

TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Computergestützte Modellierung und Simulation

**Methoden zur Abbildung menschlichen Navigationsverhaltens  
bei der Modellierung von Fußgängerströmen**

Angelika Kneidl

Vollständiger Abdruck der von der Ingenieur fakultät Bau Geo Umwelt der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. rer. nat. E. Rank

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. A. Borrmann
2. Univ.-Prof. Dr. rer. nat. M. Bader
3. Prof. Dr. rer. nat. G. Köster,  
Hochschule München

Die Dissertation wurde am 06.03.2013 bei der Technischen Universität München eingereicht und durch die Ingenieur fakultät Bau Geo Umwelt am 28.05.2013 angenommen.



## Zusammenfassung

Die vorliegende Arbeit befasst sich mit Methoden zur Abbildung menschlichen Navigationsverhaltens im Rahmen von mikroskopischen Fußgängersimulationsmodellen. Als Grundlage wurden Methoden zur Erzeugung von Navigationsgraphen entwickelt. Diese Graphen sollen auf der einen Seite so ausgedünnt wie möglich sein, um die Performanz der auf ihnen implementierten Algorithmen zu erhöhen. Gleichzeitig sollen sie automatisiert aus einer Szenariogeometrie zu erstellen sein, um auch große komplexe Szenarien modellieren zu können. Daher wird ein kegelbasierter Algorithmus eingeführt, der einen ausgedünnten Sichtbarkeitsgraphen, der bei minimaler Kantenanzahl gleichzeitig die räumliche Gebietsabdeckung erhält, automatisiert aus einer gegebenen Szenariogeometrie erstellt. Der so generierte Navigationsgraph dient in einem neuen Verfahren als Schnittstelle zu makroskopischen Netzwerkflussmodellen. Durch Kopplung des mikroskopischen Modells mit einem makroskopischen Modell können Evakuierungszeiten präziser berechnet werden.

Um menschliches Wegefindungsverhalten in das mikroskopische Simulationsmodell zu integrieren, werden kognitive Abläufe während der Wegefindung in Algorithmen überführt. Es wird zwischen ortskundigen und ortsfremden Personen unterschieden, deren Verhalten jeweils in Normalsituationen und im Evakuierungsfall modelliert wird. Dazu werden unterschiedliche Metriken auf dem Graphen definiert.

Die Validierung der entwickelten Algorithmen erfolgt durch Daten aus einem an der Technischen Universität München durchgeführten Experiment. Dazu wird eine neue Metrik „similarity“ eingeführt, die es erlaubt, aussagekräftige Vergleiche zwischen Simulationsergebnissen und Daten des Experiments durchzuführen. Die Vergleichsergebnisse stützen die These, dass sich Fußgänger nicht ausschließlich nach dem schnellsten Weg orientieren, sondern eine Vielzahl von Faktoren die Wegewahl beeinflussen. Diese Faktoren werden von den vorgestellten Algorithmen realistisch abgebildet.

Der eingeführte ausgedünnte Navigationsgraph ermöglicht eine Kombination von mikroskopischen Modellen mit makroskopischen Netzwerkflussmodellen, um realistischere untere Schranken für Evakuierungszeiten zu ermitteln. Die Abbildung von kognitiven Prozessen in Algorithmen führt zu einer realistischen Modellierung menschlichen Navigationsverhaltens.

## Abstract

This thesis introduces new methods for modelling human wayfinding behavior in the context of microscopic pedestrian simulation models. The basis forms a navigation graph, which is automatically derived from the underlying geometry of a scenario. Applying a cone-based search method, the resulting graph is on the one hand as sparse as possible and on the other hand detailed enough to be able to serve as a wayfinding basis.

It is used to combine the microscopic simulation model with a macroscopic dy-

dynamic network model in terms of a control cycle. As a result, the gap between the evacuation times computed by both models is reduced: The coupled model considers both optimized routing strategies and microscopic effects.

To model human wayfinding behavior, cognitive processes are considered. These processes are translated into wayfinding algorithms, which are based on the derived navigation graph. Different levels of familiarity of a place as well as the distinction between everyday life and evacuation situations are considered by defining different metrics on the graph.

These algorithms are validated by an experiment performed at Technische Universität München. To compare real-world data to the results of the algorithms, a new metric „similarity“ is introduced. This metric allows meaningful comparisons between simulation data and experimental data. The results showed a good match between the real world data and the algorithms. This leads to the conclusion that human wayfinding does not only rely on one specific criterion, but is influenced by different factors.

The usage of sparse navigation graphs enables a combination between macroscopic network flow models and microscopic models to derive more realistic evacuation times. The incorporation of cognitive processes led to the realistic modelling of human wayfinding behavior.

## Vorwort

Diese Arbeit entstand von April 2009 bis Dezember 2012 im Rahmen eines von der Siemens AG gewährten Promotionsstipendiums während meiner Tätigkeit am Lehrstuhl für Computation in Engineering, und ab Februar 2011 am Lehrstuhl für Computergestützte Modellierung und Simulation der Technischen Universität München.

Ein ganz besondere Dank geht hierbei an Prof. Dr. rer.-nat. Ernst Rank und an Prof. Dr.-Ing. André Borrmann. Professor Rank danke ich für sein Vertrauen in meine Fähigkeiten und vor allem zu Beginn meiner Promotion an die äußerst produktiