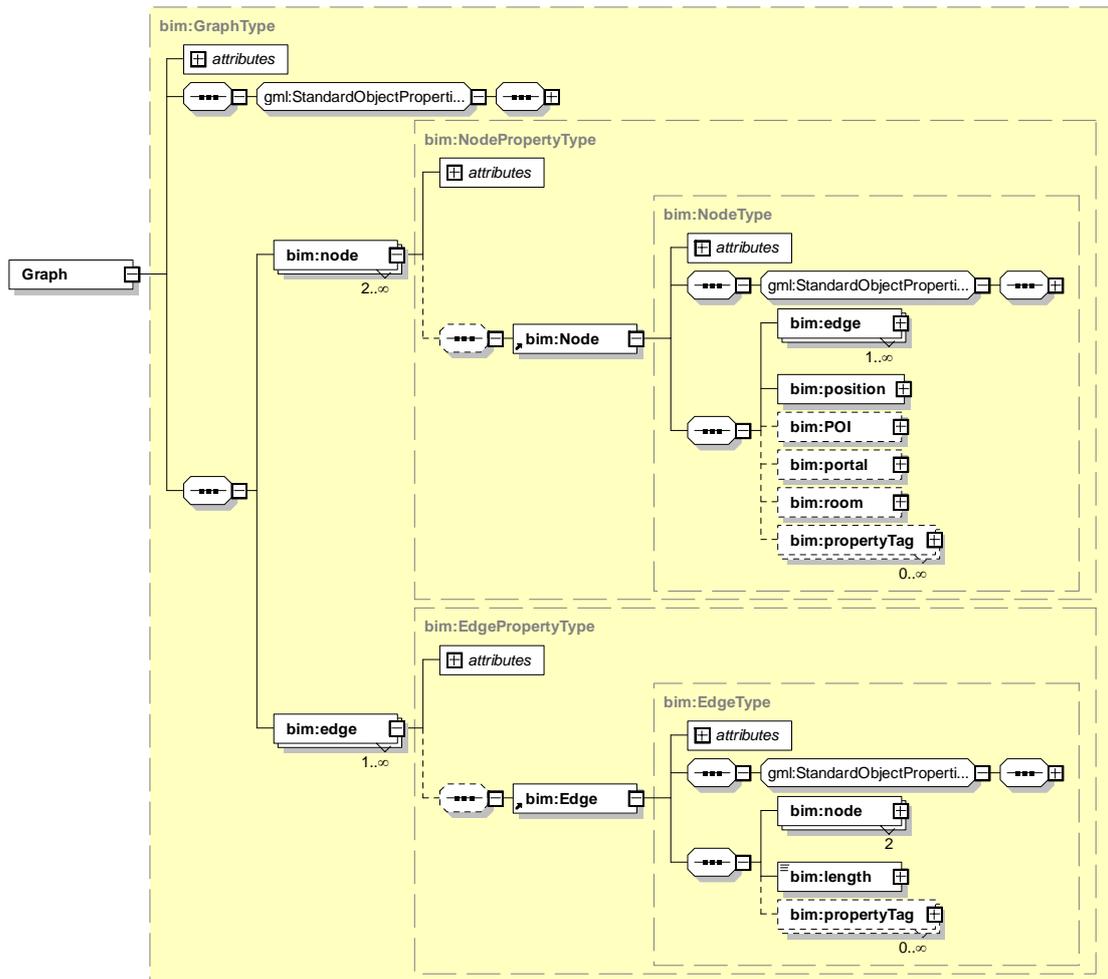


Abbildung 4.11.: Aufbau des Wegenetzes (*Graph*, *Node* und *Edge*).

ten in einem zwei- oder dreidimensionalen Koordinatensystem, so müssen einmalig die Transformationsparameter für die Koordiantentransformation bestimmt werden (vgl. Abschnitt 3.3.1). Da viele Positionierungsverfahren intern bereits auf der Grundlage von zweidimensionalen Kartesischen Koordinatensystemen operieren, gestaltet sich die Bestimmung der Transformationsparameter in diesem Fall noch einfacher, denn es sind nur zwei gemeinsame Stützpunkte erforderlich. Liefert ein Positionierungssystem Positionsdaten in Form von symbolischen Bezeichnern, so lassen sich diese ebenfalls auf das Modell abbilden. Dazu können die Elemente des Modells mittels der *PropertyTags* mit entsprechenden Namen versehen werden.

- *Abbildung von Orten von Interesse*: POIs lassen sich direkt in das Modell aufnehmen und können mit beliebigen zusätzlichen Attributen versehen werden. Sie sind über geometrische sowie symbolische Koordinaten referenzierbar und über dedizierte Knoten im Wegenetz angebunden. In konkaven Räumen kann die Sichtbarkeit eines POI

4. Ein GML-basiertes Umgebungsmodell für I-LBS

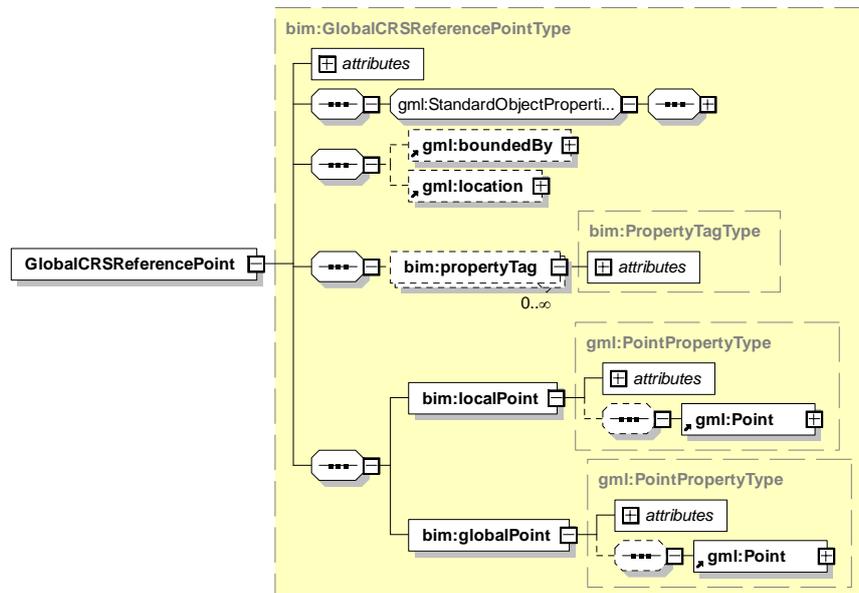


Abbildung 4.12.: Aufbau eines Referenz-Punktes für die Koordinaten-Transformation (*GlobalCRSReferencePoint*).

p ausgehend von einer Position q ebenfalls ermittelt werden: dazu muss überprüft werden, ob die Strecke \overline{pq} das Polygon des umgebenden Raumes schneidet. Dieser Test ist für ein Raumpolygon mit n Kanten in $O(n)$ möglich.

- *Abbildung von Bereichen von Interesse:* eine *Area of Interest* wird als direktes Element im Modell unterstützt und kann sich über mehrere Räume, Stockwerke und Gebäude erstrecken.
- *Durchführung von Bereichsanfragen:* Allen realen Objekten des Modells sind geometrische Koordinaten zugeordnet, sodass sie durch eine Punkt- bzw. Bereichsanfrage gefunden werden können. Aufgrund der hierarchischen Anordnung der realen Objekte können diese Anfragen zudem effizient beantwortet werden. Auch werden Bereichsanfragen in Abhängigkeit der gehbaren Distanz unterstützt. Dazu wird ausgehend vom nächstgelegenen Knoten im Wegenetz eine Breitensuche gestartet, bei der die angefragte Distanz als Abbruchkriterium gilt. Über die Referenzen der gefundenen Knoten können die zugehörigen Räume identifiziert werden.
- *Abbildung der Gebäude-Topologie und Gebäude-Geometrie:* Sowohl die geometrischen als auch die topologischen Eigenschaften der Objekte sind explizit im Modell enthalten. Der Graph des Umgebungsmodells kann unmittelbar zur Suche kürzester Wege herangezogen werden. Alle direkt benachbarten Wegpunkte weisen eine gegenseitige Sichtverbindung auf, sodass die Wegpunkte entlang eines Pfades auch gleichzeitig zur Visualisierung des Pfades geeignet sind.

4. Ein GML-basiertes Umgebungsmodell für I-LBS

- *Bereitstellung von Navigationsdaten:* Die Trennung zwischen Lage und Form der Räume sowie andererseits ihrer Verbundenheit und Erreichbarkeit ermöglicht es, dass das Wegenetz eines Gebäudes auch direkt für die Indoor-Navigation herangezogen werden kann: Indem bei der Erzeugung des Wegenetzes ein geeigneter Algorithmus verwendet wird (z.B. der zuvor beschriebene *WaypointGraph*), entsteht ein Graph, dessen Knoten auch zur Erzeugung von Navigationsanweisungen verwendet werden können. Auf die Erzeugung dieser Anweisungen wird im folgenden Kapitel eingegangen.
- *Bereitstellung von Visualisierungsdaten:* Für die Visualisierung eines Gebäudes sind Lage und Form von Räumen, Stockwerken und Gebäuden in dem Modell enthalten. Über die Stützpunkte einer *Site* ist es möglich, alle Koordinaten der Objekte auch in globale Koordinaten umzuwandeln, sodass diese in anderen Tools z.B. anhand von WGS84-Koordinaten angezeigt werden. Mit Hilfe des XML-basierten Formates lassen sich die Daten auf verschiedenen Plattformen verarbeiten.
- *Austauschbarkeit des Wegenetzes:* In dem vorgestellten Modell ist die genaue Ausprägung des Wegenetzes nicht vorgegeben. Es müssen lediglich Portale im Wegenetz vorhanden sein, die Anzahl und Position restlicher Wegpunkte ist frei definierbar. Dadurch ist es möglich, dass unterschiedliche Wegenetze gespeichert werden. So kann für einen Dienst beispielsweise ein *CornerGraph* gespeichert werden, während für einen anderen Dienst die Wege anhand eines regelmäßigen Gitters aufgebaut werden.

Das vorgestellte Modell erfüllt somit die eingangs formulierten Anforderungen, und es kann auf unterschiedliche Weise genutzt werden. Denn je nachdem, welche Informationen ein Dienst oder eine Anwendung benötigen, sind nicht zwangsläufig alle Elemente des Modells relevant. Für eine einfache Anwendung, in der lediglich eine Karte der Umgebung angezeigt werden soll, müssen eventuelle nur die Räume bekannt sein. In einem anderen Szenario kann auch alleine der *Graph* ausreichend sein, wenn z.B. in einem Dienst nur die Entfernung zwischen POIs bestimmt werden muss. Erst für aufwändigere Dienste, wie beispielsweise die Navigation durch ein Gebäude, werden alle Elemente des hybriden Umgebungsmodells benötigt.

Mit der Umsetzung als GML-Anwendungsschema ist zugleich auch eine gute Erweiterbarkeit des Modells gewährleistet. Denkbar wäre beispielsweise, dass zukünftig auch Informationen über vorhandene Positionierungsverfahren direkt im Umgebungsmodell mit abgespeichert werden. Diese könnten einem Dienst bzw. einer mobilen Anwendung den Zugriff auf verfügbare Positionierungsverfahren dadurch erleichtern, indem beispielsweise Zugangsdaten für ein bestimmtes System oder Namen und Positionen umliegender Basisstationen angegeben werden.

Die gewählte Form der Speicherung in einem XML-Format hat den Vorteil, dass die Daten auf zahlreichen vorhandenen Plattformen direkt verarbeitet werden können. Das XML-Schema (siehe Anhang A) ermöglicht zudem über gängige Tools eine automatische

4. Ein GML-basiertes Umgebungsmodell für I-LBS

Codeerzeugung in einer objektorientierten Programmumgebung. Darüber kann das Modell schnell in neue Programme eingebunden werden.

Kapitel 3 hat die Erzeugung von Umgebungsmodellen und Wegenetzen behandelt und in diesem Kapitel wurde ein einheitliches Format zur Speicherung vorgestellt. Zusammen kann dadurch ein Modell für verschiedenste Arten von I-LBS bereitgestellt werden. Das nachfolgende Kapitel geht näher auf eine bestimmte Art dieser Dienste ein und stellt Ansätze vor, die der Erzeugung von Anweisungen zur Indoor-Navigation dienen.

5. Indoor Navigation

Navigation hat sich mittlerweile zu einem in vielen verschiedenen Bereichen etablierten Begriff entwickelt und umfasst verschiedene Aspekte. Im Umfeld der Indoor Location-based Services zählen dazu:

- die Positionsbestimmung eines Zielobjekts bzw. mehrerer beteiligter Zielobjekte,
- die Ermittlung und Verarbeitung von benötigten *Umgebungsinformationen* und
- die Berechnung und Bereitstellung von *Navigationsdaten*.

Umgebungsinformationen im Sinne der Navigation sind Informationen, die die Umgebung und deren Eigenschaften näher beschreiben und zur Indoor Navigation herangezogen werden können. Neben den bereits in Kapitel 3 beschriebenen Eigenschaften zählen dazu auch weitere Informationen wie beispielsweise Landmarken (markante Punkte in der Umgebung, die eine Orientierung bieten) oder auch benutzerspezifische Zutrittsrechte bzw. organisatorische Eigenschaften der Umgebung. Navigationsdaten sind Informationen, die auf der Grundlage von Positions- und Umgebungsinformationen gewonnen werden und unmittelbar den Prozess der Navigation unterstützen. Beispiele dafür sind Wegpunkte auf einer Karte, textuelle Navigationsanweisungen, Symbole oder auch Geräusche in der Umgebung. Allgemein erfordert die Navigation durch ein Gebäude visuelle, akustische oder andere erfahrbare Hilfsmittel, die den Benutzer zu seinem Ziel leiten. Bei der Entwicklung und Umsetzung von Indoor Navigations-Anwendungen gilt daher es, insbesondere folgende Probleme zu lösen:

- Die bekannten Ansätze und Konzepte aus dem Bereich der Outdoor-Navigation können nicht unmittelbar auf die Indoor Navigation übertragen werden, da sich die Benutzer innerhalb von Gebäuden nicht auf festgelegten Straßen, sondern frei durch Räume, Stockwerke und Hallen bewegen können.
- Die Heterogenität von Wegenetzen führt dazu, dass ein Indoor Navigationssystem auf unterschiedlich komplexen Wegegraphen operieren und trotzdem verständliche Navigationsanweisungen erzeugen muss.
- Endgerät-unterstützte und netzbasierte Positionierungsverfahren bedingen auf technischer Ebene eine enge Kopplung von Positionierung- und Navigationssystem. Dies stellt zum einen eine organisatorische Hürde dar, zum anderen muss ein Navigationssystem entsprechend mit den Positionsdaten aus heterogenen Positionierungsverfahren umgehen können.

5. Indoor Navigation

- Der spontane Abruf von Navigationsdaten durch einen Benutzer in einer fremden Umgebung erfordert geeignete Schnittstellen. Dabei können entweder die benötigten Umgebungsinformationen, oder auch vorberechnete Navigationsdaten bereit gestellt werden. Letzteres hat den Vorteil, dass der Berechnungsaufwand auf mobilen Endgeräten dadurch reduziert werden kann.
- Indoor Navigationssysteme können in sehr dynamischen Umgebungen eingesetzt werden und müssen dann auf Änderungen in den Umgebungsinformationen reagieren können. Dies bedeutet z.B., dass eine Route, die über mehrere Orte von Interesse verläuft, entweder aufgrund von Zustandsänderungen der POIs oder aufgrund von Änderungen im Wegenetz aktualisiert werden kann.

Dieses Kapitel befasst sich vorrangig mit dem Problem der *Berechnung von Navigationsanweisungen*. Der Schwerpunkt liegt dabei auf der Frage, wie die dazu benötigten Navigationsdaten aus heterogenen Wegenetzen erzeugt werden können und auf welche Art und Weise Navigationsanweisungen gegeben werden können. Ausgehend von einem Ansatz zur Einteilung von Anweisungen in verschiedenen Richtungsangaben wird auf verschiedene Probleme und deren Lösungen bei der Verwendung von heterogenen Wegenetzen eingegangen. Anschließend wird diskutiert, wie Landmarken in die Erzeugung von Navigationsdaten und -anweisungen mit einbezogen werden können.

Die Umsetzung dieser Konzepte wird anhand einer prototypischen Implementierung eines personalisierten Navigationssystems am Flughafen München diskutiert. Das vorgestellte System berücksichtigt dabei nicht nur den Verlauf eines Weges und umliegende Landmarken, sondern auch Routen über mehrere Stationen sowie die Gehzeit und zeitliche Einschränkungen bei der Planung der Wege.

Einen Spezialfall stellen Szenarien dar, in denen ein Benutzer in einer fremden Umgebung unter 100%-tiger Wahrung seiner Privatsphäre effizient über sein mobiles Endgerät auf Navigationsdaten zugreifen will. Dazu wird ein Ansatz vorgestellt, bei dem vorberechnete Navigationsdaten in Barcodes hinterlegt werden.

5.1. Verwandte Arbeiten

Im Umfeld der Fußgängernavigation existieren bereits zahlreiche verschiedene Ansätze, um einen Benutzer zu seinem Ziel zu leiten, und mobile Endgeräte spielen dabei eine zunehmend größere Rolle (vgl. Butz et al. [24], Kray et al. [87], Heidmann und Hermann [56], Arikawa et al. [8]). Unabhängig davon, ob Navigationsanweisungen auf einem mobilen Endgerät oder über fest installierte Systeme in der Umgebung ausgegeben werden, unterscheiden sich die existierenden Arbeiten anhand der Art der übermittelten Information, z.B. anhand von Karten, graphischen Symbolen oder textbasierten Darstellungen. Eine aktuelle Übersicht geben auch Huang und Gartner [62].

Während sich frühe Arbeiten vor allem auf die Anzeige der Position auf einer Karte konzentrierten (Abowd et al. [2]), wurde zunehmend auch die Wegewahl berücksichtigt

5. Indoor Navigation

(Wahlster et al. [148]). Der Kontextwechsel beim Betreten eines Gebäudes stellt ein weiteres Problem dar (Baus et al. [14], Rehr et al. [120]), und ein gutes Beispiel dafür ist die multimodale Verknüpfung der Navigation im Auto mit der Fußgängernavigation im Innen- und Außenbereich (Krüger et al. [89]). Innerhalb von Gebäuden können neben den Karten und darauf eingezeichneten Wegen zusätzlich Fotos eingeblendet werden (Kargl et al. [70]), oder es erfolgt die Anzeige einer virtuellen 3D-Umgebung über ein *Head Mounted Display* (Peternier et al. [115]). Darüber hinaus existieren auch z.B. spezielle Ansätze für die Blinden-Navigation (Ran et al. [116]), die einen Benutzer mittels Sprache über in der Umgebung befindliche Objekte und Hindernisse informieren.

Komponenten in den Gebäuden wie beispielsweise Monitore, kleine LCD-Displays, LEDs, etc. können ebenfalls für eine Navigation verwendet werden. Dadurch können den Besuchern z.B. Richtungspfeile auf einem Monitor angezeigt werden (Kray et al. [88]). Rukzio et al. [124] stellen ein Konzept vor, in dem nicht ein statischer Pfeil angezeigt wird, sondern an einer Kreuzung zyklisch alle Richtungen angezeigt werden und der Benutzer über ein Vibrationssignal seines mobilen Endgerätes informiert wird, welcher Anweisung er zu folgen hat. Andere Ansätze verwenden kleine elektronische Türschilder, auf denen personalisierte Richtungsanweisungen angezeigt werden (Kray et al. [86]). Diese Richtungsanweisungen lassen sich auch weiter personalisieren, wenn sich ein Benutzer mit einem Sender (*Tag*) in der Nähe befindet (Lijding et al. [95]).

O'Neil [105] vergleicht in einer Studie die Auswirkungen von graphischer und textueller Beschilderung auf die Geschwindigkeit und die Fehlerrate (Abweichung vom korrekten Weg) bei der Wegefindung. Die graphische Beschilderung führt zwar zu einer höheren Geschwindigkeit, jedoch erweist sich die textuelle Beschilderung als am effektivsten bei der Vermeidung von Fehlern bei der Wegefindung. Chewar und McCrickard [30] untersuchen zusätzlich auch Karten und Kartenausschnitte. Sie kommen zu dem Ergebnis, dass einfache Pfeile (graphische Beschilderung) und eine Liste von Richtungsanweisungen (textuelle Beschilderung) einen Besucher am schnellsten zum Ziel führen. Beide Darstellungen sind den Karten auch in Bezug auf die Fehlerrate überlegen, wobei hier im Gegensatz zu O'Neil die Richtungspfeile geringfügig besser abschneiden. Eine Studie von Hölscher et al. [60] vergleicht Karten und Beschilderung sowie eine Kombination aus beiden. Darin werden unter anderem die für eine Navigationsaufgabe benötigte Zeit, die zurückgelegte Distanz und deren Verhältnis zum kürzest möglichen Weg verglichen. Die Beschilderung zeigt dabei bessere Ergebnisse als die Karte, besonders hinsichtlich der benötigten Zeit. Am besten erweist sich in dieser Studie jedoch die gleichzeitige Verwendung von Karte und Beschilderung: Die insgesamt benötigte Zeit reduziert sich merklich und die zurückgelegte Distanz ist nur noch 36% höher als der kürzest mögliche Weg (nur Karte: 64%, nur Beschilderung: 62%). Dieselben Autoren untersuchen in einer ähnlichen Arbeit auch die Auswirkungen von schematisierten Gebäudeplänen (Meilinger et al. [99]). Eines der Ergebnisse ist, dass schematische Karten die Wegefindung in großen Gebäuden verbessern. Gleichzeitig weisen die Autoren darauf hin, dass an einem Entscheidungspunkt in einer unbekanntenen Umgebung eine eindeutige Richtungsanweisung immer wichtiger ist, als die detaillierte Visualisierung auf einer Karte. In einer anderen Studie untersuchen Lovelace et al. [96], welche Arten von Anweisungen verwendet werden, wenn Probanden eine Wegbeschreibung für einen Weg

5. Indoor Navigation

abgeben sollen, den sie zuvor gelaufen sind. Unter anderem ist dabei ein Ergebnis, dass 40-50% der in den Wegbeschreibungen verwendeten Landmarken nicht unmittelbar an einem Entscheidungspunkt (z.B. einer Abzweigung) liegen.

Auch Videosequenzen oder virtuelle Rundgänge wurden zur Navigation vorgeschlagen: Für den Outdoor-Bereich beschreibt Kolbe [80] einen Ansatz, bei dem zuvor aufgenommene Videos oder Fotos mit zusätzlichen Informationen (z.B. Richtungspfeile oder Straßennummern) angereichert und dem Benutzer auf seinem mobilen Endgerät angezeigt werden (siehe auch Kolbe et al. [81]). Für den Indoor-Bereich stellen Münzer und Stahl [100] einen Ansatz vor, bei dem dem Benutzer an Entscheidungspunkten eine 3D-Darstellung der Umgebung aus einer egozentrischen Perspektive angezeigt wird.

Die mobilen bzw. personalisierten Indoor-Navigationssysteme stellen hohe Anforderungen an die Genauigkeit des Positionierungsverfahrens, vor allem auch an die Messung der aktuellen Blickrichtung. Viele Ansätze bedienen sich daher diskreter Referenzpunkte, an denen die aktuelle Position und Blickrichtung genau bestimmt werden kann. Ein populäres Beispiel dafür sind Barcodes, die an mehreren verschiedenen Orten im Gebäude an Wänden oder Gegenständen angebracht werden. In dem von Mulloni et al. [102] entwickelten System wird über die Kamera eines Smartphones der Barcode erkannt und anschließend die Position des Benutzers auf der Karte angezeigt. Je nachdem, wie viele Freiheitsgrade bei der Berechnung mit einbezogen werden, kann die Position der Kamera (des Benutzers) dabei in 2D oder 3D relativ zur Position des Barcodes ermittelt werden. Chang et al. [27] beschreiben ein Indoor Navigationssystem für Personen mit geistigen Behinderungen und verwenden dazu ebenfalls Barcodes auf Wänden, Türen, etc., anhand derer Fotos mit Richtungs- und Handlungsanweisungen auf einem mobilen Endgerät angezeigt werden können.

5.2. Navigationsanweisungen und Landmarken

Wie bereits im Abschnitt 2.1.5 behandelt, unterscheiden sich Indoor Navigationssysteme vor allem hinsichtlich der Eigenschaften *Mobilität*, *Grad der Personalisierung*, *Granularität der Navigationsanweisungen*, *Positionierungssystem*, *Routenberechnung*, *Dynamisches Verhalten* und *Location Privacy*. Die Granularität der Anweisungen reicht vom Anzeigen eines Ziels auf einer Karte über Wege, die auf der Karte eingezeichnet werden, bis hin zu dedizierten Navigationshinweisen, die den Benutzer unterwegs unterstützen. Letztere Klasse von Anweisungen zählt zu den komplexesten, da dafür sowohl der gesamte Weg, als auch die aktuelle Position und Blickrichtung der Zielperson bekannt sein müssen. Die Herausforderung besteht dabei darin, für eine beliebige Position entlang des Weges zum Ziel eine passende Navigationsanweisung berechnen zu können, die einem an dieser Position befindlichen Benutzer einen verständlichen Hinweis dazu gibt, wie sein nächster Schritt aussieht. Ob die Anweisung in Form einer graphischen Darstellung (beispielsweise durch einen Richtungspfeil) oder mittels eines textbasierten Kommandos erfolgt, ist für die Berechnung zunächst nicht entscheidend. Relevant sind vor allem die Form der aktuellen Umgebung, der weitere Verlauf des Weges und die Anzahl der möglichen Richtungswechsel

5. Indoor Navigation

an diesem Ort. Jede einzelne Anweisung hat immer nur eine bestimmte räumliche Gültigkeit, d.h. nachdem diese verlassen wurde, ist eine neue Anweisung erforderlich, um zum Ziel zu finden. Insgesamt sollte auf einem Weg die Anzahl der benötigten Anweisungen möglichst klein sein, gleichzeitig sollte jedoch auch jede einzelne Anweisung spezifisch genug sein, damit ein Benutzer an einer Kreuzung nicht falsch abbiegt.

Geht man davon aus, dass die Position und Blickrichtung des Benutzers bekannt sind und gleichzeitig auch die geplante Route zum Ziel durch eine Abfolge von Wegpunkten gegeben ist, so gilt es nun, aus diesen Informationen eine geeignete Anweisung zu berechnen. Dabei stellt sich jedoch auch das Problem, dass die Art des Wegenetzes im Allgemeinen nicht im Voraus bekannt ist. Ansonsten könnte ein einfacher Ansatz darin bestehen, für eine gegebene Position den Benutzer einfach in Richtung des nächsten Wegpunktes zu leiten. Da aber ein Wegenetz auch eine sehr hohe Anzahl von Knoten enthalten kann, stellt dies keine zufriedenstellende Lösung dar.

Eine zweite Herausforderung besteht darin, auch Landmarken in die Navigationsanweisungen mit einzubeziehen. Landmarken erleichtern die Orientierung und helfen dem Benutzer, seinen Weg entlang wieder erkennbarer Objekte leichter zu finden. Da jede Landmarke jedoch immer nur von einem begrenzten Bereich aus sichtbar ist, gilt es, für die geplante Route diejenigen Landmarken zu identifizieren, die eine sinnvolle Einbindung erlauben.

Im Folgenden wird zunächst auf das Problem der Erzeugung von Navigationsanweisungen auf der Grundlage von heterogenen Wegenetzen eingegangen. Anschließend wird die Verwendung von Landmarken in den Navigationsanweisungen diskutiert. Die Ansätze wurden in [126] vorgestellt, eine ausführliche Behandlung findet sich in [131].

5.2.1. Erzeugung von Navigationsanweisungen aus heterogenen Wegenetzen

Navigationsanweisungen für den Indoor-Bereich unterscheiden sich grundlegend von den für Straßennetze verwendeten Anweisungen. Innerhalb eines Gebäudes existiert zwar in der Regel eine Hierarchie von Stockwerken, Fluren und Räumen, aber ein Benutzer kann sich sehr frei durch diese hindurch bewegen. Unter der Annahme, dass der gegenwärtige Aufenthaltsort, die Blickrichtung, sowie Richtung und Entfernung zum nächsten Wegpunkt eines Benutzers bekannt sind, können einfache Navigationsanweisungen auf der Grundlage eines *8-Corner-Systems (8CS)* generiert werden. Ein 8CS unterscheidet acht verschiedene Richtungen: geradeaus, schräg rechts, rechts, schräg rechts zurück, zurück, schräg links zurück, links und schräg links. Abbildung 5.1(a) zeigt ein 8CS, in dem jeder Richtung ein Winkel von 45° zugeordnet ist. Die Winkelangaben beziehen sich auf die aktuelle Blickrichtung des Betrachters, wobei ein Winkel von 0° einem Blick nach vorn entspricht.

Die Richtungen können zusammen mit Entfernungsangaben zu einer Navigationsanweisung kombiniert werden, indem für die Abschnitte des Weges eine Abfolge von Richtungsangaben und dazwischen liegenden Entfernungsangaben berechnet wird. Voraussetzung dafür ist wie bereits erwähnt, dass die aktuelle Blickrichtung bekannt ist, so dass der Benutzer die Anweisung relativ zu seiner aktuellen Perspektive ausführen kann. Sei $WG = (V, E)$ ein

5. Indoor Navigation

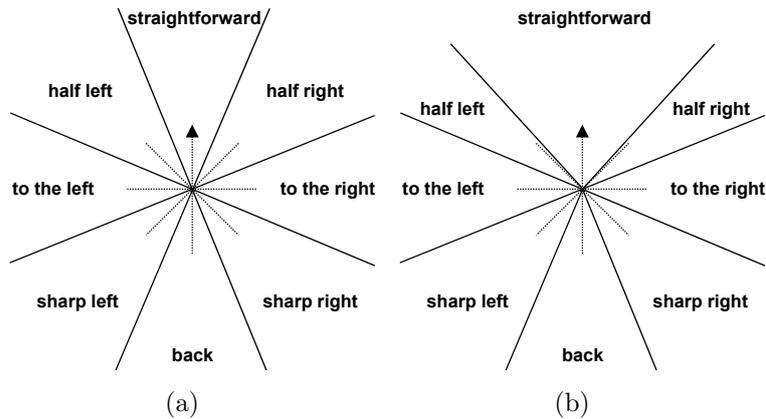


Abbildung 5.1.: (a) Richtungsangaben im 8-Corner-System (8CS) (b) Modifiziertes 8CS mit einem breiteren Winkel für die *geradeaus*-Anweisung.

Wegpunktegraph und sei (s, w_1, \dots, w_n, t) mit $w_i \in V, 1 \leq i \leq n$ eine Liste von Wegpunkten auf einem Pfad von s nach t . An jedem Wegpunkt lässt sich eine Richtungsangabe ermitteln, indem man den Blickwinkel zum nächsten Wegpunkt sowie den Winkel zwischen aktuellem, nächsten und übernächsten Wegpunkt betrachtet. Dazu sei β_i der Winkel $\sphericalangle w_i w_{i+1} w_{i+2}$ und $d(w_i, w_j)$ der Abstand zwischen zwei beliebigen Wegpunkten. Für einen Benutzer am Wegpunkt w_i kann beispielsweise folgende Anweisungen für den Weg zum nächsten Wegpunkt w_{i+1} erzeugt werden: *nach 15 Metern biegen Sie rechts ab* bzw. allgemein: *nach $\langle x \rangle$ Metern \langle Richtung \rangle* . Dabei ist $\langle x \rangle = d(w_i, w_{i+1})$ und \langle Richtung \rangle berechnet sich in Abhängigkeit von β_i wie folgt:

$$\text{Richtung} = \begin{cases} \text{gehen Sie geradeaus,} & \text{falls } 337,5^\circ \leq \beta_i \leq 22,5^\circ \\ \text{gehen Sie schräg rechts,} & \text{falls } 22,5^\circ < \beta_i \leq 67,5^\circ \\ \text{biegen Sie rechts ab,} & \text{falls } 67,5^\circ < \beta_i \leq 112,5^\circ \\ \text{gehen Sie schräg rechts zurück,} & \text{falls } 112,5^\circ < \beta_i < 157,5^\circ \\ \text{gehen Sie zurück,} & \text{falls } 157,5^\circ \leq \beta_i \leq 202,5^\circ \\ \text{gehen Sie schräg links zurück,} & \text{falls } 202,5^\circ < \beta_i \leq 247,5^\circ \\ \text{biegen Sie links ab,} & \text{falls } 247,5^\circ < \beta_i \leq 292,5^\circ \\ \text{gehen Sie schräg links,} & \text{falls } 292,5^\circ < \beta_i < 337,5^\circ \end{cases}$$

Abbildung 5.2 zeigt ein Beispiel für vier aufeinander folgende Navigationsanweisungen auf einem Weg von einem Startpunkt s zu einem Ziel t . Darin werden jedoch nur die einzelnen Wegpunkte und noch nicht die Sichtbarkeit zwischen aufeinander folgenden Wegpunkten berücksichtigt. Bei diesem Ansatz hängt die Qualität der Navigationsanweisungen jedoch unmittelbar von der Lage der einzelnen Wegpunkte und der gewählten Winkel im 8CS ab. Wie Abbildung 5.3 zeigt, können durch ungünstig gelegene Wegpunkte alternierende Navigationsanweisungen entstehen: das Wegenetz enthält mehr Wegpunkte als zuvor, und diese sind in einem Zick-Zack-Muster angeordnet. Eine solche Konstellation kann beispielsweise bei einem Wegenetz auftreten, das durch eine Triangulierung erzeugt wurde.

5. Indoor Navigation

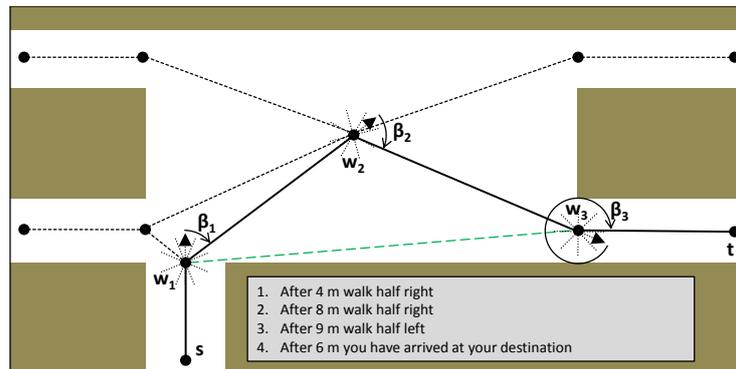


Abbildung 5.2.: Ein Wegpunktgraph und eine Abfolge von Navigationsanweisungen für den Weg von s nach t basierend auf einem 8-Corner-System. In diesem einfachen Beispiel wird nur die Abfolge der Wegpunkte, nicht jedoch deren gegenseitige Sichtbarkeit berücksichtigt.

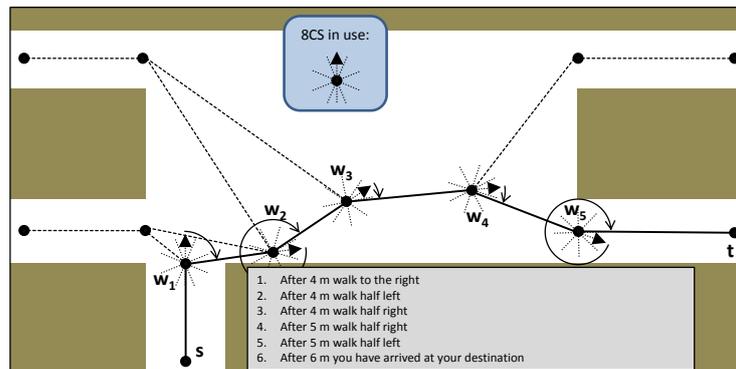


Abbildung 5.3.: Eine ungünstige Lage der Wegpunkte kann alternierende Anweisungen verursachen.

Daran wird deutlich, warum es nicht immer sinnvoll ist, jeden einzelnen Wegpunkt in den Navigationsanweisungen zu berücksichtigen: die Anzahl der Anweisungen erhöht sich, jedoch ohne dass diese notwendig wären. In diesem Fall können die Anweisungen durch zwei Maßnahmen verbessert werden: zum einen kann der Winkel für die *geradeaus*-Anweisung von 45° auf 60° vergrößert werden (siehe auch Abbildung 5.1(b)). Dann würden sich die

5. Indoor Navigation

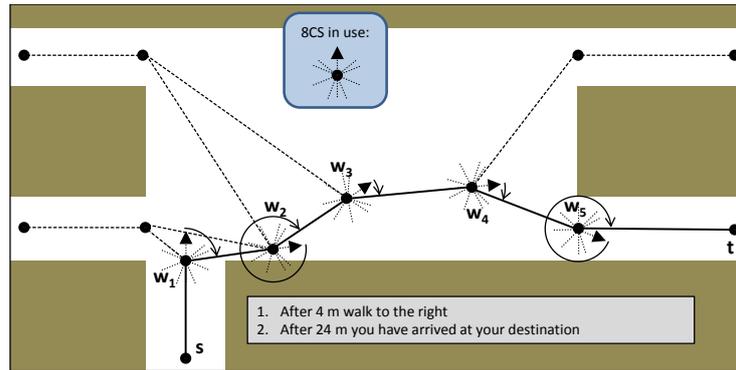


Abbildung 5.4.: Durch die Veränderung der Winkel im 8CS und durch das Zusammenfassen von *geradeaus*-Instruktionen steigt die Verständlichkeit der Navigationsanweisungen.

Navigationsanweisungen wie folgt ändern:

$$\langle \text{Richtung} \rangle = \begin{cases} \text{gehen Sie geradeaus,} & \text{falls } 330^\circ \leq \beta_i \leq 30^\circ \\ \text{gehen Sie schräg rechts,} & \text{falls } 30^\circ < \beta_i \leq 67,5^\circ \\ \text{biegen Sie rechts ab,} & \text{falls } 67,5^\circ < \beta_i \leq 112,5^\circ \\ \text{gehen Sie schräg rechts zurück,} & \text{falls } 112,5^\circ < \beta_i < 157,5^\circ \\ \text{gehen Sie zurück,} & \text{falls } 157,5^\circ \leq \beta_i \leq 202,5^\circ \\ \text{gehen Sie schräg links zurück,} & \text{falls } 202,5^\circ < \beta_i \leq 247,5^\circ \\ \text{biegen Sie links ab,} & \text{falls } 247,5^\circ < \beta_i \leq 292,5^\circ \\ \text{gehen Sie schräg links,} & \text{falls } 292,5^\circ < \beta_i < 330^\circ \end{cases}$$

Zum anderen können aufeinanderfolgende *geradeaus*- bzw. *zurück*-Anweisungen zusammengefasst werden. Dadurch reduziert sich die Anzahl der einzelnen Anweisungen für eine Route, und es kommt zu weniger Richtungswechseln für den Benutzer. Abbildung 5.4 zeigt ein entsprechendes Beispiel, in dem der Winkel für die *geradeaus*-Anweisung vergrößert wurde und sequenzielle *geradeaus*-Anweisungen zusammengefasst werden. Auf diese Art und Weise wirkt sich eine hohe Knotenzahl im Wegenetz nicht mehr unbedingt negativ aus, und die Navigationsanweisungen werden vereinfacht.

Generell handelt es sich bei allen Anpassungen der Winkel des 8CS um einen Kompromiss zwischen einer geringen Anzahl von Navigationsanweisungen und der Genauigkeit jeder einzelnen Anweisung. So kann die Veränderung der Winkel im 8CS auch unerwünschte Effekte erzielen, wie das Beispiel in Abbildung 5.5 zeigt: bei einem großen Winkel für die *geradeaus*-Anweisung wird die generelle Tendenz eines Weges (in diesem Fall um die runde Säule herum) nicht mehr erkannt und stattdessen eine einzige *geradeaus*-Anweisung generiert. Gerade wenn ein Wegenetz in einem Gebäude sehr heterogen aufgebaut ist und keine gitterförmigen bzw. andere regelmäßigen Strukturen aufweist, reicht die alleinige Anpassung der Winkel im 8CS nicht für verwendbare Navigationsanweisungen aus.

Um mit Hilfe des 8CS Navigationsanweisungen auf beliebigen Wegpunktgraphen zu generieren, müssen daher nicht nur die Beziehungen zwischen unmittelbar benachbarten

5. Indoor Navigation

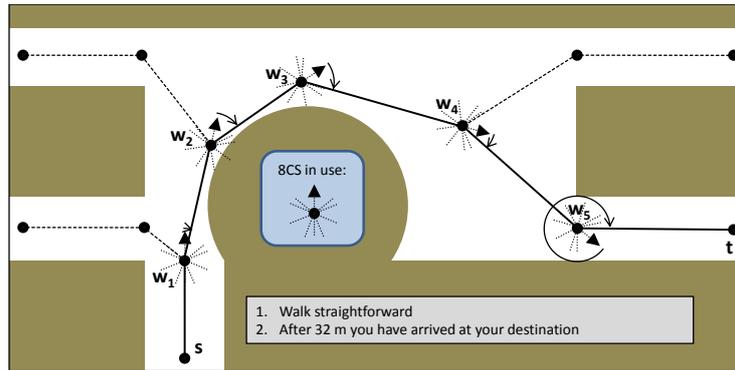


Abbildung 5.5.: Durch die alleinige Veränderung der Winkel im 8CS können nicht alle Konstellationen gelöst werden und es kann weiterhin zu falschen Anweisungen kommen.

Knoten, sondern auch zwischen auf dem Graphen weiter voneinander entfernten Knoten betrachtet werden. Diese Relationen werden in einem so genannten *Overlay-Graphen* gespeichert und dann bei der Erzeugung von Navigationsanweisungen herangezogen.

Ein *Wegpunkte-Overlay-Graph* $WOG = (V', E' | V' \subseteq V)$ enthält

- die Menge CP derjenigen Wegpunkte aus V , die eine Verbindung zwischen zwei Räumen darstellen (*Verbindungspunkte, connection points*). Dies entspricht den in Abschnitt 4.3 vorgestellten Portalknoten.
- die Menge BP derjenigen Wegpunkte aus V , die auf einem kürzesten Pfad zwischen zwei Verbindungspunkten liegen und an deren Position ein *genereller Richtungswechsel* des Pfades vorliegt (*Schaltpunkte, break points*). Solche *break points* können demzufolge nur innerhalb eines konkaven Raumes auftreten.

Die *Schaltpunkte* kennzeichnen diejenigen Wegpunkte, die in einer Sequenz von Navigationsanweisungen nicht mit anderen zusammengefasst werden dürfen, da der Weg an dieser Stelle eine signifikante Richtungsänderung erfährt. Anschaulich betrachtet handelt es sich um diejenigen Punkte auf einem Pfad, die vom Vorgängerknoten aus nicht mehr sichtbar sind. Die *Schaltpunkte* können folgendermaßen ermittelt werden: innerhalb eines Raumes wird zwischen jedem Paar (cp_i, cp_j) von Verbindungspunkten der kürzeste Pfad $(p_1, \dots, p_m), p_1 = cp_i, p_m = cp_j$ berechnet. Ausgehend von p_1 wird nun entlang dem Pfad der erste Punkt $p_k, 1 \leq k \leq m$, gesucht, der keine direkte Sichtverbindung mehr zu p_1 aufweist, d.h. dass die Strecke $\overline{p_1 p_k}$ das Raumpolygon schneidet. In diesem Fall ist p_{k-1} ein *break point* und wird zur Menge BP hinzugefügt. Anschließend werden die verbleibenden Punkte (p_k, \dots, p_m) in Bezug auf p_{k-1} untersucht und wiederum der nächste nicht mehr von p_k aus sichtbare Punkt zur Menge BP hinzugefügt.

Für einen gegebenen Wegpunktgraphen WG , den dazugehörigen Wegpunkte-Overlay-Graphen WOG und einen Pfad auf WG von s nach t können die Navigationsanweisungen nun anhand der Verbindungspunkte und Schaltpunkte wie folgt verbessert werden:

5. Indoor Navigation

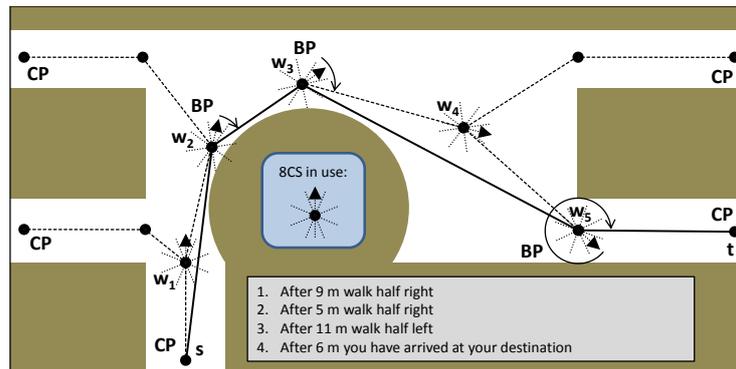


Abbildung 5.6.: Verbindungspunkte (connection points, CP) und Schaltpunkte (break points, BP) zur Vereinfachung von Navigationsanweisungen.

- Aufeinander folgende *geradeaus-* bzw. *zurück-*Anweisungen werden mit der vorangehenden Anweisung vereint, wenn der aktuelle Wegpunkt nicht in CP oder BP enthalten ist. Die Vereinigung von zwei Anweisungen führt dazu, dass der entsprechende Winkel und die Entfernung zwischen Anfang und Ende des betroffenen Abschnitts neu berechnet werden müssen, um daraus eine einzelne neue Anweisung zu generieren.
- Wurden zwei Anweisungen vereint, so muss anschließend überprüft werden, ob sich die resultierende Anweisung nochmals mit ihrem Vorgänger oder Nachfolger vereinen lässt.

Abbildung 5.6 zeigt ein Beispiel für die Verwendung der Schaltpunkte: der Wegpunkt w_3 ist nicht mehr direkt von s aus sichtbar und deshalb w_2 ein *Schaltpunkt*. Analog sind auch w_3 und w_5 *Schaltpunkte*. Die Anzahl der resultierenden Anweisungen reduziert sich in dem Beispiel auf vier, und die Richtungswechsel entlang des Pfades werden besser abgebildet.

Mittels des vorgestellten Ansatzes können nicht nur textuelle Navigationsanweisungen erzeugt werden, es lassen sich damit auch graphische Navigationsanweisungen generieren. Ein einfacher Richtungspfeil kann direkt anhand der Richtungsangabe angezeigt werden. In Verbindung mit der Entfernungsangabe ist es zusätzlich auch möglich, die Länge des Pfeils zu variieren und dadurch dem Benutzer einen besseren Eindruck von der Entfernung bis zum nächsten Richtungswechsel zu geben.

5.2.2. Einbeziehung von Landmarken

Eine weitere Möglichkeit zur Verbesserung von Navigationsanweisungen ist die Einbeziehung von Landmarken. Wie bereits in den Abschnitten 2.1.5 und 5.1 erwähnt, stellen diese eine zusätzliche Hilfe für den Benutzer bei der Navigation dar. Landmarken sind sichtbare und leicht zu erkennende Objekte in der Umgebung, deren absolute oder relative Lage in die Navigationsanweisungen mit einbezogen werden kann. Jedoch sind nicht alle Arten von Objekten gleich gut geeignet. So zeigen beispielsweise Ruddle et al. [123], dass abstrakte

5. Indoor Navigation

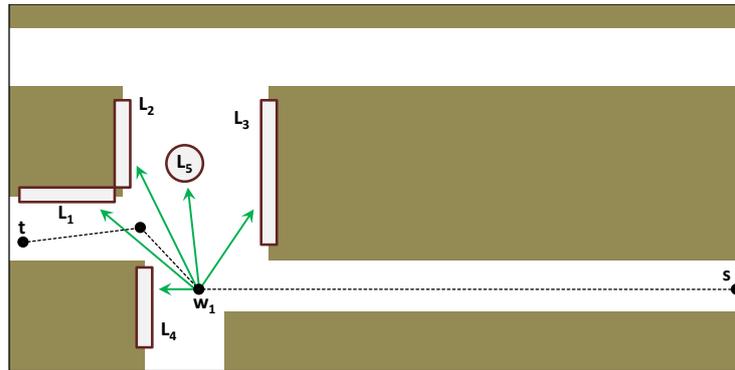


Abbildung 5.7.: Landmarken zur Verbesserung von Navigationsanweisungen.

Farbmuster nicht unbedingt zu einer höheren Genauigkeit bei der Wegfindung führen; die Verwendung von bekannten Objekten als Landmarken kann hingegen die Genauigkeit der Navigation verbessern. Mit Hilfe von Landmarken kann dann z.B. anstatt einer einfachen Anweisung „Gehen Sie geradeaus und biegen sie nach 30m rechts ab“ eine besser verständliche Anweisung „Gehen Sie geradeaus und biegen Sie vor der großen Wendeltreppe rechts ab“ gegeben werden. Unter der Annahme, dass eine Navigationsanweisung immer für einen spezifischen Wegpunkt erzeugt wird, können dementsprechend auch immer alle Landmarken bestimmt werden, die von der jeweiligen Position aus sichtbar sind.

In vielen Indoor Navigationssystemen wird die Anzahl der Änderungen an Wegpunkten und Landmarken deutlich niedriger sein als die Anzahl der Berechnungsanfragen. In diesem Fall empfiehlt es sich, die von einem Wegpunkt aus sichtbaren Landmarken in einem *Sichtbarkeitsgraphen (visibility graph) VG* zu speichern. *VG* ordnet jedem Wegpunkt eine Menge von Landmarken zu, die von diesem Wegpunkt aus direkt sichtbar sind.

Für einen Wegpunkt w_i muss demnach zunächst eine geeignete Landmarke ausgewählt werden. Die Wahl der passenden Landmarke hängt dabei davon ab, wie der weitere Weg verläuft und wie die Landmarke innerhalb der Navigationsanweisung verwendet werden soll. Abbildung 5.7 zeigt ein Beispiel mit fünf verschiedenen Landmarken L_1, \dots, L_5 , wobei L_1 bis L_4 Objekte entlang einer Wand darstellen (z.B. Ladengeschäfte) und L_5 repräsentiert eine große Säule. Ein Benutzer, der von s aus kommend am Wegpunkt w_1 steht, kann von diesem Punkt aus alle fünf Landmarken sehen. Muss er auf seinem weiteren Weg nach links abbiegen (in der Abbildung nach unten), so könnte L_4 in einer Navigationsanweisung verwendet werden. Verläuft der weitere Weg jedoch schräg rechts in Richtung t , so muss eine geeignete Landmarke ausgewählt werden. Diese Auswahl richtet sich wiederum nach dem Typ der gewünschten Navigationsanweisung (*type of instruction, TOI*), d.h. wie genau in der Anweisung auf die jeweilige Landmarke Bezug genommen wird. Beispiele für unterschiedliche TOIs sind:

- Nach 30 Metern gehen Sie schräg rechts *in Richtung* Rolltreppe.
- Nach 10 Metern biegen Sie *vor* der Cafeteria links ab.

5. Indoor Navigation

Typ	Beschreibung	Schlüsselwörter
TOI_{near}	Nahe am Weg gelegene Landmarke, Wegpunkt liegt in unmittelbarer Nähe	vor, hinter, bei, am
TOI_{far}	Entfernte Landmarke, über den Verlauf des Pfadabschnittes sichtbar	in Richtung, entlang, bis zu
$TOI_{vertical}$	Vertikale Landmarken	nach oben, nach unten, in den x-ten Stock

Tabelle 5.1.: Verschiedene Typen von Navigationsanweisungen.

- Biegen Sie *hinter* dem Informationsstand rechts ab.
- Gehen Sie *bei* der großen Säule geradeaus.

Es existieren viele weitere Möglichkeiten, sie unterscheiden sich jedoch alle in erster Linie darin, ob eine Landmarke entweder als Orientierung auf dem nächsten Abschnitt des Pfades verwendet wird („in Richtung“), oder ob sie lediglich als Findungshilfe für den Ort der nächsten Richtungsänderung dient („vor/nach/bei“).

Nachfolgend werden daher, wie in Tabelle 5.1 dargestellt, die TOIs anhand von entfernten, nahe gelegenen sowie vertikalen Landmarken unterschieden. Um für jeden Wegpunkt w_i die dort geeigneten TOIs zu bestimmen, werden die von diesem Wegpunkt aus sichtbaren Landmarken L_{w_i} anhand ihrer Entfernung und anhand ihrer Lage in Bezug auf w_i klassifiziert. Die nahe gelegenen Landmarken (TOI_{near}) befinden sich in unmittelbarer Nähe eines Wegpunktes, so dass anhand von Schlüsselwörtern wie „vor“, „hinter“ oder „bei“ darauf Bezug genommen werden kann. Eine solche Landmarke lässt sich unmittelbar in die oben vorgestellten Anweisungen einfügen, wie folgende Beispiele zeigen: „Nach 30 Metern biegen Sie *vor dem Aufzug* links ab“ oder „Nach 20 Metern gehen Sie *hinter der Rolltreppe* schräg links“. Eine entfernte Landmarke ist aus einer größeren Entfernung vom Wegpunkt aus sichtbar und eignet sich, um dem Benutzer eine grundsätzliche Orientierungshilfe zu geben. Beispiele dafür sind: „Nach 50 Metern biegen Sie rechts ab, dann gehen Sie *in Richtung Mensa*“ oder „Gehen Sie *bis zur Treppe*, dann biegen Sie rechts ab“.

Um nun Landmarken in einer Navigationsanweisung zu verwenden, muss zunächst für den jeweiligen Wegpunkt anhand des Sichtbarkeitsgraphen überprüft werden, ob eine entsprechende Landmarke sichtbar ist. Handelt es sich um eine nahe gelegene Landmarke, so wird als nächstes bestimmt, ob sie sich aus Sicht des herannahenden Benutzers vor, hinter, oder neben dem Wegpunkt befindet. Ebenso muss für eine entfernte Landmarke überprüft werden, ob sie sich im Blickwinkel des herannahenden Benutzers befindet, d.h. ob sie in der konkreten Konstellation überhaupt verwendet werden kann. Liegt eine entfernte Landmarke hinter dem Wegpunkt, so kann eine Anweisung z.B. mit dem Hinweis *in Richtung* versehen werden, liegt sie vor dem Wegpunkt so kann das Schlüsselwort *entlang* eingesetzt werden. Sind von einem Wegpunkt aus mehrere Landmarken sichtbar, so sollte diejenige verwendet werden, die dem Wegpunkt am nächsten liegt. Allerdings ist es auch sinnvoll, wenn bei mehreren aufeinander folgenden Navigationsanweisungen wech-

5. Indoor Navigation

selnde TOSs eingesetzt werden, damit sich ein Benutzer den Weg besser einprägen kann. Eine weitere Möglichkeit besteht darin, eine Landmarke anhand ihrer äußerlichen Merkmale auszuwählen: Raubal und Winter [118] schlagen für Landmarken im Außenbereich eine Metrik vor, die die Auffälligkeit einer Landmarke anhand von z.B. Fläche, Form, Farbe oder auch kultureller Wichtigkeit angibt. Die Autoren verweisen dabei aber auch darauf, dass es sich bei der Auffälligkeit um eine lokale und relative Eigenschaft handelt, d.h. dass die Auffälligkeit immer auch abhängig von anderen in der Nähe befindlichen Landmarken ist.

Der vorgestellte Ansatz ermöglicht es, dass die Berechnung der Navigationsanweisungen unabhängig von der konkreten Art des Wegenetzes und unabhängig von der genauen Lage der Landmarken durchgeführt werden kann. Solange ein Wegenetz die Koordinaten der einzelnen Punkte und den Umriss von Räumen enthält, können alle Schaltungspunkte und Vereinfachungen der Anweisungen bestimmt werden. Die Ausgabe der erzeugten Navigationsanweisungen ist nicht an eine bestimmte Art von Gerät gebunden, d.h. es werden sowohl mobile als auch stationäre Systeme unterstützt. Im Falle eines mobilen Navigationssystems muss jedoch insbesondere die aktuelle Blickrichtung zuverlässig ermittelt werden können, um genaue Anweisungen geben zu können. Für ein stationäres Anzeigesystem gestaltet sich dies wesentlich einfacher, da die Ausrichtung eines Monitors bzw. einer Anzeige in der Regel genau bekannt ist und sich nicht ändert. Stationäre Anzeigesysteme haben den zusätzlichen Vorteil, dass die Anweisungen immer nur für diskrete Positionen, d.h. für den jeweiligen Standort der Anzeige, ermittelt werden müssen. Dies lässt insbesondere zusätzliche Optimierungen bei der Wahl der Landmarken zu.

Die Umsetzung des vorgestellten Ansatzes wird im folgenden Abschnitt anhand eines Navigationssystems für Flughäfen diskutiert.

5.3. Indoor Navigation auf verteilten Anzeigesystemen

Ein Flughafen stellt mit weitläufigen Bereichen, großen Hallen, vielen Räumen und langen Wegen eine herausfordernde Umgebung für die Indoor Navigation dar. Viele Besucher befinden sich an einem Flughafen in der Situation, dass ihnen die Umgebung gänzlich unbekannt ist und dass sie im Vorhinein weder den Weg zum Ziel, noch die voraussichtliche Gehzeit bis dorthin abschätzen können. Hinzu kommt, dass Passagiere eine partiell geordnete Abfolge von Stationen durchlaufen müssen, um zu ihrem Gate zu gelangen, den Anschluss-Flug zu finden oder um nach der Landung den Flughafen wieder zu verlassen. Gleichzeitig kann jeder Passagier zusätzliche Anforderungen aufweisen, wenn er auf dem Weg zum Ziel beispielsweise noch ein bestimmtes Produkt kaufen oder ein Restaurant aufsuchen möchte. Für die Wegewahl an einem Flughafen ergibt sich daraus, dass eine Route nicht nur auf der Grundlage des Start- und eines Zielpunktes erfolgt, sondern dass mehrere Zwischen-Ziele berücksichtigt werden müssen. Manche dieser Ziele sind zusätzlich temporal geordnet: beispielsweise muss der Check-In spätestens x Minuten vor der Boarding-Zeit erfolgen, die Sicherheitsüberprüfung sollte so durchlaufen werden, dass noch ausreichend

5. Indoor Navigation

Gehzeit y zum Gate übrig bleibt, oder das Gate sollte z Minuten vor der Boarding-Zeit erreicht werden. Andere Ziele wie z.B. Restaurants oder Geschäfte liegen außerhalb dieser Ordnung und können an einer beliebigen Position in der Reihenfolge der Ziele besucht werden, vorausgesetzt, es befindet sich in dem jeweiligen Bereich ein POI der gewünschten Klasse.

Im Folgenden wird dazu ein neuer Ansatz für die Indoor Navigation diskutiert, der im Rahmen einer engen Kooperation mit dem Flughafen München prototypisch umgesetzt wurde. Der Ansatz wurde in [126] vorgestellt, eine ausführliche Diskussion der Navigationsanweisungen und der Routenberechnung findet sich in [131] und [65].

5.3.1. Verteilte Anzeigesysteme

An einem Flughafen existieren heutzutage eine Vielzahl von Anzeigen und Monitoren, die für verschiedene Zwecke wie beispielsweise Informationen über abfliegende und ankommende Flüge, Werbung oder aktuelle Statusinformationen verwendet werden. Zusammengekommen ergibt sich an einem größeren Flughafen eine Menge von mehreren tausend Anzeigen, die in der Regel vernetzt sind und damit entfernt gesteuert werden können. Bislang sind diese Systeme nicht in der Lage, auch personalisierte Informationen anzuzeigen. Versieht man sie jedoch mit einer Komponente, anhand derer ein bestimmter Benutzer erkannt bzw. wiedererkannt werden kann, so lassen sich solche verteilten Anzeigesysteme auch zur Anzeige von personalisierten Navigationsanweisungen verwenden. An dieser Stelle wird bewusst der Begriff *Benutzer* und nicht *Person* verwendet, da es nicht Aufgabe des Navigationssystems ist, die personenbezogenen Daten eines Benutzers zu kennen. Für die Navigation ist es lediglich erforderlich, dass eine bestimmte *Sitzung* zugeordnet werden kann. Diese enthält einen eindeutigen Identifikator, das gewünschte Ziel bzw. die Ziele und möglicherweise weitere Anforderungen für die gewünschte Route. Weitere Informationen zur Identität des Benutzers können zwar im Rahmen zusätzlicher Dienste benötigt werden, für die Navigation sind sie jedoch nicht erforderlich.

Zur Identifikation eines Benutzers, der vor einer Anzeige steht, sind verschiedene Ansätze denkbar: die Eingabe eines Codes durch den Benutzer, die Interaktion mit dem mobilen Endgerät des Benutzers (z.B. durch Scannen eines auf dem Display angezeigten Barcodes), eine biometrische Erkennung oder die Identifikation anhand eines Kennzeichens (*Tag*), das der Benutzer mit sich trägt. Letzteres kann z.B. durch einen RFID- oder NFC-Tag erreicht werden und hat den großen Vorteil, dass diese Tags klein und portabel sind, kein großes Investitionsrisiko darstellen und anonym ausgegeben werden können.

Ein Benutzer, der sich mittels eines verteilten Anzeigesystems navigieren lassen möchte, benötigt eine geeignete Schnittstelle zur Formulierung seiner Anforderungen. Auch hier bieten sich die verschiedensten Möglichkeiten an: von Tastatur-, Sprach- und Gestensteuerung über das Scannen von Flugtickets bis hin zum Abruf von Reisedaten und -profilen, die in externen Systemen hinterlegt sind. Eine sehr effektive Möglichkeit ist jedoch auch die persönliche Betreuung durch einen Mitarbeiter, der vor Ort oder per Video-Konferenz die Anforderungen des Benutzers aufnehmen kann. In dem anvisierten Szenario der Flughafen-Navigation hat die persönliche Betreuung über eine Video-Konferenz den großen Vorteil,

5. Indoor Navigation

dass dadurch auf dem großen Areal eines Flughafens auch an weit abgelegenen Positionen wie beispielsweise in einem Parkhaus individuell auf die Bedürfnisse eines Benutzers eingegangen werden kann.

Sind die Ziele des Benutzers dem System bekannt und kann anhand eines Wegenetzes eine entsprechende Route ermittelt werden, so müssen im letzten Schritt die Navigationsanweisungen berechnet werden. Die aktuelle Blickrichtung des Benutzers kann bei einem stationären Anzeigesystem einfach anhand der Ausrichtung der Anzeige ermittelt werden. Landmarken können ebenfalls einbezogen werden, da die aktuelle Position der Anzeige und damit auch des Benutzers bekannt ist. Der Grad an dargestellten Details für die Navigation hängt wiederum von den Präferenzen des Benutzers ab. An manchen Entscheidungspunkten entlang eines Weges ist möglicherweise ein einfacher Pfeil oder eine einfache Anweisung (z.B. *gehen Sie links*) ausreichend. An anderen Stellen möchte sich der Benutzer unter Umständen einen besseren Überblick verschaffen und benötigt dazu eine Karte oder eine Sequenz von Instruktionen, die ihn zu seinem nächsten Ziel leiten. Die Unterscheidung zwischen einfachen und komplexen Navigationsanweisungen kann auch durch den Typ der Anzeige bestimmt sein, da nicht notwendigerweise überall in einem Flughafen die gleichen Arten von Anzeigen vorhanden sind und nicht jeder Typ z.B. eine Kartendarstellung zulässt.

Ein Indoor Navigationssystem hilft dem Benutzer nicht nur, *wie* er zu seinem Ziel gelangt. Auch das *wann* ist eine essentielle Information, versetzt sie den Benutzer doch in die Lage, seine verbleibende Zeit besser einzuschätzen und zu nutzen. Die temporalen Eigenschaften der Route gründen sich zum einen auf der antizipierten Geschwindigkeit des Benutzers, zum anderen aber auch auf Wartezeiten, die an bestimmten Wegpunkten berücksichtigt werden müssen. Ein Beispiel dafür die die Wartezeit bei der Sicherheitsüberprüfung.

Das entwickelte System verwendet für die Navigation zum einen bereits am Flughafen vorhandene Monitore, zum anderen auch zusätzliche Terminals, die mit Videokameras, Mikrofon, Lautsprecher, Touchscreen, Drucker und einer Komponente zur Ausgabe von RFID-Karten ausgestattet sind. An diesen Terminals kann ein Benutzer per Videokonferenz einem Mitarbeiter seine Anforderungen angeben. Das Videobild des Mitarbeiters wird dem Benutzer dabei auf einem Monitor annähernd in Lebensgröße dargestellt (siehe Abbildung 5.8(a)). Nachdem der Mitarbeiter alle Fragen beantwortet und die gewünschten Ziele aufgenommen hat, wird dem Benutzer über ein Ausgabegerät eine RFID-Karte ausgehändigt. Der Mitarbeiter wird im Hintergrund durch ein Auswahlssystem unterstützt, das verfügbare Orte von Interesse anzeigt und über das alle Ziele eingegeben werden können. Anhand der ID der RFID-Karte wird im System anschließend eine Navigationssitzung gespeichert.

Nachdem der Benutzer seine RFID-Karte erhalten hat, kann er diese von nun an an den Kartenleser eines Terminals oder eines Monitors halten und bekommt seine personalisierten Navigationsanweisungen angezeigt. Jedes Mal, wenn es zu einer solchen Interaktion kommt, wird im Hintergrund eine neue Route entsprechend der ID der Karte und des aktuellen Standorts des Terminals bzw. des Monitors berechnet und daraus Navigationsanweisungen ermittelt. Abbildung 5.8(b) zeigt die Darstellung von ausführlichen Navigationsanweisun-

5. Indoor Navigation

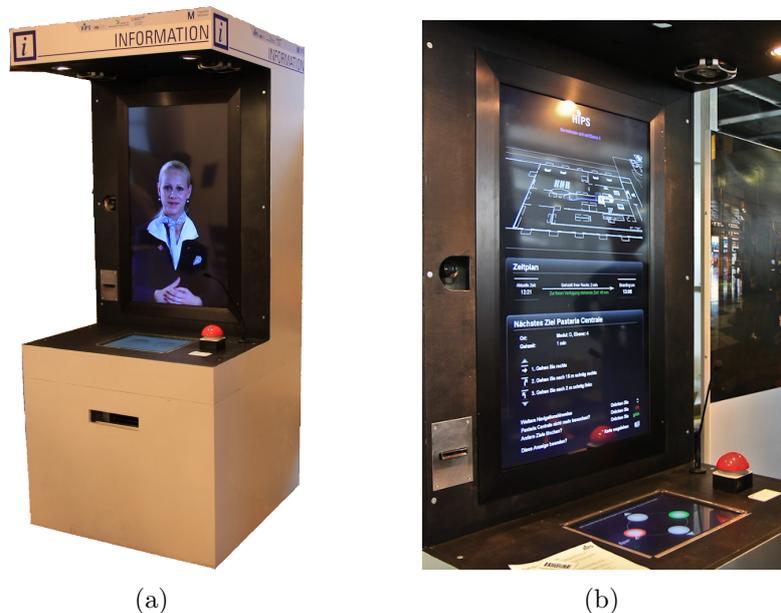


Abbildung 5.8.: Prototyp des entwickelten Systems: (a) Terminal zur Videokommunikation und Ausgabe von RFID-Karten. (b) Anzeige von erweiterten Navigationsanweisungen.

gen auf einem Terminal, Abbildung 5.9 zeigt einen Monitor für Flugzeitinformationen sowie eine einfache Navigationsanweisung, die nach Auflegen der RFID-Karte erscheint. Auf die genaue Darstellung der Navigationsanweisungen wird weiter unten eingegangen.

Da ein Benutzer immer nur dann lokalisiert werden kann, wenn er seine Karte auf eines der Lesegeräte legt, ist es nicht bei allen Zwischen-Zielen möglich festzustellen, ob der Benutzer diese schon besucht hat. Aus diesem Grund ist jedes Lesegerät mit vier Buttons ausgestattet, über die der Benutzer bereits besuchte Ziele abhaken kann, damit werden sie aus der Routenberechnung entfernt. Manche Ziele können automatisch abgehakt werden: beispielsweise ist nach Passieren der Sicherheitsüberprüfung der Check-In bereits erfolgt, auch der Besuch eines außerhalb des Sicherheitsbereiches befindlichen POIs ist damit ausgeschlossen. Falls sich im letzteren Fall jedoch ein äquivalenter POI (z.B. ein äquivalentes Restaurant) innerhalb des Sicherheitsbereiches befindet und noch ausreichend Zeit vorhanden ist, so kann dieses in die Routenberechnung mit aufgenommen werden.

Alle Routen und Navigationsanweisungen werden durch einen zentralen Server berechnet. Dieser verwaltet das Wegenetz und kann gleichzeitig auch auf dynamische Änderungen reagieren, wie z.B. den Wechsel eines Gates oder die Änderung der Wartezeit an einem Knoten. Die einzelnen Terminals und Monitore sind im Wegenetz durch Knoten angebunden (vgl. das Konzept des POI in Abschnitt 4.3). Da beim Auflegen der RFID-Karte des Benutzers jedes Mal eine neue Routenberechnung angestoßen wird, kommen die Terminals und Monitore ohne jegliches Zustandswissen aus. Lediglich statische Daten, die immer wieder zur Visualisierung benötigt werden, sind in den Anzeigen hinterlegt.

5. Indoor Navigation

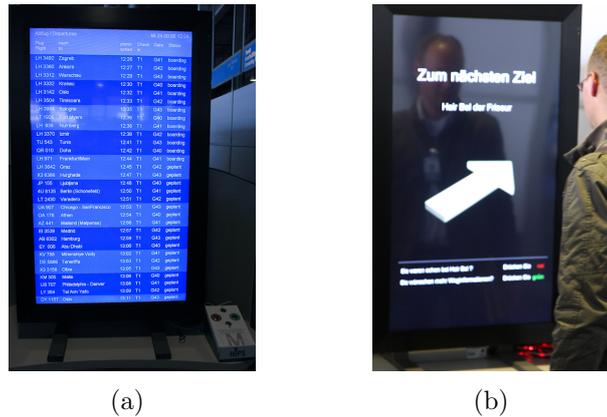


Abbildung 5.9.: Personalisierte Navigation auf öffentlichem Display: (a) Display zur Anzeige von Flugzeiten. (b) personalisierte einfache Navigationsanweisung.

Sowohl auf den Terminals als auch auf den Monitoren existieren zwei unterschiedliche Ansichten für die Navigation. Die erste Ansicht (siehe Abbildung 5.10(a)) umfasst den Namen des nächsten Ziels sowie einen einfachen Pfeil, der dem Benutzer anhand der acht verschiedenen Richtungen in einem 8-Corner-System die nächste Richtung angibt. Diese Ansicht wird jedes Mal angezeigt, wenn der Benutzer seine RFID-Karte auflegt und ermöglicht eine schnelle Orientierung. Zusätzlich kann aus dieser Ansicht heraus auch ein bereits besuchtes Ziel abgehakt werden. Benötigt der Benutzer genauere Anweisungen, so kann er die zweite Ansicht durch Drücken einer Taste bzw. auf dem Touchscreen auswählen. Die zweite Ansicht (siehe Abbildung 5.10(b)) gliedert sich in drei verschiedene Bereiche:

- im oberen Bereich der Anzeige erscheint eine Karte, auf der der aktuelle Standort, die geplanten Ziele und die berechnete Route angezeigt werden. Die Darstellung ist animiert und durchläuft in einer Schleife die verschiedenen Wegpunkte bis zum letzten Ziel. Ein Wechsel des Stockwerks wird durch eine nach oben bzw. nach unten verlaufende Animation der Karte sowie durch entsprechende Symbole angedeutet.
- im mittleren Teil der Anzeige werden zeitabhängige Informationen angezeigt. Dazu zählen die aktuelle Zeit, der Zeitpunkt, an dem das letzte Ziel erreicht werden muss, die Gehzeit der gesamten geplanten Route, sowie die resultierende noch zur freien Verfügung stehende Zeit. Zur Berechnung der Gehzeit werden nicht nur die Geschwindigkeit des Benutzers, sondern beispielsweise auch die Verfügbarkeit und Geschwindigkeit von Laufbändern einbezogen. Anhand dieser Angaben kann der Benutzer unmittelbar erkennen, wie viel freie Zeit ihm noch zur Verfügung steht. Falls die freie Zeit knapp wird, so bekommt der Benutzer eine Meldung angezeigt und wird aufgefordert, sich zu beeilen.
- der dritte Teil der Anzeige besteht aus einer Abfolge von textbasierten Navigationsanweisungen. Diese enthalten sowohl Richtungs- und Entfernungsangaben, als auch Landmarken, und werden wie in den Abschnitten 5.2.1 und 5.2.2 beschrieben erzeugt.

5. Indoor Navigation

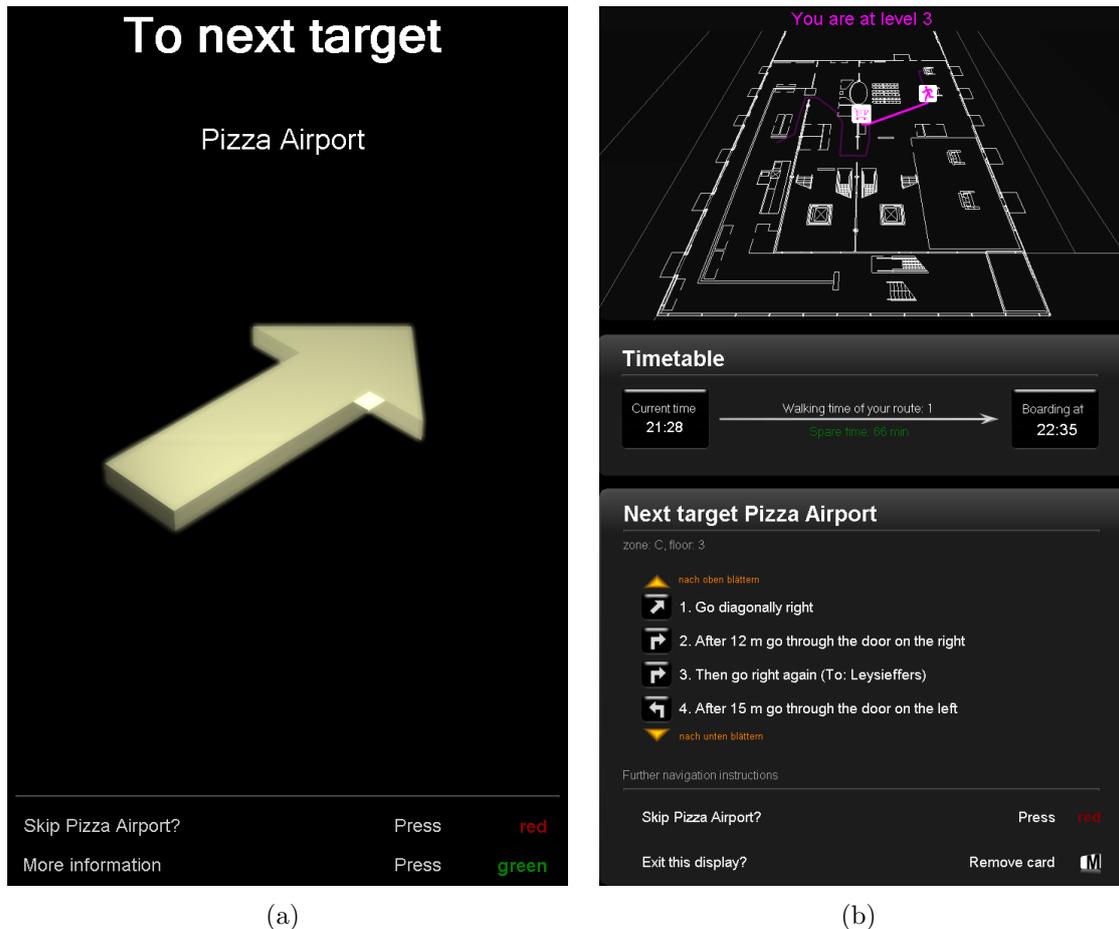


Abbildung 5.10.: Screenshot des Navigationssystems: (a) einfache Ansicht zeigt den Namen des Ziels und die Richtung für den nächsten Abschnitt des Weges. (b) Detaillierte Ansicht mit Karte, Zeitplanung und textbasierten Navigationsanweisungen.

Es werden bis zu vier zusammenhängende Anweisungs-Texte gleichzeitig angezeigt. Über entsprechende Buttons kann der Benutzer durch weitere Anweisungen blättern. Links neben den Anweisungen werden zusätzlich Icons eingeblendet, die die jeweilige Anweisung nochmals graphisch verdeutlichen.

Eine Navigationssitzung läuft ab, wenn entweder der Benutzer sein letztes Ziel abgehakt hat, oder wenn der geplante Zeitpunkt für das Erreichen des letzten Ziels um eine bestimmtes zusätzliches Zeitintervall überschritten wurde. Spätestens zum Betriebsschluss können alle gespeicherten Sitzungen gelöscht werden. Für den Fall, dass ein Benutzer während einer laufenden Navigationssitzung Änderungen an seinen Zielen vornehmen möchte, so kann er dies mittels eines Terminals einem Mitarbeiter mitteilen.

Der allgemeine Ablauf des Systems und die beteiligten Komponenten sind in Abbildung 5.11 zusammengefasst: Am Anfang steht das Einlesen von Gebäudeplänen im DXF-Format,

5. Indoor Navigation

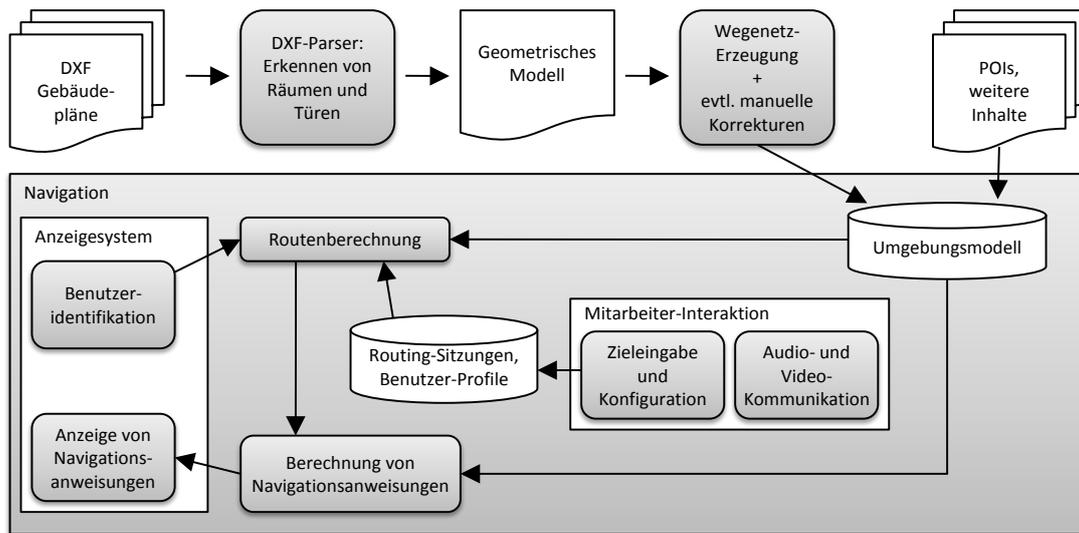


Abbildung 5.11.: Überblick des Navigationssystems und der beteiligten Komponenten.

aus denen die Lage und Form von Räumen, Türen, Treppen etc. ausgelesen wird. Für dieses geometrische Modell wird anschließend ein Wegenetz erzeugt. Dabei kann es aufgrund von Inkonsistenzen in den Bauplänen erforderlich sein, dass das Wegenetz noch stellenweise manuell korrigiert werden muss, wenn beispielsweise eine Tür nicht korrekt erkannt wurde oder wenn ein Polygonzug nicht geschlossen war. Zusammen mit weiteren Informationen über POIs und besondere Übergänge im Wegenetz, z.B. eine Sicherheitsüberprüfung, entsteht ein Umgebungsmodell, das in einer Datenbank hinterlegt wird. Nachdem ein Benutzer einem Mitarbeiter seine Präferenzen mitgeteilt hat, wird eine neue Sitzung mit einem Benutzer-Profil angelegt, in dem die zu besuchenden POIs sowie Daten zur Art der gewünschten Route, z.B. Vermeidung von Treppen, hinterlegt sind. Die Routenberechnung wird gestartet, sobald ein Benutzer an einem der Terminals oder an einem Monitor identifiziert wurde. Dazu wird zunächst mit Hilfe des Umgebungsmodells und anhand des Benutzer-Profiles eine geeignete Route berechnet, anschließend wird die Route an ein weiteres Modul übergeben, das aus der Abfolge der Wegpunkte und unter Zuhilfenahme des Umgebungsmodells die passenden Navigationsanweisungen ermittelt. Im letzten Schritt werden die Anweisungen an das Anzeigesystem übergeben und dem Benutzer angezeigt.

5.3.2. Kontextsensitive Wegewahl

Die Berücksichtigung mehrerer Zwischenziele und die zeitlichen Abhängigkeiten zwischen diesen Zielen stellen bei der Wegewahl ein nicht triviales Problem dar. Allgemein handelt es sich um ein Travelling-Salesman-Problem mit gewissen Zusatzbedingungen:

- Der erste und letzte Knoten des Weges sind durch den aktuellen Standort und das letzte Ziel (z.B. Gate, S-Bahn-Station oder Parkhaus) fest vorgegeben.

5. Indoor Navigation

- Zwischen manchen der Ziele besteht eine Abhängigkeit in Bezug auf die erlaubte Reihenfolge (z.B. Check-In vor Gate, alle Knoten im öffentlichen Bereich vor allen Knoten im Sicherheitsbereich).
- Manche Ziele erfordern eine gewisse Aufenthaltsdauer, bevor der Weg fortgesetzt werden kann. Oder aber das Ziel kann nur bis zu bzw. ab einem bestimmten Zeitpunkt aufgesucht werden.
- In einer Anfrage kann neben expliziten Stationen auch ein *Zieltyp* angegeben werden. Dieser repräsentiert einen bestimmten Typ von POI, z.B. *Restaurant*, *Bank* oder *Frisör*, und bedeutet, dass eine Instanz dieses Zieltyps besucht werden soll.

Die Datengrundlage für die Routenberechnung sind die Wegpunkte und Kanten des Wegenetzes sowie die POIs. Den Kanten sind zusätzlich Eigenschaften zugeordnet, um so z.B. zwischen normalen Verbindungen, Treppen, Aufzügen, Laufbändern etc. zu unterscheiden. Wie bereits im Kapitel 4 diskutiert, werden die Übergänge zwischen zwei direkt verbundenen Räumen durch zwei Knoten dargestellt. Dadurch existiert eine dedizierte Kante für den Übergang und kann so zum Beispiel auch Zutrittsberechtigungen abbilden.

Angenommen es soll eine kürzester Weg über n verschiedene POIs berechnet werden, so ergeben sich allgemein $n!$ mögliche Permutationen für die Reihenfolge der POIs. Sei Z die Menge der verschiedenen *Ziele* für eine Routenberechnung und sei $z_t \in Z$ ein bestimmter Zieltyp, dem eine nicht leere Menge von POIs angehört. Bei der Suche nach einem kürzesten Weg über n verschiedene Ziele, die jeweils einem anderen Zieltyp angehören, erhöht sich die Anzahl der möglichen Wege auf $\prod_{t=1}^n (|z_t|) * n!$. Angenommen, es existieren vier verschiedene Ziele und für jedes Ziel stehen drei Alternativen zur Auswahl, beispielsweise drei Frisöre, drei Restaurants, drei Buchläden und drei Cafés, dann ergeben sich (ohne zusätzliche Einschränkungen) bereits $3 * 3 * 3 * 3 * 24 = 1944$ verschiedene Wege. Für jeden dieser Wege muss seine Länge bestimmt werden, um so den optimalen Weg zu finden. Die Anzahl der Permutationen lässt sich jedoch stark reduzieren, wenn die Reihenfolgebeziehungen zwischen den Zielen im Vorhinein ausgewertet werden. Auf diese Art und Weise werden nur diejenigen Permutationen weiter ausgewertet, die eine zulässige Abfolge von Zielen darstellen.

Unterschiedliche Wege lassen sich einerseits anhand ihrer Länge vergleichen, andererseits kann auch die insgesamt benötigte Zeit betrachtet werden. Dazu müssen sowohl die Gehzeiten (Länge der Kante / Geschwindigkeit des Benutzers) als auch die Wartezeiten bei Zielen beachtet werden. Bevor jedoch die eigentliche Routenberechnung gestartet wird, können auch noch die Präferenzen bzw. das Profil des Benutzers berücksichtigt werden. Dadurch können bestimmte Kanten, die nicht in das Profil des Benutzers passen (z.B. Treppen) im Vorhinein bei der Suche ausgeblendet werden.

5.3.3. Test und Evaluierung

Das vorgestellte System wurde in Form eines Prototypen in einem nichtöffentlichen Bereich am Flughafen München umgesetzt. Auf einer Fläche von $2.800m^2$ und verteilt über zwei

5. Indoor Navigation

Stockwerke wurde eine Testumgebung aufgebaut, in der durch Trennwände ein Parcours mit über 40 POIs inklusive Sicherheitsüberprüfung, Check-In und vier Gates aufgebaut wurde.

In einem Nutzertest wurden an vier Tagen insgesamt 11 Frauen und 21 Männer im Alter zwischen 20 und 72 Jahren einzeln vor die Aufgabe gestellt, mit Hilfe des Navigationssystems verschiedene Routen zu absolvieren. Dazu wurden den Teilnehmern auf Hinweisblättern unterschiedliche Rollen mit Aufgaben zugeteilt: z.B. „*Sie fliegen mit dem Flug LH971 nach Frankfurt, möchten vorher noch einen Kaffee trinken und Briefmarken kaufen. Außerdem und suchen noch ein Schuhgeschäft.*“ Neben der Rolle eines abfliegenden Passagiers wurde auch ein Fluggast betrachtet, der gerade am Flughafen gelandet ist.

Vor Beginn des Test wurden den Teilnehmern keine weiteren Informationen darüber gegeben, was genau sie erwartet. Lediglich die Angaben des Hinweisblattes zu ihrer Rolle waren bekannt. Alle Teilnehmer wurden einzeln mit dem System konfrontiert und hielten sich vor dem Start in einem gesonderten Bereich auf. An zwei Video-Terminals (siehe Abbildung 5.8) konnten die Teilnehmer mit Mitarbeiterinnen und Mitarbeitern des Flughafens sprechen, die sich an einem Arbeitsplatz in einem separaten Raum befanden. Dort erfolgte die Eingabe der gewünschten Ziele, und anschließend wurde über das Terminal eine RFID-Karte an den Teilnehmer ausgegeben. Sowohl an den Terminals, als auch an fünf weiteren Fluggastanzeigen (siehe Abbildung 5.9) konnten sich die Teilnehmer anschließend die einfache und die detaillierte Ansicht (siehe Abbildung 5.10) der personalisierten Navigationsanweisungen anzeigen lassen. Jeder Testlauf dauerte ca. 25 Minuten, und anschließend wurde jeder Teilnehmer sowohl über einen Fragebogen als auch mündlich befragt. Für 18 Teilnehmer war die einfache Ansicht mit einem Pfeil ausreichend, um alle Ziele zu finden. Hiervon gehörten 11 Teilnehmer der Altersgruppe 50+ an. Dabei gab es keine geschlechtsspezifischen Unterschiede. Die älteren Teilnehmer bevorzugten nach eigener Aussage die Pfeil-Ansicht, da sie einfacher zu verstehen sei. 24 Teilnehmer bewerteten die detaillierte Anzeige als hilfreich. Präferenzen zur Kartendarstellung oder zur textbasierten Darstellung der Route waren unter den Teilnehmern gleich verteilt. Ein weiteres Ergebnis war, dass sich die detaillierten Entfernungsangaben nur bei in der Nähe befindlichen Zielen als nützlich erweisen, für weiter entfernte Ziele sollten abstraktere Anweisungen gegeben werden. Alle Teilnehmer gaben ohne Ausnahme an, dass die Einbeziehung von Landmarken für sie hilfreich war.

Die automatische Erzeugung von Wegenetzen aus CAD-Dateien stellt nach wie vor eine Herausforderung dar, da immer wieder kleine Fehler enthalten sind, die manuell korrigiert werden müssen. Dazu sind in Zukunft geeignete Tools notwendig, mit deren Hilfe sich das Wegenetz schneller und einfacher bearbeiten lässt. Gleichzeitig fehlen bislang geeignete Konzepte für unterschiedliche Sichten auf die CAD-Daten. Ein Problem stellen hier die verschiedenen Bereiche eines Gebäudes dar, die zum Teil öffentlich sind und zum Teil nur von speziellem Personal betreten werden dürfen. Um in Zukunft eine manuelle Aufbereitung bzw. Filterung solcher Daten zu vermeiden, bedarf es bereits beim Anlegen der Pläne geeigneter Mittel, um Berechtigungen für die verschiedenen Plandetails festzulegen. Bei der Navigation hat sich das Konzept des Wegpunkte-Overlay-Graph als ein wichtiges Hilfsmittel für die Verbesserung der Navigationsanweisungen erwiesen.

5. Indoor Navigation

In diesem Abschnitt wurde die Navigation auf verteilten Anzeigesystemen demonstriert. Das Konzept lässt sich prinzipiell auch auf mobile Endgeräte übertragen, nur sind dafür vor allem geeignete Positionierungsverfahren erforderlich, die auch in entlegenen Arealen oder Bereichen mit einer sehr hohen Anzahl von Personen funktionieren. Im folgenden Abschnitt wird ein orthogonaler Ansatz beschrieben, bei dem die Navigationsanweisungen auf einem mobilen Endgerät dargestellt werden und bei dem keine Kommunikation mit einem zentralen Server erforderlich ist.

5.4. Spontane und anonyme Navigation

Navigation kann sowohl mittels stationärer als auch mobiler Systeme realisiert werden. Aus der Sicht des Benutzers ist es zusätzlich auch entscheidend, wie und durch welchen Akteur die Navigationsanweisungen berechnet werden, da in manchen Konstellationen die Privatsphäre des Benutzers gefährdet sein kann. Insbesondere dann, wenn ein mobiles Endgerät verwendet wird, stellt sich die Frage nach der geeigneten Berechnung der Navigationsdaten. Angenommen ein Benutzer betritt ein fremdes Gebäude zum ersten Mal und möchte spontan einen Indoor Navigationsdienst auf seinem mobilen Endgerät nutzen. Dazu müssen mindestens

1. ein geeignetes Positionierungsverfahren und
2. alle benötigten Navigationsdaten vorhanden sein.

Für etliche Szenarien wie z.B. die Navigation auf einem Campus, in einem Krankenhaus oder an einem Flughafen ist es wünschenswert, dass jeder Benutzer anonym bleibt, d.h. dass weder sein Aufenthaltsort noch seine Identität bekannt wird. Ausgenommen davon sind Dienste, bei denen der Benutzer ein berechtigtes Interesse daran hat, sich Dritten gegenüber kenntlich zu machen. Die Verwendung von satellitengestützten Positionierungsverfahren auf kleinen mobilen Endgeräten scheidet, zumindest nach dem heutigen Stand der Technik, aus, und es stellt sich bei den meisten der heute verfügbaren Indoor Positionierungsverfahren das Problem, dass der Benutzer verfolgt oder sogar identifiziert werden kann. Beispiele dafür sind der Verbindungsaufbau zu einem WLAN-Access-Point oder das Übermitteln von gemessenen Parametern an einen zentralen Positionierungsserver. Aber auch personalisierte Navigationsanweisungen können ein Datenschutzproblem darstellen, wenn sie durch einen zentralen Anbieter erzeugt werden und hinreichend viele Informationen enthalten, die einen einzelnen Benutzer identifizierbar machen. So wird z.B. von Mulloni [102] ein Ansatz beschrieben, bei dem der Benutzer mit seinem mobilen Endgerät einen Barcode [71] fotografiert und anschließend von einem Server eine entsprechende Karte herunter geladen wird.

Dem Datenschutz gegenüber steht der Wunsch nach genauen und aktuellen Navigationsanweisungen. Angenommen es wird ein rein endgerätbasiertes Positionierungsverfahren eingesetzt, d.h. die Positionsbestimmung erfolgt auf dem Endgerät, und es werden keine Daten an die Infrastruktur übermittelt. Dann bleibt zunächst das Problem der anonymen

5. Indoor Navigation

Navigationsdaten bestehen. Liegt jedoch ein vollständiges Umgebungsmodell auf dem Endgerät vor, so können alle Berechnungen spontan und anonym durchgeführt werden.

Im Folgenden wird ein Ansatz diskutiert, der erstmals in [125] vorgestellt wurde und der Navigationsdaten in einem kompakten Format für mobile Endgeräte bereitstellt sowie alle benötigten Navigationsanweisungen bereits vollständig enthält. Gleichzeitig wird ein Barcode-basiertes Positionierungsverfahren eingesetzt, das eine anonyme Positionsbestimmung erlaubt. Die Grundidee besteht darin, dass ein Benutzer vor dem Betreten eines fremden Gebäudes zunächst eine Datei mit im Voraus berechneten Routen auf seinem mobilen Endgerät speichert. In einer mobilen Navigationsanwendung kann das gewünschte Ziel eingegeben werden und jedes Mal, wenn einer der in der Umgebung befindlichen Barcodes gescannt wurde, wird dem Benutzer eine Navigationsanweisung angezeigt. Es wird davon ausgegangen, dass innerhalb des Gebäudes an ausreichend vielen Stellen entsprechende Barcodes angebracht sind, beispielsweise neben jeder Türe und an den Abzweigungen von Fluren. Um die Anzahl der anzubringenden Barcodes zu reduzieren bietet sich auch eine Verteilung analog zu dem Ansatz von Brunner-Friedrich und Radoczky [22] an: darin werden aktive Landmarken (Funksender) in einem Gebäude an den Positionen angebracht, an denen die Knoten im zugrunde liegenden Wegegraphen den höchsten Verzweigungsgrad aufweisen.

Da alle benötigten Navigationsdaten auf dem Endgerät vorliegen, ist für die Navigation keine weitere Kommunikation mit einem zentralen Server notwendig. Ebenso wird durch die Verwendung einer barcodebasierten Positionierung gewährleistet, dass auch das Positionierungsverfahren keinen Datenaustausch mit externen Systemen erfordert.

Eine Vorberechnung der Routen ermöglicht es, dass auf dem Endgerät Anfragen schnell und ohne größeren Aufwand beantwortet werden können. Dazu müssen einmalig alle Routen zwischen allen Paaren von Zielpunkten berechnet werden. Zur Lösung dieses *All-Pairs-Shortest-Path-Problems* empfiehlt sich beispielsweise der Floyd-Warshall-Algorithmus [48] oder Johnsons Algorithmus [67]. Ersterer bietet sich an für Graphen, deren Anzahl von Kanten nahe der maximalen Anzahl der möglichen Kanten liegt. Jedoch hat er eine Zeitkomplexität von $O(V^3)$, wobei V die Anzahl der Knoten ist. Bei vielen Gebäuden wird die Anzahl der Kanten wesentlich geringer sein, und dann eignet sich Johnsons Algorithmus mit einer Zeitkomplexität von $O(V^2 \log V + VE)$.

Die Basis für die Vorberechnung sind 1.) Informationen über zugehöriges Kartenmaterial für die einzelnen Stockwerke und deren *Pixel-pro-Meter-Verhältnis* sowie 2.) Informationen über jeden Barcode im Gebäude. Dazu wird jedem Barcode nicht nur eine eindeutige ID und eine geometrische Koordinate, sondern auch ein natürlichsprachlicher Bezeichner zugeordnet. So können Ziele auch anhand eines Namens, beispielsweise der Raumbezeichnung oder des Mitarbeiternamens gesucht werden, und anhand der Koordinaten können Positionen auf einer Karte dargestellt werden.

Routeninformationen werden gespeichert, indem zu jeder Route nur jeweils der nächste Abschnitt bis zum nächsten Barcode abgebildet wird. Betrachtet man beispielsweise eine Route von Barcode b_a zum Barcode b_e , die über drei weitere dazwischen liegende Positionen verläuft ($b_a \rightarrow b_b \rightarrow b_c \rightarrow b_d \rightarrow b_e$). In diesem Fall wird für die Position b_a und das Ziel

5. Indoor Navigation

<code><DATA></code>	::= <code><HEADER> <MAPS> <BAR_INFO> <BAR_DEST></code>
<code><HEADER></code>	::= <code><VERSION> <NAME_BUILDING> <LENGTH_BARCODE> <LENGTH_FLOOR></code>
<code><MAPS></code>	::= <code><LENGTH_MAPS> (<FID> <PPM> <LENGTH_MAP> <MAP>)*</code>
<code><BAR_INFO></code>	::= <code><LENGTH_INFO> (<BARCODE> <DESCR> <FID> <X> <Y>)*</code>
<code><BAR_DEST></code>	::= <code><LENGTH_DEST> (<BARCODE>, <LENGTH_BARDEST> (<BARCODE> <DIR> <LENGTH_SEGMENT> <BARCODE>)*)*</code>
<code><BARCODE></code>	::= Number of a barcode
<code><X></code>	::= X coordinate of a barcode on a map
<code><Y></code>	::= Y coordinate of a barcode on a map
<code><MAP></code>	::= Image of a floor
<code><VERSION></code>	::= Protocol version
<code><NAME_BUILDING></code>	::= Name of building
<code><LENGTH_BARCODE></code>	::= Number of bits used to represent a barcode
<code><LENGTH_FLOOR></code>	::= Number of bits used to represent a floor
<code><LENGTH_MAPS></code>	::= Number of bits used for all floor descriptions together
<code><FID></code>	::= Floor ID
<code><PPM></code>	::= Pixel per meter ratio
<code><LENGTH_MAP></code>	::= Size (in bits) of a map
<code><LENGTH_INFO></code>	::= Number of bits used for all barcode information
<code><DESCR></code>	::= Description of a barcode
<code><LENGTH_DEST></code>	::= Number of bits used for all routes
<code><LENGTH_BARDEST></code>	::= Number of bits used for all routes starting from one specific barcode
<code><DIR></code>	::= Direction to go to next barcode
<code><LENGTH_SEGMENT></code>	::= Distance between current position and next barcode

Abbildung 5.12.: Aufbau der Navigationsdaten in Backus-Naur-Form

b_e nur der nächste Abschnitt ($b_a \rightarrow b_b$) abgespeichert. Für b_b und das Ziel b_e wird ebenso nur der nächste Abschnitt ($b_b \rightarrow b_c$) hinterlegt usw. Die vollständige Route von b_a nach b_e kann durch die Konkatenation der einzelnen Abschnitte berechnet werden. Für jedes Paar von Barcode-IDs wird lediglich ein Tripel bestehend aus der ID, der Richtung und der Entfernung zum nächsten Barcode gespeichert. So kann eine Navigationsanweisung auf dem mobilen Endgerät in $O(1)$ ermittelt werden und eine gesamte Route aus m Abschnitten in $O(m)$. In obigem Beispiel wären somit vier Aufrufe notwendig, um die gesamte Route zu ermitteln.

Abbildung 5.12 zeigt den Aufbau des Datenformates in Backus-Naur-Form. Das Ziel dieses Formates ist es, den Speicherplatzbedarf auf dem mobilen Endgerät zu minimieren und alle benötigten Navigationsdaten ohne Redundanzen abzubilden. Entscheidend ist hierbei die im Header definierte Anzahl von Bits für eine einzelne Barcode-ID. Die Länge dieses Feldes kann an die Gesamtanzahl der in einem Gebäude betrachteten Barcode-ID angepasst werden. Zusätzlich ist eine schnelle Suche in der Datei möglich, da alle Einträge in den Feldern *MAPS*, *BAR.INFO* und *BAR.DEST* entweder eine fixe oder definierte Größe haben. Das Feld *DIR* gibt die Richtung zum nächsten Wegpunkt an, bei einer Länge von 3 Bit kann damit ein 8-Corner-System für die Richtungsangaben verwendet werden.

Speicherkomplexität Das primäre Ziel des vorgestellten Datenformates ist es, den Speicherplatzbedarf auf dem mobilen Endgerät zu minimieren und für verschiedene Arten und Größen von Gebäuden einsetzbar zu sein. Nachfolgend soll deshalb betrachtet werden, wie

5. Indoor Navigation

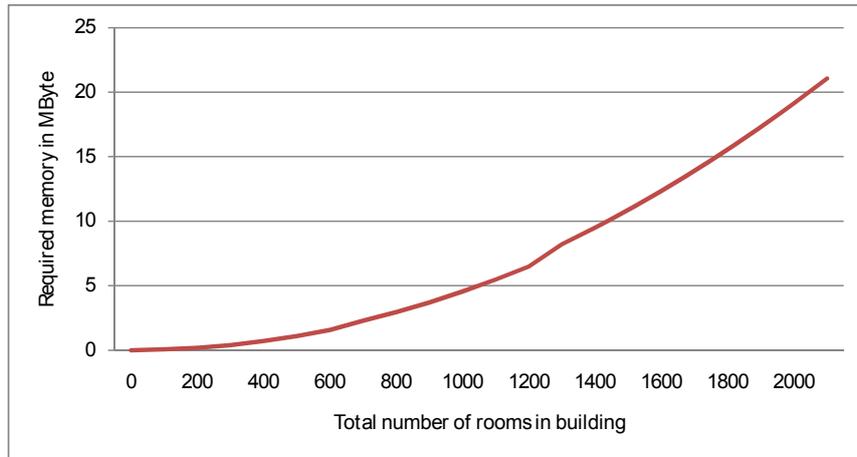


Abbildung 5.13.: Speicherplatzbedarf in Abhängigkeit der Anzahl der Barcodes.

sich der Speicherplatzbedarf in Abhängigkeit der Anzahl von Barcodes und Stockwerken verändert.

Sei n die Anzahl der in einem Gebäude vorhandenen Barcodes. In vielen Umgebungen wird dies etwa der Anzahl an Räumen im Gebäude entsprechen. Sei f die Anzahl der Stockwerke in einem Gebäude. Ferner seien $Length_{Desc}$ die maximale Bitlänge für eine Raumbezeichnung, $Length_X$ und $Length_Y$ die Bitlänge der Koordinaten, sowie $Length_{BID}$ die Bitlänge einer Barcode-ID. Dann lässt sich der benötigte Speicherplatz MEM folgendermaßen berechnen:

$$\begin{aligned}
 MEM &= (Length_{BID} * n) * (Length_{Desc} + Length_X + Length_Y) \\
 &\quad + n * (n - 1) * (3 * Length_{BID} + Length_{Dir}) \\
 &\quad + overhead
 \end{aligned} \tag{5.1}$$

Dabei gilt für die für die Bitlänge zur Speicherung einer Barcode-ID:

$$Length_{BID} = \left\lceil \log_2\left(\frac{n}{f}\right) \right\rceil + \lceil \log_2(f) \rceil = \lceil \log_2(n) \rceil \tag{5.2}$$

Abbildung 5.13 zeigt die resultierende Größe der Datei in Abhängigkeit der Anzahl von Barcodes. Dabei wurde für die Beschreibung von Räumen ein Wert von $Length_{Desc} = 160$ Bit (entspricht 20 Zeichen), sowie 16 Bit pro Koordinate gewählt. So kann man beispielsweise erkennen, dass in einem Gebäude mit 1000 angebrachten Barcodes eine Datei mit einer Größe von 5 MB ausreicht, um alle nötigen Navigationsdaten zu speichern.

Der Ansatz wurde als Prototyp für *.NET Compact Framework* umgesetzt und in einem Teil eines Gebäudes wurden Barcodes an den Türrahmen in Augenhöhe angebracht. Die Routenberechnung auf dem Endgerät funktioniert schnell und unkompliziert, ebenso die Anzeige. Offensichtlich ist das vorgestellte Vorgehen nicht an die Verwendung von Barcodes gebunden. Denkbar ist auch jedes andere Verfahren, das es einem Benutzer ermöglicht,

5. Indoor Navigation

sich ohne Einbeziehung der Infrastruktur zu positionieren. So könnten im einfachsten Fall anstatt von Barcode-IDs auch die empfangenen *Basic Service Set Identifier* umliegender WLAN-Access-Points verwendet werden. Wenngleich dies zu einer wesentlich ungenaueren Position führt, wäre doch ein Einsatz in großen Anlagen wie z.B. einem Flughafen denkbar. Auch muss beachtet werden, dass die rein passive Nutzung eines WLAN-Adapters oftmals Modifikationen an bestehenden Treibern erfordert. *Near Field Communication* [42] ist eine weitere Möglichkeit, hier ist wie bei Barcodes die Verteilung zahlreicher Tags im Gebäude notwendig.

5.5. Zusammenfassung und Diskussion

Die Indoor Navigation stellt ein Umfeld dar, in dem viele verschiedene Herausforderungen zusammentreffen. Zum einen sind Positionierungsverfahren erforderlich, zum anderen müssen kürzeste Wege berechnet werden, und für den Benutzer wird eine leicht zu verstehende Schnittstelle benötigt. In diesem Kapitel wurde näher auf die Berechnung und Anzeige von Navigationsanweisungen eingegangen. Konzentriert man sich nicht auf eine spezielle Anwendung, so besteht ein Problem darin, dass bei heterogenen Wegenetzen verschiedene Effekte auftreten können, die die automatisierte Erzeugung von Navigationsanweisungen beeinflussen. Mit dem Zusammenfassen von Anweisungen und der Verwendung von Verbindungs- und Schaltungspunkten wurde eine Möglichkeit vorgestellt, wie sich Navigationsanweisungen in einem 8-Corner-System verbessern lassen. Die Umsetzung dieses Konzeptes wurde anhand einer prototypischen Implementierung eines Indoor Navigationssystems am Flughafen München vorgestellt.

Die verteilten Anzeigen und die Einbindung eines Navigationssystems in den organisatorischen Ablauf eines Flughafens haben zahlreiche weitere Probleme aufgeworfen, deren Lösungen im Rahmen dieser Arbeit nicht weiter behandelt wurden. Je höher die Anzahl der unterschiedlichen Kontext-Quellen ist (z.B. Flugdaten in Echtzeit, Länge von Warteschlangen, dynamische Berechtigungen für einzelne Durchgänge), umso größer wird letztendlich der Aufwand für die Integration dieser Informationen. Dasselbe gilt für eine große Anzahl von heterogenen Hardware-Elementen in den Terminals.

Bei der Interaktion des Benutzers mit einer öffentlichen Anzeige besteht grundsätzlich das Problem der Privatsphäre: der Benutzer möchte einerseits personalisierte Informationen übersichtlich angezeigt bekommen, andererseits sollen diese so dargestellt werden, dass sie nicht jede gerade vorbeilaufende Person sofort lesen kann. Daher sollten persönliche Daten, falls überhaupt notwendig, nur in kleinem Format angezeigt werden bzw. nur auf Terminals eingeblendet werden, die über einen entsprechenden Sichtschutz verfügen.

In Zukunft wird mit Sicherheit auch die Indoor-Navigation über mobile Endgeräte zunehmen. Der Ansatz zur spontanen und anonymen Navigation stellt dazu eine Möglichkeit dar, wie ein kompaktes Modell vollständig auf dem Endgerät vorgehalten werden kann.

6. Zusammenfassung und Ausblick

Die Realisierung ortsbezogener Dienste in Gebäuden stellt auf verschiedenen Ebenen eine Herausforderung dar, und diese Arbeit hat sich auf die Bereitstellung von Umgebungsmodellen und Navigationsdaten für solche Dienste konzentriert. Die vorgestellten Ansätze und Konzepte ermöglichen es, die Eigenschaften einer Umgebung bzw. eines Gebäudes so abzubilden, dass mehrere unterschiedliche Dienste und Anwendungen darauf zugreifen und die jeweils relevanten Informationen erhalten können. Ein wichtiger Kerngedanke besteht darin, die Begehbarkeit der Umgebung in variierender Komplexität zu betrachten. Dadurch können für unterschiedliche Arten von Diensten auch unterschiedliche Arten von entsprechenden Wegenetzen verwendet werden. Dies wird besonders dann relevant, wenn ein Wegenetz nicht nur für die Berechnung kürzester Wege benötigt wird, sondern beispielsweise auch für die Visualisierung des Weges oder zur Erzeugung von Navigationsdaten eingesetzt werden soll. Für die Indoor Navigation wurden Ansätze vorgestellt, die es ermöglichen, Navigationsanweisungen auf der Grundlage des Umgebungsmodells zu erzeugen. Solche Anweisungen können in stationären und in mobilen Navigationssystemen verwendet werden, um einen Benutzer zu seinem Ziel zu leiten. Dazu wurde auch gezeigt, wie Landmarken in diese Anweisungen eingebunden werden können. Zur Positionierung mobiler Zielobjekte von Mehrbenutzer-LBS wurde ein neuer Ansatz zur energieeffizienten Kombination von Positionierungsverfahren vorgestellt. Dieser sieht einen dynamischen Wechsel des Verfahrens in Abhängigkeit der Aufenthaltsorte der Zielobjekte vor und ermöglicht aufgrund des geringeren Energieverbrauchs eine wesentlich längere Dienstnutzung.

Die Untersuchung der verschiedenen Arten von Wegenetzen hat die jeweiligen Vor- und Nachteile bei der Verwendung für Indoor Location-based Services aufgezeigt. Anhand der empirischen Untersuchung wurde gleichzeitig die Umsetzbarkeit des Ansatzes demonstriert, und mittels des vorgestellten GML-Anwendungsschemas können Umgebungsmodelle für I-LBS in einem einheitlichen Format gespeichert und bereitgestellt werden. In Bezug auf die Nutzung von Umgebungsmodellen durch mobile Dienste wurden Ansätze vorgestellt, die eine Bündelung von Anfragen mehrerer mobiler Anwendungen ermöglichen und die das Modell eines Gebäudes in einem kompakten Format für die Indoor-Navigation bereitstellen. Die Umsetzung eines Indoor-Navigationssystems am Flughafen München vereint viele der vorgestellten Herausforderungen, und die vorgestellte prototypische Implementierung zeigt entsprechende Lösungsansätze auf.

6.1. Ausblick

Aus den vorgestellten Problemen und Lösungen ergeben sich auch weitere Fragestellungen, auf die im Folgenden näher eingegangen wird.

6.1.1. Marktplätze für Umgebungsmodelle

Die Erstellung eines Umgebungsmodells erfordert nicht selten eine genaue Kenntnis des Gebäudes, um beispielsweise beim Import von CAD-Daten oder bei der Interpretation der Gebäudepläne Fehler zu vermeiden. Ist das Modell jedoch erst einmal erzeugt, dann kann es ohne Weiteres vollautomatisch verarbeitet werden. Um entsprechende Anfragen eines Dienstes zu bearbeiten (z.B. die Suche nach einem Raum oder die Berechnung eines kürzesten Weges), bedarf es jedoch der geeigneten Softwarekomponenten, um diese Berechnungen durchführen zu können. Sofern jeder neue Dienst diese Funktionalitäten selbst bereit stellen müsste, wäre damit ein unnötiger Mehraufwand verbunden. Eine mögliche Erweiterung der vorgestellten Konzepte ist daher ein „Marktplatz“, auf dem verschiedene Anbieter ihre Umgebungsmodelle nicht nur zum Download für andere Dienste anbieten können, sondern über den auch gezielt Berechnungsanfragen zu einem spezifischen Umgebungsmodell gestellt werden können. Eine solche Plattform würde es ermöglichen, dass sich der einzelne Umgebungsmodell-Erzeuger nicht um die Details der späteren Berechnungen kümmern muss und lediglich ein konformes Modell bereit stellt. Aus Sicht der Dienste hätte ein solcher Marktplatz den Vorteil, dass Umgebungsinformationen zu verschiedenen Gebäuden unter einer einheitlichen Schnittstelle verfügbar wären. Eine ähnliche Idee wurde von Bareth et al. [12] vorgestellt: darin wird ein Marktplatz beschrieben, auf dem Anbieter ihre ortsbezogenen *Dienste* einstellen können. Bei den Diensten handelt es sich um sogenannte *Geofencing-Dienste*, die in Abhängigkeit von Aufenthaltsort, geographischen Bereichen und einem Regelwerk bestimmte Aktionen auslösen. Auch auf einem Marktplatz für Umgebungsmodelle könnte der Betreiber des Marktplatzes solche zusätzlichen Funktionen anbieten und so beispielsweise Positionsdaten von mobilen Zielobjekten verwalten.

6.1.2. Spontaner Zugriff auf Indoor-Positionierungsverfahren

Eine Herausforderung bei der Nutzung von Indoor-Positionierungsverfahren ist nach wie vor der „erste Kontakt“ zwischen dem Zielobjekt und dem Positionsermittler (der Positionierungs-Infrastruktur). Das Zielobjekt ist dabei idealerweise in der Lage, autonom zu erkennen, welche Positionierungsverfahren gerade in der Umgebung vorhanden sind und kann selbstständig eine Verbindung zu dem jeweiligen System aufbauen. Für die vom Terminal unterstützten und die netzbasierten Positionierungsverfahren bedeutet dies, dass sie ihre Verfügbarkeit über eine einheitliche Schnittstelle ankündigen müssen. Eine Möglichkeit dafür wäre beispielsweise, diese Informationen über *Beacon-Frames* zu verteilen, die ein WLAN-Access-Point periodisch ausstrahlt. In diesen Rahmen können unter anderem frei definierbare sogenannte *vendor specific information elements* übertragen werden, d.h. es

6. Zusammenfassung und Ausblick

lassen sich dadurch auf der Ebene des Medienzugriffsverfahrens beliebige zusätzliche Informationen einbinden [63, S. 128]. Eine Bekanntmachung von verfügbaren Positionierungsverfahren über einen solchen oder andere Broadcast-Kanäle hätte zusätzlich den Vorteil, dass ein Zielobjekt keine explizite Anfrage an ein System stellen muss und dadurch seine Anwesenheit zunächst verborgen bleibt. Auch der in Abschnitt 5.4 vorgestellte Ansatz zur anonymen Navigation könnte durch einen solchen Mechanismus erweitert werden, indem eine Basisstation über zusätzliche Kontrollrahmen periodisch im Broadcast einen Ausschnitt des Umgebungsmodells versendet.

6.1.3. Indoor-Navigation über neue Benutzerschnittstellen

Sequenzen von Navigationsanweisungen bilden eine Grundlage für die Indoor-Navigation, sie müssen jedoch immer in Verbindung mit einer konkreten Darstellung in einem System betrachtet werden. Im Rahmen dieser Arbeit wurde eine Möglichkeit zur Umsetzung auf öffentlichen Anzeigesystemen gezeigt. Für andere Umgebungen bzw. Szenarien sind auch viele andere Arten von Benutzerschnittstellen denkbar. Die zahlreich verfügbaren mobilen Endgeräte bieten dazu verschiedenste Möglichkeiten, um über Sprache, visuelle Instruktionen oder auch haptische Reize zu navigieren. Mobile bzw. tragbare Systeme stehen jedoch mitunter vor dem Problem, dass die aktuelle Position des Benutzers nicht oder nur ungenau bekannt ist. In solchen Situationen wären Ansätze von Interesse, die Navigationsanweisungen dynamisch an die gegenwärtige Ungenauigkeit des Positionierungssystems anpassen. Ein weiterer Aspekt sind Rückmeldungen durch den Benutzer: Falls eine Anweisung nicht korrekt befolgt und der geplante Weg verlassen wurde, so kann dies entweder an einer bewussten Entscheidung oder an einer Fehlinterpretation durch den Benutzer liegen. Im letzteren Fall könnte mit Hilfe von impliziten (oder auch expliziten) Rückmeldungen vom Benutzer darauf reagiert werden, sodass beispielsweise eine generischere Anweisung gegeben wird. Vielversprechende Möglichkeiten ergeben sich auch durch die Kombination mit Virtual-Reality-Anwendungen, so dass Navigationsinformationen in einem Echtzeit-Video auf einem mobilen Endgerät eingeblendet werden.

Abbildungsverzeichnis

2.1. Rollenmodell für Indoor Location-based Services	8
2.2. Terminalbasierte WLAN-Positionierung	12
2.3. Merkmale von Navigationssystemen	19
3.1. Hierarchie von Orten	27
3.2. Zwei Varianten eines einfachen graphbasierten Modells	28
3.3. Kombination von GPS und Zellen-basierter Positionierung	45
3.4. Histogramm der Größe von GSM-Zellen in einem städtischen Mobilfunketz	46
3.5. GPS und GSM-Cell-ID-Positionierung: Simulationsergebnisse	48
3.6. Histogramm der GPS-Zeitintervalle	49
3.7. Genauigkeit des <i>first fix</i> nach dem Einschalten eines GPS-Empfängers . . .	50
3.8. Zonen für die Überwachung von Zielobjekten in Gebäuden	52
3.9. Verbindung von Raumpolygonen durch Wegpunkte	58
3.10. Konkave Räume und Wegpunkte	59
3.11. Corner Graph	61
3.12. Straight Skeleton	62
3.13. Straight Skeleton eines Polygons mit Loch. Quelle: [44]	63
3.14. Konvexe Partitionierung	63
3.15. Triangulierung einer Punktemenge	65
3.16. Voronoi-Diagramm und die Triangulierung der Orte. Quelle: [49]	66
3.17. Triangulierungen und Möglichkeiten zur Wahl von Wegpunkten	67
3.18. Gitterförmige Wegenetze	69
3.19. Beispiel eines kreisbasierten Wegenetzes. Quelle: [144]	70
3.20. Beispiele für unterschiedliche Klassen von Polygonen.	75
3.21. Einzelnes Stockwerk des zur Simulation verwendeten Gebäudes.	76
3.22. Instanz des <i>CornerGraph</i> für ein einzelnes Stockwerk.	77
3.23. Nachbearbeitung von Polygonen	79
3.24. Empirischer Vergleich von Algorithmen zur Erzeugung von Wegenetzen . .	80
3.25. Beispiele des Waypoint-Graphs und einer Triangulierung	81
3.26. Beispiele der Konvexen Partitionierung	82
3.27. Dynamisches Nachladen aus einem Umgebungsmodell	86
3.28. Simulationsergebnisse zur Speicherung bereits bekannter Routen	89
4.1. GeoMobility-Architektur des OpenGIS Location Service. Quelle: [110] . . .	101
4.2. Übersicht der Klassen in BIGML	104
4.3. BIGML: Site	105

Abbildungsverzeichnis

4.4. BIGML: Building	106
4.5. BIGML: Floor	107
4.6. BIGML: Room	108
4.7. BIGML: AbstractPortalType	109
4.8. BIGML: POI	110
4.9. BIGML: AOI	111
4.10. Area of Interest: Zuordnung von Räumen	112
4.11. BIGML: Graph, Node und Edge	113
4.12. BIGML: GlobalCRSReferencePoint	114
5.1. Richtungsangaben im 8-Corner-System	122
5.2. Einfaches Beispiel für Navigationsanweisungen im 8-Corner-System.	123
5.3. Alternierende Anweisungen aufgrund ungünstig gelegener Wegpunkte	123
5.4. Veränderte Winkel im 8CS und zusammengefasste Anweisungen.	124
5.5. Probleme bei der alleinigen Anpassung der Winkel im 8CS	125
5.6. Verbindungspunkte und Schaltpunkte	126
5.7. Landmarken	127
5.8. Terminal zur Videokommunikation und zur Navigation	132
5.9. Personalisierte Navigation auf öffentlichem Display	133
5.10. Screenshot des Navigationssystems	134
5.11. Überblick des Navigationssystems	135
5.12. Aufbau der Navigationsdaten in Backus-Naur-Form	140
5.13. Speicherplatzbedarf in Abhängigkeit der Anzahl der Barcodes	141

Literaturverzeichnis

- [1] 3RD GENERATION PARTNERSHIP PROJECT. Technical Specification Group Services and System Aspects; Functional stage 2 description of Location Services (LCS) (Release 9) TS 23.271, 2009.
- [2] ABOWD, G. D., ATKESON, C. G., HONG, J., LONG, S., KOOPER, R., PINKERTON, M. Cyberguide: a mobile context-aware tour guide. *Wirel. Netw.* 3 (October 1997), S. 421–433.
- [3] ABOWD, G. D., DEY, A. K. Towards a Better Understanding of Context and Context-Awareness. Technischer Bericht GIT-GVU-99-22, Georgia Institute of Technology, 1999.
- [4] AICHHOLZER, O., AURENHAMMER, F., ALBERTS, D., ARTNER, B. G. A Novel Type of Skeleton for Polygons. In *Journal of Universal Computer Science*, vol. 1, no. 12 (1995).
- [5] AMIR, A., EFRAT, A., MYLLYMAKI, J., PALANIAPPAN, L., WAMPLER, K. Buddy tracking - efficient proximity detection among mobile friends. *Pervasive Mob. Comput.* 3, 5 (2007), S. 489–511.
- [6] ANANTHANARAYANAN, G., HARIDASAN, M., MOHOMED, I., TERRY, D., THEK-KATH, C. A. StarTrack: a framework for enabling track-based applications. In *Proceedings of the 7th international conference on Mobile systems, applications, and services* (New York, NY, USA, 2009), MobiSys '09, ACM, S. 207–220.
- [7] ARBEITSGEMEINSCHAFT DER VERMESSUNGSVERWALTUNGEN DER LÄNDER DER BUNDESREPUBLIK DEUTSCHLAND (ADV). ALKIS-Objektartenkatalog, Version 6.0, April 2008. <http://www.adv-online.de>.
- [8] ARIKAWA, M., KONOMI, S., OHNISHI, K. Navitime: Supporting Pedestrian Navigation in the Real World. *IEEE Pervasive Computing* 6 (2007), S. 21–29.
- [9] ASTUR, R. S., ORTIZ, M. L., SUTHERLAND, R. J. A characterization of performance by men and women in a virtual Morris water task: A large and reliable sex difference. *Behavioural Brain Research* 93, 1-2 (1998), S. 185–190.
- [10] BAHL, P., PADMANABHAN, V. N. RADAR: an in-building RF-based user location and tracking system. In *Proceedings of IEEE Infocom 2000* (Tel-Aviv, Israel, März 2000).

Literaturverzeichnis

- [11] BAMBA, B., SESHADRI, S., LIU, L. Scaling Location-based Services with Dynamically Composed Location Index. In *Proceedings of the IEEE International Conference on Services Computing* (2008).
- [12] BARETH, U., KÜPPER, A., RUPPEL, P. geoXmart - a Marketplace for Geofence-Based Mobile Services. In *Proceedings of the IEEE 34th Annual Computer Software and Applications Conference (COMPSAC 2010)* (Jul 2010), S. 101–106.
- [13] BAUER, M., BECKER, C., ROTHERMEL, K. Location Models from the Perspective of Context-Aware Applications and Mobile Ad Hoc Networks. *Personal Ubiquitous Comput.* 6 (Januar 2002), S. 322–328.
- [14] BAUS, J., KRÜGER, A., WAHLSTER, W. A resource-adaptive mobile navigation system. In *IUI '02: Proceedings of the 7th international conference on Intelligent user interfaces* (New York, NY, USA, 2002), ACM, S. 15–22.
- [15] BECKER, C., DÜRR, F. On location models for ubiquitous computing. *Personal Ubiquitous Comput.* 9, 1 (2005), S. 20–31.
- [16] BECKER, C., DÜRR, F., KNOLL, M., NICKLAS, D., WEIS, T. Entwicklung ortsbezogener Anwendungen. *PIK - Praxis der Informationsverarbeitung und Kommunikation* 29, 1 (2006), S. 30–36.
- [17] BECKER, T., NAGEL, C., KOLBE, T. H. Supporting Contexts for Indoor Navigation Using a Multilayered Space Model. In *Proceedings of The IEEE International Conference on Mobile Data Management* (Los Alamitos, CA, USA, 2009), IEEE Computer Society, S. 680–685.
- [18] BEIGL, M., ZIMMER, T., DECKER, C. A Location Model for Communicating and Processing of Context. *Personal Ubiquitous Comput.* 6 (Januar 2002), S. 341–357.
- [19] BELLAVISTA, P., KÜPPER, A., HELAL, S. Location-Based Services: Back to the Future. *IEEE Pervasive Computing* 7 (2008), S. 85–89.
- [20] BORRIELLO, G., CHALMERS, M., LAMARCA, A., NIXON, P. Delivering real-world ubiquitous location systems. *Commun. ACM* 48 (März 2005), S. 36–41.
- [21] BRUMITT, B., SHAFER, S. Topological World Modeling Using Semantic Spaces. In *Proceedings of the Workshop on Location Modeling for Ubiquitous Computing* (Atlanta, September 2001).
- [22] BRUNNER-FRIEDRICH, B., RADOZKY, V. Active Landmarks in Indoor Environments. In *Visual Information and Information Systems*, S. Bres und R. Laurini, Eds., vol. 3736 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2006, S. 203–215.

Literaturverzeichnis

- [23] BRY, F., LORENZ, B., OHLBACH, H. J., ROSNER, M. A Geospatial World Model for the Semantic Web. In *Principles and Practice of Semantic Web Reasoning*, vol. 3703 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2005, S. 145–159.
- [24] BUTZ, A., BAUS, J., KRÜGER, A., LOHSE, M. A hybrid indoor navigation system. In *IUI '01: Proceedings of the 6th international conference on Intelligent user interfaces* (New York, NY, USA, 2001), ACM, S. 25–32.
- [25] CARROLL, A., HEISER, G. An Analysis of Power Consumption in a Smartphone. In *Usenix Annual Technical Conference* (Boston, MA, USA, Juni 2010).
- [26] CASTRO, P., CHIU, P., KREMENEK, T., MUNTZ, R. R. A Probabilistic Room Location Service for Wireless Networked Environments. In *UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing* (London, UK, 2001), Springer-Verlag, S. 18–34.
- [27] CHANG, Y.-J., TSAI, S.-K., WANG, T.-Y. A context aware handheld wayfinding system for individuals with cognitive impairments. In *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility* (New York, NY, USA, 2008), ACM, S. 27–34.
- [28] CHATILA, R., LAUMOND, J. Position referencing and consistent world modeling for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation* (März 1985), vol. 2, S. 138–145.
- [29] CHAZELLE, B. Convex partitions of polyhedra: a lower bound and worst-case optimal algorithm. *SIAM J. Comput.* 13, 3 (1984), S. 488–507.
- [30] CHEWAR, C. M., MCCRICKARD, D. S. Dynamic route descriptions: tradeoffs by usage goals and user characteristics. In *SMARTGRAPH '02: Proceedings of the 2nd international symposium on Smart graphics* (New York, NY, USA, 2002), ACM, S. 71–78.
- [31] CHIN, F., SNOEYINK, J., WANG, C. A. Finding the Medial Axis of a Simple Polygon in Linear Time. In *Discrete Comput. Geom* (1995), Springer-Verlag, S. 382–391.
- [32] CORBETT, J. P. *Topological Principles in Cartography*. US Bureau of the Census, 1979.
- [33] DE BERG, M., CHEONG, O., VAN KREVELD, M., OVERMARS, M. *Computational Geometry: Algorithms and Applications*. Springer, 2008.
- [34] DEBLAUWE, N. *GSM-based Positioning: Techniques and Application*. Dissertation, Vrije Universiteit Brussel, 2008.

Literaturverzeichnis

- [35] DEBLAUWE, N., RUPPEL, P. Combining GPS and GSM Cell-ID positioning for Proactive Location-based Services. In *Proceedings of the 4th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS)* (Philadelphia, PA, August 2007).
- [36] DEBLAUWE, N., RUPPEL, P., VAN BIESEN, L. Modelling the GPS-receiver: a look from the LBS-side. In *Proceedings of the European Navigation Conference (ENC-GNSS)* (Toulouse, April 2008).
- [37] DEMYEN, D., BURO, M. Fast Pathfinding Based on Triangulation Abstractions. In *AI Game Programming Wisdom 4*, S. Rabin, Ed. Course Technology, 2008, S. 143–157.
- [38] DIJKSTRA, E. W. A note on two problems in connexion with graphs. *Numerische Mathematik 1* (1959), S. 269–271.
- [39] DOBSON, S. Leveraging the subtleties of location. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies* (New York, NY, USA, 2005), ACM, S. 189–193.
- [40] DOMNITCHEVA, S. Location Modeling: State of the Art and Challenges. In *Proceedings of the Workshop on Location Modeling for Ubiquitous Computing* (Atlanta, September 2001).
- [41] DÜRR, F., ROTHERMEL, K. On a Location Model for Fine-Grained Geocast. In *Proceedings of the 5th International Conference on Ubiquitous Computing (UbiComp)* (Seattle, WA, USA, October 2003).
- [42] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. TS 102 190: Near Field Communication (NFC) IP-1; Interface and Protocol (NFCIP-1) V1.1.1 (2003-03).
- [43] FARNSTROM, F. Improving on Near-Optimality: More Techniques for Building Navigation Meshes. In *AI Game Programming Wisdom 3*, S. Rabin, Ed. Charles River Media, 2006, S. 113–128.
- [44] FELKEL, P., OBRZALEK, S. Straight Skeleton Implementation. In *Proceedings of Spring Conference on Computer Graphics* (1998), S. 210–218.
- [45] FERNANDEZ, J., GONZALEZ, J. Hierarchical graph search for mobile robot path planning. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on* (1998), vol. 1, S. 656–661.
- [46] FINKEL, R. A., BENTLEY, J. L. Quad trees a data structure for retrieval on composite keys. *Acta Informatica 4* (1974), S. 1–9.

Literaturverzeichnis

- [47] FISCHER, C., GELLERSEN, H. Location and Navigation Support for Emergency Responders: A Survey. *IEEE Pervasive Computing* 9 (2010), S. 38–47.
- [48] FLOYD, R. W. Algorithm 97: Shortest path. *Commun. ACM* 5 (June 1962), S. 345.
- [49] FORTUNE, S. Voronoi diagrams and Delaunay triangulations. In *Computing in Euclidean Geometry* (1992), S. 193–233.
- [50] GEZICI, S., TIAN, Z., GIANNAKIS, G. B., KOBAYASHI, H., MOLISCH, A. F. Localization via Ultra-Wideband Radios. *IEEE Signal Processing Magazine* 22 (2005), S. 70–84.
- [51] GSCHWANDTNER, F. WLAN-Ortung am Flughafen München. Diplomarbeit, Ludwig-Maximilians-Universität München, Dezember 2007.
- [52] GSCHWANDTNER, F., RUPPEL, P. Automatic Calibration of Large-scale WiFi Positioning Systems. In *IEEE Global Communications Conference - Design & Developers Forum* (December 2008), IEEE.
- [53] GSM ASSOCIATION. Location-based Services. Permanent Reference Document: SE.23; Version 3.1.0, Januar 2003.
- [54] HAMM, D. Navigation Mesh Generation: An Empirical Approach. In *AI Game Programming Wisdom 4*, S. Rabin, Ed. Course Technology, 2008, S. 113–123.
- [55] HART, P. E., NILSSON, N. J., RAPHAEL, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems, Science, and Cybernetics SSC-4*, 2 (1968), S. 100–107.
- [56] HEIDMANN, F., HERMANN, F. Benutzerzentrierte Visualisierung raumbezogener Informationen für ultraportable mobile Systeme. In *Kartographische Schriften Band 7: Visualisierung und Erschließung von Geodaten*. Deutsche Gesellschaft für Kartographie, 2003, S. 121–131.
- [57] HERRMANN, T., SCHWEIZER, K., JANZEN, G., KATZ, S. Routen- und Überblickswissen - konzeptuelle Überlegungen. *Kognitionswissenschaft* 7, 4 (1998), S. 145–159.
- [58] HIGHTOWER, J., BORRIELLO, G. Location Systems for Ubiquitous Computing. *IEEE Computer* 34 (2001), S. 57–66.
- [59] HOFFMANN-WELLENHOF, B., LICHTENEGGER, H., COLLINS, J. *GPS - Theory and Practice*. Springer, 2001.
- [60] HÖLSCHER, C., BÜCHNER, S. J., BRÖSAMLE, M., MEILINGER, T., STRUBE, G. Signs and Maps: Cognitive Economy in the Use of External Aids for Indoor Navigation. In *29th Annual Conference of the Cognitive Science Society* (08 2007), S. 377–382.

Literaturverzeichnis

- [61] HU, H., LEE, D.-L. Semantic Location Modeling for Location Navigation in Mobile Environment. In *Proceedings of the IEEE International Conference on Mobile Data Management* (2004), S. 52–61.
- [62] HUANG, H., GARTNER, G. A Survey of Mobile Indoor Navigation Systems. In *Cartography in Central and Eastern Europe*, G. Gartner und F. Ortog, Eds., Lecture Notes in Geoinformation and Cartography. Springer Berlin Heidelberg, 2010, S. 305–319.
- [63] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC. Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2007.
- [64] JAVA COMMUNITY PROCESS. Location API for Java 2 Micro Edition Version 1.0.1. <http://jcp.org/en/jsr/detail?id=179>. Zuletzt besucht: 05.02.11.
- [65] JENI, M. Multimodale kontextsensitive Routenberechnung. Diplomarbeit, Ludwig-Maximilians-Universität München, Januar 2009.
- [66] JIANG, C., STEENKISTE, P. A Hybrid Location Model with a Computable Location Identifier for Ubiquitous Computing. In *Proceedings of the 4th international conference on Ubiquitous Computing* (London, UK, 2002), Springer-Verlag, S. 246–263.
- [67] JOHNSON, D. B. Efficient Algorithms for Shortest Paths in Sparse Networks. *J. ACM* 24 (Januar 1977), S. 1–13.
- [68] JONES, S. Toward an Acceptable Definition of Service. *IEEE Software* 22 (2005), S. 87–93.
- [69] KAINZ, W., EGENHOFER, M. J., GREASLEY, I. Modeling spatial relations and operations with partially ordered sets. *International Journal of Geographic Information Systems* 7, 3 (1993), S. 215–229.
- [70] KARGL, F., GESSLER, S., FLERLAGE, F. The iNAV Indoor Navigation System. In *Ubiquitous Computing Systems*, H. Ichikawa, W.-D. Cho, I. Satoh, und H. Youn, Eds., vol. 4836 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2007, S. 110–117.
- [71] KATO, H., TAN, K. Pervasive 2D Barcodes for Camera Phone Applications. *Pervasive Computing, IEEE* 6, 4 (Oct.-Dec. 2007), S. 76–85.
- [72] KEIL, J. M. Decomposing a Polygon into Simpler Components. *SIAM Journal on Computing* 14, 4 (1985), S. 799–817.

Literaturverzeichnis

- [73] KESSEL, M. Erzeugung von Umgebungsmodellen aus digitalen Gebäudeplänen für Ortsbezogene Dienste. Diplomarbeit, Ludwig-Maximilians-Universität München, Januar 2010.
- [74] KESSEL, M., RUPPEL, P., GSCHWANDTNER, F. BIGML: A Location Model with Individual Waypoint Graphs for Indoor Location-based Services. *PIK - Praxis der Informationsverarbeitung und Kommunikation* 13 (2010), S. 260–266.
- [75] KESSEL, M., RUPPEL, P., GSCHWANDTNER, F. Waypoint Graph Creation for Indoor Location-based Services. In *Tagungsband zum 7. GI/KuVS-Fachgespräch „Ortsbasierte Anwendungen und Dienste“* (2010), Logos-Verlag, Berlin.
- [76] KJÆRGAARD, M. Minimizing the Power Consumption of Location-Based Services on Mobile Phones. *IEEE Pervasive Computing* 99, PrePrints (2010).
- [77] KJÆRGAARD, M., TREU, G., LINNHOFF-POPIEN, C. Zone-Based RSS Reporting for Location Fingerprinting. In *Pervasive Computing*, A. LaMarca, M. Langheinrich, und K. Truong, Eds., vol. 4480 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2007, S. 316–333.
- [78] KJÆRGAARD, M. B. Indoor location fingerprinting with heterogeneous clients. *Pervasive and Mobile Computing In Press, Corrected Proof* (2010).
- [79] KJÆRGAARD, M. B., TREU, G., RUPPEL, P., KÜPPER, A. Efficient Indoor Proximity and Separation Detection for Location Fingerprinting. In *Mobilware 2008, First International Conference on MOBILE Wireless MiddleWARE, Operating Systems, and Applications* (Innsbruck, Austria, 2008).
- [80] KOLBE, T. H. Augmented Videos and Panoramas for Pedestrian Navigation. In *Proceedings of the 2nd Symposium on Location Based Services and TeleCartography* (2004).
- [81] KOLBE, T. H., MIDDEL, A., PLÜMER, L. 3D-Kartographie für die Fußgänger-navigation: Virtuelle Wegweiser in Panoramen. *Mitteilungen des Bundesamtes für Kartographie und Geodäsie* 31 (September 2004).
- [82] KOLODZIEJ, K., DANADO, J. In-Building Positioning: Modeling Location for Indoor World. In *Proceedings of the 15th International Workshop on Database and Expert Systems Applications* (Los Alamitos, CA, USA, 2004), IEEE Computer Society, S. 830–834.
- [83] KOLODZIEJ, K. W., HJELM, J. *Local Positioning Systems*. CRC Taylor and Francis, 2006.
- [84] KÜPPER, A. *Location-based Services - Fundamentals and Operation*. John Wiley and Sons, 2005.

Literaturverzeichnis

- [85] KÜPPER, A., TREU, G., LINNHOFF-POPIEN, C. TraX: A Device-Centric Middleware Framework for Location-Based Services. *IEEE Communications Magazine* 44, 9 (2006), S. 114–120.
- [86] KRAY, C., CHEVERST, K., HARRISON, M., HAMHOUM, F., WAGNER, J. Towards a location model for indoor navigation support through public displays and mobile devices. In *Proceedings of The Mobile Interaction with the Real World (MIRW) workshop* (2008).
- [87] KRAY, C., ELTING, C., LAAKSO, K., COORS, V. Presenting route instructions on mobile devices. In *Proceedings of the 8th international conference on Intelligent user interfaces* (New York, NY, USA, 2003), S. 117–124.
- [88] KRAY, C., KORTUEM, G., KRÜGER, A. Adaptive navigation support with public displays. In *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces* (New York, NY, USA, 2005), ACM, S. 326–328.
- [89] KRÜGER, A., BUTZ, A., MÜLLER, C., STAHL, C., WASINGER, R., STEINBERG, K.-E., DIRSCHL, A. The connected user interface: realizing a personal situated navigation service. In *Proceedings of the 9th international conference on Intelligent user interfaces* (New York, NY, USA, 2004), S. 161–168.
- [90] KÜPPER, A., TREU, G. Efficient proximity and separation detection among mobile targets for supporting location-based community services. *SIGMOBILE Mob. Comput. Commun. Rev.* 10, 3 (2006), S. 1–12.
- [91] LEMELSON, H., KJÆRGAARD, M. B., HANSEN, R., KING, T. Error Estimation for Indoor 802.11 Location Fingerprinting. In *Proceedings of the 4th International Symposium on Location and Context Awareness* (Berlin, Heidelberg, 2009), Springer-Verlag, S. 138–155.
- [92] LEONHARDT, U. *Supporting Location-awareness in Open Distributed Systems*. Dissertation, University of London, 1998.
- [93] LI, D., LEE, D. L. A Lattice-Based Semantic Location Model for Indoor Navigation. In *Proceedings of The IEEE International Conference on Mobile Data Management* (Los Alamitos, CA, USA, 2008), IEEE Computer Society, S. 17–24.
- [94] LIEBICH, T., ADACHI, Y., FORESTER, J., HYVARINEN, J., RICHTER, S., CHIPMAN, T., WEISE, M., WIX, J. Industry Foundation Classes IFC2x Edition 4 Release Candidate 2. <http://www.iai-tech.org/ifc/IFC2x4/rc2/html/>. Zuletzt besucht: 05.02.11.
- [95] LIJDING, M., BENZ, H., MERATNIA, N., HAVINGA, P. Smart Signs: Showing the way in Smart Surroundings. Technischer Bericht TR-CTIT-06-20, University of Twente, April 2006.

Literaturverzeichnis

- [96] LOVELACE, K. L., HEGARTY, M., MONTELLO, D. R. Elements of Good Route Directions in Familiar and Unfamiliar Environments. In *Proceedings of the International Conference on Spatial Information Theory: Cognitive and Computational Foundations of Geographic Information Science* (London, UK, 1999), Springer-Verlag, S. 65–82.
- [97] MAGUIRE, E. A., BURGESS, N., DONNETT, J. G., FRACKOWIAK, R. S. J., FRITH, C. D., O’KEEFE, J. Knowing Where and Getting There: A Human Navigation Network. *Science* 280, 5365 (1998), S. 921–924.
- [98] MAGUIRE, E. A., BURGESS, N., O’KEEFE, J. Human spatial navigation: cognitive maps, sexual dimorphism, and neural substrates. *Current Opinion in Neurobiology* 9 (1999), S. 171–177.
- [99] MEILINGER, T., HÖLSCHER, C., BÜCHNER, S., BRÖSAMLE, M. How Much Information Do You Need? Schematic Maps in Wayfinding and Self Localisation. In *Spatial Cognition V Reasoning, Action, Interaction*, vol. 4387 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2007, S. 381–400.
- [100] MÜNZER, S., STAHL, C. Providing Individual Route Instructions for Indoor Wayfinding in Complex, Multi-Level Buildings. In *Proceedings of the 5th Geographic Information Days* (2007), S. 241–246.
- [101] MOFFAT, S. D., HAMPSON, E., HATZIPANTELIS, M. Navigation in a Virtual Maze: Sex Differences and Correlation With Psychometric Measures of Spatial Ability in Humans. *Evolution and Human Behavior* 19, 2 (1998), S. 73 – 87.
- [102] MULLONI, A., WAGNER, D., BARAKONYI, I., SCHMALSTIEG, D. Indoor Positioning and Navigation with Camera Phones. *IEEE Pervasive Computing* 8, 2 (2009), S. 22–31.
- [103] NICKLAS, D. *Ein umfassendes Umgebungsmodell als Integrationsstrategie für ortsbezogene Daten und Dienste*. Dissertation, Universität Stuttgart, 2005.
- [104] OBELLEIRO, J., SAMPEDRO, R., CERPA, D. H. RTS Terrain Analysis: An Image-Processing Approach. In *AI Game Programming Wisdom 4*, S. Rabin, Ed. Course Technology, 2008, S. 361–372.
- [105] O’NEILL, M. J. Effects of Signage and Floor Plan Configuration on Wayfinding Accuracy. *Environment and Behavior* 23, 5 (1991), S. 553–574.
- [106] OPEN GEOSPATIAL CONSORTIUM INC. OpenGIS® Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture, 2006.
- [107] OPEN GEOSPATIAL CONSORTIUM INC. OpenGIS® Geography Markup Language (GML) Encoding Standard, August 2007.

Literaturverzeichnis

- [108] OPEN GEOSPATIAL CONSORTIUM INC. OGC KML Version 2.2.0, April 2008.
- [109] OPEN GEOSPATIAL CONSORTIUM INC. OpenGIS® City Geography Markup Language (CityGML) Encoding Standard, August 2008.
- [110] OPEN GEOSPATIAL CONSORTIUM INC. OpenGIS® Location Services (OpenLS): Core Services, 2008.
- [111] OPEN GEOSPATIAL CONSORTIUM INC. OpenGIS® LocationServices (OpenLS): Part 6-Navigation Service, 2008.
- [112] OPEN GEOSPATIAL CONSORTIUM INC. OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 2: SQL option, 2010.
- [113] OPEN MOBILE ALLIANCE. Mobile Location Protocol (MLP), 2004.
- [114] O’ROURKE, J. *Computational Geometry in C*. Cambridge University Press, 1998.
- [115] PETERNIER, A., RIGHETTI, X., HOPMANN, M., THALMANN, D., REPETTOY, M., PAPAGIANNAKIS, G., DAVY, P., LIM, M., MAGNENAT-THALMANN, N., BAR-SOCCHI, P., FRAGOPOULOS, T., SERPANOS, D., GIALELIS, Y., KIRYKOU, A. Chloe@University: an indoor, mobile mixed reality guidance system. In *Proceedings of the 2007 ACM symposium on Virtual reality software and technology* (New York, NY, USA, 2007), S. 227–228.
- [116] RAN, L., HELAL, S., MOORE, S. Drishti: An Integrated Indoor/Outdoor Blind Navigation System and Service. In *Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications* (März 2004), S. 23–30.
- [117] RANGANATHAN, A., AL-MUHTADI, J., CHETAN, S., CAMPBELL, R., MICKUNAS, M. D. MiddleWhere: A Middleware for Location Awareness in Ubiquitous Computing Applications. In *Middleware ’04: Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware* (New York, NY, USA, 2004), Springer-Verlag New York, Inc., S. 397–416.
- [118] RAUBAL, M., WINTER, S. Enriching Wayfinding Instructions with Local Landmarks. In *Geographic Information Science - Second International Conference, GIScience 2002, Boulder, CO, USA, September 2002* (Berlin, 2002), M. Egenhofer und D. Mark, Eds., vol. 2478 of *Lecture Notes in Computer Science*, Springer, S. 243–259.
- [119] REDISH, A. D. *Beyond the cognitive map: from place cells to episodic memory*. MIT Press, 1999.
- [120] REHRL, K., GÖLL, N., LEITINGER, S., BRUNTSCH, S. Combined indoor/outdoor Smartphone navigation for public transport travellers. In *Proceedings of the 3rd Symposium on LBS & TeleCartography. Geowissenschaftliche Mitteilungen Nr. 74* (Austria, 2005).

Literaturverzeichnis

- [121] ROTH, J. A Decentralized Location Service Providing Semantic Locations. Habilitationsschrift, FernUniversität Hagen, 2004.
- [122] ROTHERMEL, K., ERTL, T., DIETER, F., KÜHN, P. J., MITSCHANG, B., WESTKÄMPER, E., BECKER, C., DUDKOWSKI, D., GUTSCHER, A., HAUSER, C., JENDOUBI, L., NICKLAS, D., VOLZ, S., WIELAND, M. SFB 627 - Umgebungsmodelle für mobile kontextbezogene Systeme. *Informatik - Forschung und Entwicklung* 21, 1-2 (Juni 2006), S. 105–113.
- [123] RUDDLE, R. A., PAYNE, S. J., JONES, D. M. Navigating Buildings in 'Desk-Top' Virtual Environments: Experimental Investigations Using Extended Navigational Experience. *Journal of Experimental Psychology* 3, 2 (1997), S. 143–159.
- [124] RUKZIO, E., SCHMIDT, A., KRÜGER, A. The rotating compass: a novel interaction technique for mobile navigation. In *CHI '05 extended abstracts on Human factors in computing systems* (New York, NY, USA, 2005), S. 1761–1764.
- [125] RUPPEL, P., GSCHWANDTNER, F. Spontaneous and Privacy-friendly Mobile Indoor Routing and Navigation. In *Proceedings of the 2nd Workshop on Services, Platforms, Innovations and Research for new Infrastructures in Telecommunications* (Lübeck, Germany, 2009).
- [126] RUPPEL, P., GSCHWANDTNER, F., SCHINDHELM, C. K., LINNHOF-POPIEN, C. Indoor Navigation on Distributed Stationary Display Systems. In *Proceedings of the 33rd Annual IEEE International Computer Software and Applications Conference* (July 2009).
- [127] RUPPEL, P., KLEIN, C., LINNHOF-POPIEN, C. Indooria – A Platform for Proactive Indoor Location-based Services. In *IEEE Global Communications Conference - Design & Developers Forum* (December 2008), IEEE.
- [128] RUPPEL, P., TREU, G., KÜPPER, A., LINNHOF-POPIEN, C. Anonymous User Tracking for Location-Based Community Services. In *Location- and Context-Awareness*, vol. 3987 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2006, S. 116–133.
- [129] SANDSTROM, N. J., KAUFMAN, J., HUETTEL, S. A. Males and females use different distal cues in a virtual environment navigation task. *Cognitive Brain Research* 6, 4 (1998), S. 351 – 360.
- [130] SCHILIT, B., THEIMER, M. Disseminating active map information to mobile hosts. *IEEE Network* 8, 5 (Sep/Oct 1994), S. 22–32.
- [131] SCHINDHELM, C. K. Indoor Navigation mittels verteilter Anzeigesysteme. Diplomarbeit, Ludwig-Maximilians-Universität München, November 2008.

Literaturverzeichnis

- [132] SHAFER, S. A Framework for Creating and Using Maps of Privately Owned Spaces. In *Location and Context Awareness*, vol. 5561 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2009, S. 174–191.
- [133] STAHL, C., HAUPERT, J. Taking Location Modelling to New Levels: A Map Modelling Toolkit for Intelligent Environments. Technischer bericht, Saarland University, 2006.
- [134] STANTCHEV, V., SCHULZ, T., HOANG, T. D., RATCHINSKI, I. Optimizing Clinical Processes with Position-Sensing. *IT Professional* 10, 2 (März-April 2008), S. 31–37.
- [135] STENTZ, A. Optimal and Efficient Path Planning for Partially-Known Environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '94)* (May 1994), vol. 4, S. 3310 – 3317.
- [136] STENTZ, A. The Focussed D* Algorithm for Real-Time Replanning. In *Proceedings of the International Joint Conference on Artificial Intelligence* (August 1995).
- [137] STEVENSON, G., YE, J., DOBSON, S., NIXON, P. LOC8: A Location Model and Extensible Framework for Programming with Location. *IEEE Pervasive Computing* 9 (2010), S. 28–37.
- [138] STOFFEL, E.-P., LORENZ, B., OHLBACH, H. Towards a Semantic Spatial Model for Pedestrian Indoor Navigation. In *Advances in Conceptual Modeling - Foundations and Applications*, vol. 4802 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2007, S. 328–337.
- [139] STOFFEL, E.-P., SCHODER, K., OHLBACH, H. J. Applying hierarchical graphs to pedestrian indoor navigation. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems* (New York, NY, USA, 2008), GIS '08, ACM, S. 54:1–54:4.
- [140] STRANG, T., LINNHOFF-POPIEN, C. A Context Modeling Survey. In *Proceedings of the Workshop on Advanced Context Modelling, Reasoning and Management* (Nottingham/England, September 2004).
- [141] TOR, S. B., MIDDLEDITCH, A. E. Convex Decomposition of Simple Polygons. *ACM Trans. Graph.* 3, 4 (1984), S. 244–265.
- [142] TOZOUR, P. Building a Near-Optimal Navigation Mesh. In *AI Game Programming Wisdom*, S. Rabin, Ed. Charles River Media, 2002, S. 171–185.
- [143] TOZOUR, P. Search Space Representations. In *AI Game Programming Wisdom 2*, S. Rabin, Ed. Charles River Media, 2004, S. 85–102.
- [144] TOZOUR, P. Pathfinding Algorithms & Search Space Representations Demo. <http://www.ai-blog.net/archives/000091.html>, 2005. Zuletzt besucht: 23.01.11.

Literaturverzeichnis

- [145] TREU, G. *Position Management für ortsbezogene Community-Dienste*. Dissertation, Ludwig-Maximilians-Universität München, 2007.
- [146] TREU, G., RUPPEL, P., KÜPPER, A. Anonymization in Proactive Location Based Community Services. In *Advances in Pervasive Computing. Adjunct Proceedings of the Third International Conference on Pervasive Computing* (Mai 2005), Oesterreichische Computer Gesellschaft.
- [147] TSETSOS, V., ANAGNOSTOPOULOS, C., KIKIRAS, P., HASIOTIS, P., HADJIEFT-HYMIADES, S. A human-centered semantic navigation system for indoor environments. In *Proceedings of the International Conference on Pervasive Services* (Los Alamitos, CA, USA, 2005), S. 146–155.
- [148] WAHLSTER, W., BAUS, J., KRAY, C., KRÜGER, A. REAL: Ein ressourcenadaptierendes mobiles Navigationssystem. *Informatik - Forschung und Entwicklung 16* (2001), S. 233–241.
- [149] WANT, R., HOPPER, A., FALCÃO, V., GIBBONS, J. The active badge location system. *ACM Trans. Inf. Syst. 10* (Januar 1992), S. 91–102.
- [150] WHITE, M. S. Technical Requirements and Standards for a Multipurpose Geographic Data System. *Cartography and Geographic Information Science 11*, 1 (April 1984), S. 15–26.
- [151] WHITING, E., BATTAT, J., TELLER, S. Topology of Urban Environments. In *Proceedings of the 12th International CAAD Futures Conference* (2007), Springer.
- [152] WORLD WIDE WEB CONSORTIUM (W3C). Resource Description Framework (RDF) Model and Syntax Specification, Februar 1999.
- [153] WORLD WIDE WEB CONSORTIUM (W3C). XML Schema Part 1: Structures Second Edition. <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/structures.html>, Oktober 2004. Zuletzt besucht: 05.02.2011.
- [154] WORLD WIDE WEB CONSORTIUM (W3C). XML Schema Part 2: Datatypes Second Edition. <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/datatypes.html>, Oktober 2004. Zuletzt besucht: 05.02.2011.
- [155] XU, Z., JACOBSEN, H.-A. Efficient constraint processing for location-aware computing. In *Proceedings of the 6th international conference on Mobile data management* (New York, NY, USA, 2005), ACM, S. 3–12.
- [156] XU, Z., JACOBSEN, H.-A. Adaptive location constraint processing. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data* (New York, NY, USA, 2007), ACM, S. 581–592.

Literaturverzeichnis

- [157] YE, J., COYLE, L., DOBSON, S., NIXON, P. A Unified Semantics Space Model. In *Location- and Context-Awareness*, vol. 4718 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2007, S. 103–120.
- [158] YOUSSEF, M., AGRAWALA, A. The Horus WLAN location determination system. In *MobiSys '05: Proceedings of the 3rd international conference on Mobile systems, applications, and services* (New York, NY, USA, 2005), ACM, S. 205–218.

A. Building Information Geography Markup Language

```
1 <?xml version="1.0" encoding="UTF-8"?>
  <schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:bim="http://www.mobile.ifi.lmu.de/schemas/bigml" xmlns:xAL="urn:oasis:names:tc:ciq:xsd:schema:xAL:2.0" xmlns:Q1="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.mobile.ifi.lmu.de/schemas/bigml" elementFormDefault="qualified" attributeFormDefault="unqualified" version="01.03.2010">
3   <import namespace="http://www.opengis.net/gml/3.2" schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
   <import namespace="urn:oasis:names:tc:ciq:xsd:schema:xAL:2.0" schemaLocation="http://docs.oasis-open.org/election/external/xAL.xsd"/>
5   <element name="AbstractBuildingInformation" type="bim:AbstractBuildingInformationType" abstract="true" substitutionGroup="gml:AbstractFeature"/>
   <complexType name="AbstractBuildingInformationType" abstract="true">
7     <complexContent>
       <extension base="gml:AbstractFeatureType">
9         <sequence>
           <element name="propertyTag" type="bim:PropertyTagType" minOccurs="0" maxOccurs="unbounded">
11             <annotation>
               <documentation>Any additional property not covered by the other attributes.</documentation>
13             </annotation>
           </element>
15         </sequence>
       </extension>
17     </complexContent>
   </complexType>
19   <complexType name="AbstractBuildingInformationPropertyType">
     <sequence minOccurs="0">
21       <element ref="bim:AbstractBuildingInformation"/>
     </sequence>
23     <attributeGroup ref="gml:AssociationAttributeGroup"/>
   </complexType>
25   <element name="Site" type="bim:SiteType" substitutionGroup="bim:AbstractBuildingInformation"/>
   <complexType name="SiteType">
27     <complexContent>
       <extension base="bim:AbstractBuildingInformationType">
29         <sequence>
           <element name="class" type="bim:SiteClassType" minOccurs="0"/>

```

A. Building Information Geography Markup Language

```
31 <element name="usage" type="bim:SiteUsageType" minOccurs="0" maxOccurs="unbounded"/
   >
32 <element name="owner" type="string" minOccurs="0">
33 <annotation>
34 <documentation>Owner of the site. </documentation>
35 </annotation>
36 </element>
37 <element name="URI" type="anyURI">
38 <annotation>
39 <documentation>Uniform identifier for this site. Should be globally unique.</
   documentation>
40 </annotation>
41 </element>
42 <element name="website" type="anyURI" minOccurs="0" maxOccurs="unbounded">
43 <annotation>
44 <documentation>Websites connected with the site or the owner of the site.</
   documentation>
45 </annotation>
46 </element>
47 <element name="siteEntry" type="bim:SiteEntryPropertyType" maxOccurs="unbounded">
48 <annotation>
49 <documentation>Entry to the site.</documentation>
50 </annotation>
51 </element>
52 <element name="address" type="bim:AddressPropertyType" minOccurs="0" maxOccurs="
   unbounded">
53 <annotation>
54 <documentation>Adress of the site conform to the OASIS specification xAl.</
   documentation>
55 </annotation>
56 </element>
57 <element name="buildingPortal" type="bim:BuildingPortalPropertyType" maxOccurs="
   unbounded">
58 <annotation>
59 <documentation>All portals leading into the buildings on this site.</
   documentation>
60 </annotation>
61 </element>
62 <element name="outline" type="gml:PolygonType">
63 <annotation>
64 <documentation>Polygon representing the outline of this site.</documentation>
65 </annotation>
66 </element>
67 <element name="localCS" type="gml:CartesianCSPropertyType">
68 <annotation>
69 <documentation>Local Cartesian Coordinate System. All geometric positions are
   given in this coordinate system.</documentation>
70 </annotation>
71 </element>
<element name="globalCRSReferencePoint" type="
   bim:GlobalCRSReferencePointPropertyType" minOccurs="0" maxOccurs="unbounded">
```

A. Building Information Geography Markup Language

```
73     <annotation>
74         <documentation>For coordinate transformation. Gives the position of the same
75             point in the local coordinate system and in a well known global coordinate
76             reference system.</documentation>
77     </annotation>
78 </element>
79 <element name="graph" type="bim:GraphPropertyType">
80     <annotation>
81         <documentation>The graph represents the connectivity on the whole site and all
82             contained buildings.</documentation>
83     </annotation>
84 </element>
85 <element name="building" type="bim:BuildingPropertyType" maxOccurs="unbounded">
86     <annotation>
87         <documentation>Buildings contained in this site.</documentation>
88     </annotation>
89 </element>
90 </sequence>
91 </extension>
92 </complexContent>
93 </complexType>
94 <complexType name="SitePropertyType">
95     <sequence minOccurs="0">
96         <element ref="bim:Site"/>
97     </sequence>
98     <attributeGroup ref="gml:AssociationAttributeGroup"/>
99 </complexType>
100 <element name="Building" type="bim:BuildingType" substitutionGroup="
101     bim:AbstractBuildingInformation"/>
102 <complexType name="BuildingType">
103     <complexContent>
104         <extension base="bim:AbstractBuildingInformationType">
105             <sequence>
106                 <element name="class" type="bim:BuildingClassType" minOccurs="0">
107                     <annotation>
108                         <documentation>Code list derived from German authoritative standards ALKIS/ATKIS
109                             (www.adv-online.de)</documentation>
110                     </annotation>
111                 </element>
112                 <element name="usage" type="bim:BuildingUsageType" minOccurs="0" maxOccurs="
113                     unbounded">
114                     <annotation>
115                         <documentation>Code list derived from German authoritative standards ALKIS/ATKIS
116                             (www.adv-online.de)</documentation>
117                     </annotation>
118                 </element>
119                 <element name="storeysAboveGround" type="nonNegativeInteger" minOccurs="0"/>
120                 <element name="storeysBelowGround" type="nonNegativeInteger" minOccurs="0"/>
121                 <element name="height" type="gml:LengthType" minOccurs="0">
122                     <annotation>
```

A. Building Information Geography Markup Language

```

    <documentation>Height of the building, including the roofstructure.</
      documentation>
117  </annotation>
    </element>
119  <element name="depth" type="gml:LengthType" minOccurs="0">
    <annotation>
121    <documentation>How far reaches the building into the ground.</documentation>
    </annotation>
123  </element>
    <element name="heightAboveNN" type="gml:LengthType" minOccurs="0">
125    <annotation>
      <documentation>Hight above sea level, according to the assigned geodic datum.</
        documentation>
127    </annotation>
    </element>
129  <element name="yearOfConstruction" type="gYear" minOccurs="0">
    <annotation>
131    <documentation>The year the building was constructed.</documentation>
    </annotation>
133  </element>
    <element name="isOn" type="bim:SitePropertyType">
135    <annotation>
      <documentation>Reference to the site the building is on.</documentation>
137    </annotation>
    </element>
139  <element name="outline" type="gml:PolygonType">
    <annotation>
141    <documentation>Polygon representing the outline of this building.</documentation>
    </annotation>
143  </element>
    <element name="floor" type="bim:FloorPropertyType" maxOccurs="unbounded">
145    <annotation>
      <documentation>Reference to all floors the building consists of.</documentation>
147    </annotation>
    </element>
149  </sequence>
    </extension>
151  </complexContent>
  </complexType>
153  <complexType name="BuildingPropertyType">
    <sequence minOccurs="0">
155    <element ref="bim:Building"/>
    </sequence>
157    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
159  <element name="Floor" type="bim:FloorType" substitutionGroup="
    bim:AbstractBuildingInformation"/>
  <complexType name="FloorType">
161  <complexContent>
    <extension base="bim:AbstractBuildingInformationType">
163    <sequence>
```

A. Building Information Geography Markup Language

```
165 <element name="building" type="bim:BuildingPropertyType">
  <annotation>
    <documentation>Reference to the buildings this floor is in.</documentation>
  </annotation>
167 </element>
169 <element name="heightAboveGround" type="gml:LengthType">
  <annotation>
    <documentation>The height of the floor above ground level.</documentation>
171 </annotation>
173 </element>
175 <element name="level" type="int">
  <annotation>
    <documentation>An integer indicating the floor level. The ground level has the
      number zero.</documentation>
177 </annotation>
179 </element>
181 <element name="outline" type="gml:PolygonType">
  <annotation>
    <documentation>Polygon representing the outline of this floor.</documentation>
183 </annotation>
185 </element>
187 <element name="room" type="bim:RoomPropertyType" maxOccurs="unbounded">
  <annotation>
    <documentation>Reference to all rooms on this floor.</documentation>
189 </annotation>
191 </element>
193 </sequence>
195 </extension>
197 </complexContent>
199 </complexType>
201 <complexType name="FloorPropertyType">
  <sequence minOccurs="0">
    <element ref="bim:Floor"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
203 <element name="Room" type="bim:RoomType" substitutionGroup="
  bim:AbstractBuildingInformation"/>
<complexType name="RoomType">
205 <complexContent>
  <extension base="bim:AbstractBuildingInformationType">
207 <sequence>
  <element name="class" type="bim:RoomClassType" minOccurs="0">
    <annotation>
      <documentation>Code list proposed by the SIG 3D</documentation>
    </annotation>
  </element>
209 <element name="usage" type="bim:RoomUsageType" minOccurs="0" maxOccurs="unbounded">
    <annotation>
      <documentation>Code list proposed by the SIG 3D</documentation>
    </annotation>
211 </element>
```

A. Building Information Geography Markup Language

```
213 </element>
214 <element name="heightOfCeiling" type="gml:LengthType" minOccurs="0">
215 <annotation>
216 <documentation>Height of the ceiling in this room, measured from floor to ceiling
217 .</documentation>
218 </annotation>
219 </element>
220 <element name="floor" type="bim:FloorPropertyType">
221 <annotation>
222 <documentation>Reference to the floor this room is on.</documentation>
223 </annotation>
224 </element>
225 <element name="outline" type="gml:PolygonType">
226 <annotation>
227 <documentation>Polygon describing the outline of the room.</documentation>
228 </annotation>
229 </element>
230 <element name="poi" type="bim:POIPropertyType" minOccurs="0" maxOccurs="unbounded">
231 <annotation>
232 <documentation>Points of Interest in this room.</documentation>
233 </annotation>
234 </element>
235 <element name="aoi" type="bim:AOIPropertyType" minOccurs="0" maxOccurs="unbounded">
236 <annotation>
237 <documentation>Areas of Interes in this room.</documentation>
238 </annotation>
239 </element>
240 <element name="portal" type="bim:PortalPropertyType" minOccurs="0" maxOccurs="
241 unbounded">
242 <annotation>
243 <documentation>Portals in this room.</documentation>
244 </annotation>
245 </element>
246 <element name="containedNode" type="bim:NodePropertyType" minOccurs="0" maxOccurs="
247 unbounded">
248 <annotation>
249 <documentation>The nodes of the graph which lie inside this room.</documentation>
250 </annotation>
251 </element>
252 </sequence>
253 </extension>
254 </complexContent>
255 </complexType>
256 <complexType name="RoomPropertyType">
257 <sequence minOccurs="0">
258 <element ref="bim:Room"/>
259 </sequence>
260 <attributeGroup ref="gml:AssociationAttributeGroup"/>
261 </complexType>
262 <element name="SiteEntry" type="bim:SiteEntryType" substitutionGroup="
263 bim:AbstractBuildingInformation"/>
```

A. Building Information Geography Markup Language

```
<complexType name="SiteEntryType">
261 <complexContent>
  <extension base="bim:AbstractBuildingInformationType">
263 <sequence>
  <element name="node" type="bim:NodePropertyType"/>
265 </sequence>
</extension>
267 </complexContent>
</complexType>
269 <complexType name="SiteEntryPropertyType">
  <sequence minOccurs="0">
271 <element ref="bim:SiteEntry"/>
  </sequence>
273 <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
275 <element name="AbstractPortal" type="bim:AbstractPortalType" abstract="true"
  substitutionGroup="bim:AbstractBuildingInformation"/>
<complexType name="AbstractPortalType" abstract="true">
277 <annotation>
  <documentation>This type describes all kind of portals, like doors or stairs. </
    documentation>
279 </annotation>
  <complexContent>
281 <extension base="bim:AbstractBuildingInformationType">
  <sequence>
283 <element name="angle" type="gml:AngleType" minOccurs="0"/>
  <element name="linksTo" type="bim:PortalPropertyType" maxOccurs="unbounded">
285 <annotation>
  <documentation>Destination of this portal</documentation>
287 </annotation>
  </element>
289 <element name="node" type="bim:NodePropertyType">
  <annotation>
291 <documentation>The associated node, representing the entrance point of the portal
    as part of the graph.</documentation>
  </annotation>
293 </element>
  <element name="belongsTo" type="bim:AbstractBuildingInformationPropertyType">
295 <annotation>
  <documentation>Either the site or the room the portal belongs to.</documentation>
297 </annotation>
  </element>
299 <element name="edge" type="bim:EdgePropertyType">
  <annotation>
301 <documentation>The edge of the graph crossing the portal.</documentation>
  </annotation>
303 </element>
  <element name="geometry" type="gml:AbstractRingPropertyType" minOccurs="0">
305 <annotation>
  <documentation>Any geometry describing the form or extent of the portal.</
    documentation>
```

A. Building Information Geography Markup Language

```
307     </annotation>
308     </element>
309 </sequence>
310 </extension>
311 </complexContent>
312 </complexType>
313 <complexType name="PortalPropertyType" abstract="0">
314   <sequence minOccurs="0">
315     <element ref="bim:AbstractPortal"/>
316   </sequence>
317   <attributeGroup ref="gml:AssociationAttributeGroup"/>
318 </complexType>
319 <element name="BuildingPortal" type="bim:BuildingPortalType" substitutionGroup="
    bim:AbstractPortal"/>
320 <complexType name="BuildingPortalType">
321   <complexContent>
322     <extension base="bim:AbstractPortalType"/>
323   </complexContent>
324 </complexType>
325 <complexType name="BuildingPortalPropertyType">
326   <sequence minOccurs="0">
327     <element ref="bim:BuildingPortal"/>
328   </sequence>
329   <attributeGroup ref="gml:AssociationAttributeGroup"/>
330 </complexType>
331 <element name="DoorPortal" type="bim:DoorPortalType" substitutionGroup="
    bim:AbstractPortal"/>
332 <complexType name="DoorPortalType">
333   <complexContent>
334     <extension base="bim:AbstractPortalType"/>
335   </complexContent>
336 </complexType>
337 <element name="ElevatorPortal" type="bim:ElevatorPortalType" substitutionGroup="
    bim:AbstractPortal"/>
338 <complexType name="ElevatorPortalType">
339   <complexContent>
340     <extension base="bim:AbstractPortalType"/>
341   </complexContent>
342 </complexType>
343 <element name="EscalatorPortal" type="bim:EscalatorPortalType" substitutionGroup="
    bim:AbstractPortal"/>
344 <complexType name="EscalatorPortalType">
345   <complexContent>
346     <extension base="bim:AbstractPortalType"/>
347   </complexContent>
348 </complexType>
349 <element name="RampPortal" type="bim:RampPortalType" substitutionGroup="
    bim:AbstractPortal"/>
350 <complexType name="RampPortalType">
351   <complexContent>
352     <extension base="bim:AbstractPortalType"/>
```

A. Building Information Geography Markup Language

```
353 </complexContent>
</complexType>
355 <element name="StairPortal" type="bim:StairPortalType" substitutionGroup="
    bim:AbstractPortal"/>
<complexType name="StairPortalType">
357 <complexContent>
    <extension base="bim:AbstractPortalType"/>
359 </complexContent>
</complexType>
361 <element name="POI" type="bim:POIType" substitutionGroup="
    bim:AbstractBuildingInformation"/>
<complexType name="POIType">
363 <complexContent>
    <extension base="bim:AbstractBuildingInformationType">
365 <sequence>
    <element name="class" type="bim:POIClassType" minOccurs="0"/>
367 <element name="URI" type="anyURI"/>
    <element name="description" type="string" minOccurs="0"/>
369 <element name="angle" type="gml:AngleType" minOccurs="0"/>
    <element name="node" type="bim:NodePropertyType">
371 <annotation>
    <documentation>Node of the graph this POI is on.</documentation>
373 </annotation>
    </element>
375 <element name="room" type="bim:RoomPropertyType">
    <annotation>
377 <documentation>Room this Poi is in.</documentation>
    </annotation>
379 </element>
    </sequence>
381 </extension>
</complexContent>
383 </complexType>
<complexType name="POIPropertyType">
385 <sequence minOccurs="0">
    <element ref="bim:POI"/>
387 </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
389 </complexType>
<element name="AOI" type="bim:AOIType" substitutionGroup="
    bim:AbstractBuildingInformation"/>
391 <complexType name="AOIType">
    <complexContent>
393 <extension base="bim:AbstractBuildingInformationType">
    <sequence>
395 <element name="class" type="bim:AOIClassType" minOccurs="0"/>
    <element name="usage" type="bim:AOIUsageType" minOccurs="0"/>
397 <element name="URI" type="anyURI">
    <annotation>
399 <documentation>Identificator for an AOI. All AOIs which share the same URI belong
        technically to the same area of interest. So it is possible to define AOIs
```

A. Building Information Geography Markup Language

```
        beyond the extent of a single room.</documentation>
    </annotation>
</element>
401 <element name="description" type="string" minOccurs="0"/>
403 <element name="outline" type="gml:PolygonType">
    <annotation>
405     <documentation>Polygon representing the outline of the AOI. Does not necessarily
        match the outline of the containing room.</documentation>
    </annotation>
</element>
407 <element name="room" type="bim:RoomPropertyType">
    <annotation>
409     <documentation>Room containing the AOI.</documentation>
    </annotation>
411 </element>
</sequence>
413 </extension>
</complexContent>
415 </complexType>
417 <complexType name="AOIPropertyType">
    <sequence minOccurs="0">
419     <element ref="bim:AOI"/>
    </sequence>
421 <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
423 <element name="Graph" type="bim:GraphType" substitutionGroup="gml:AbstractTopology"/>
<complexType name="GraphType">
425 <complexContent>
    <extension base="gml:AbstractTopologyType">
427 <sequence>
    <element name="node" type="bim:NodePropertyType" minOccurs="2" maxOccurs="unbounded
        "/>
429 <element name="edge" type="bim:EdgePropertyType" maxOccurs="unbounded"/>
    </sequence>
431 </extension>
</complexContent>
433 </complexType>
<complexType name="GraphPropertyType">
435 <sequence minOccurs="0">
    <element ref="bim:Graph"/>
437 </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
439 </complexType>
<element name="Edge" type="bim:EdgeType" substitutionGroup="gml:AbstractTopology"/>
441 <complexType name="EdgeType">
    <complexContent>
443 <extension base="gml:AbstractTopoPrimitiveType">
    <sequence>
445 <element name="node" type="bim:NodePropertyType" minOccurs="2" maxOccurs="2">
    <annotation>
447 <documentation>Each edge has exactly two nodes.</documentation>
```

A. Building Information Geography Markup Language

```
449     </annotation>
450   </element>
451   <element name="length" type="gml:LengthType">
452     <annotation>
453       <documentation>Length of the edge.</documentation>
454     </annotation>
455   </element>
456   <element name="propertyTag" type="bim:PropertyTagType" minOccurs="0" maxOccurs="
457     unbounded">
458     <annotation>
459       <documentation>Additional properties of the edge.</documentation>
460     </annotation>
461   </element>
462 </sequence>
463 </extension>
464 </complexContent>
465 </complexType>
466 <complexType name="EdgePropertyType">
467   <sequence minOccurs="0">
468     <element ref="bim:Edge"/>
469   </sequence>
470   <attributeGroup ref="gml:AssociationAttributeGroup"/>
471 </complexType>
472 <element name="Node" type="bim:NodeType" substitutionGroup="gml:AbstractTopology"/>
473 <complexType name="NodeType">
474   <complexContent>
475     <extension base="gml:AbstractTopoPrimitiveType">
476       <sequence>
477         <element name="edge" type="bim:EdgePropertyType" maxOccurs="unbounded"/>
478         <element name="position" type="gml:PointPropertyType">
479           <annotation>
480             <documentation>Geometric position of the node.</documentation>
481           </annotation>
482         </element>
483         <element name="POI" type="bim:POIPropertyType" minOccurs="0">
484           <annotation>
485             <documentation>Link to the POI if the node represents this POI in the graph.</
486               documentation>
487           </annotation>
488         </element>
489         <element name="portal" type="bim:PortalPropertyType" minOccurs="0">
490           <annotation>
491             <documentation>If this node is on a portal, it has a reference to the portal.</
492               documentation>
493           </annotation>
494         </element>
495         <element name="room" type="bim:RoomPropertyType" minOccurs="0">
496           <annotation>
497             <documentation>Nodes in rooms have a reference to that room.</documentation>
498           </annotation>
499         </element>
```

A. Building Information Geography Markup Language

```

    <element name="propertyTag" type="bim:PropertyTagType" minOccurs="0" maxOccurs="
        unbounded">
497     <annotation>
        <documentation>Additional properties of node.</documentation>
499     </annotation>
    </element>
501 </sequence>
    </extension>
503 </complexContent>
</complexType>
505 <complexType name="NodePropertyType">
    <sequence minOccurs="0">
507     <element ref="bim:Node"/>
    </sequence>
509 <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
511 <element name="Address" type="bim:AddressType" substitutionGroup="
    bim:AbstractBuildingInformation"/>
<complexType name="AddressType">
513 <complexContent>
    <extension base="bim:AbstractBuildingInformationType">
515 <sequence>
        <element name="address" type="xAL:AddressDetails">
517 <annotation>
            <documentation>Address conform to the xAL address standard issued by the OASIS
                consortium. </documentation>
519 </annotation>
        </element>
521 <element name="siteEntry" type="bim:SiteEntryPropertyType" maxOccurs="unbounded">
            <annotation>
523 <documentation>SiteEntries with the given address.</documentation>
            </annotation>
525 </element>
        </sequence>
527 </extension>
    </complexContent>
529 </complexType>
<complexType name="AddressPropertyType">
531 <sequence minOccurs="0">
    <element ref="bim:Address"/>
533 </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
535 </complexType>
<complexType name="PropertyTagType">
537 <annotation>
    <documentation>This type represents any property specified by the key - value
        attributes. No key should be used twice in a single BuildingInformationElement.</
        documentation>
539 </annotation>
    <attribute name="key" type="string" use="required"/>
541 <attribute name="value" type="string" use="required"/>

```

A. Building Information Geography Markup Language

```
543 </complexType>
544 <element name="GlobalCRSReferencePoint" type="bim:GlobalCRSReferencePointType"
545 substitutionGroup="bim:AbstractBuildingInformation"/>
546 <complexType name="GlobalCRSReferencePointType">
547 <complexContent>
548 <extension base="bim:AbstractBuildingInformationType">
549 <sequence>
550 <element name="localPoint" type="gml:PointPropertyType">
551 <annotation>
552 <documentation>Coordinates of the point in the local coordinate system.</
553 documentation>
554 </annotation>
555 </element>
556 <element name="globalPoint" type="gml:PointPropertyType">
557 <annotation>
558 <documentation>Coordinates of the point in the global coordinate system stated in
559 srsName.</documentation>
560 </annotation>
561 </element>
562 </sequence>
563 </extension>
564 </complexContent>
565 </complexType>
566 <complexType name="GlobalCRSReferencePointPropertyType">
567 <sequence minOccurs="0">
568 <element ref="bim:GlobalCRSReferencePoint"/>
569 </sequence>
570 <attributeGroup ref="gml:AssociationAttributeGroup"/>
571 </complexType>
572 <simpleType name="BuildingClassType">
573 <restriction base="string"/>
574 </simpleType>
575 <simpleType name="BuildingUsageType">
576 <restriction base="string"/>
577 </simpleType>
578 <simpleType name="SiteClassType">
579 <restriction base="string"/>
580 </simpleType>
581 <simpleType name="SiteUsageType">
582 <restriction base="string"/>
583 </simpleType>
584 <simpleType name="RoomClassType">
585 <restriction base="string"/>
586 </simpleType>
587 <simpleType name="RoomUsageType">
588 <restriction base="string"/>
589 </simpleType>
590 <simpleType name="POIClassType">
591 <restriction base="string"/>
592 </simpleType>
593 <simpleType name="AOIClassType">
```

A. Building Information Geography Markup Language

```
    <restriction base="string"/>
591 </simpleType>
    <simpleType name="AOIUsageType">
593   <restriction base="string"/>
    </simpleType>
595 </schema>
```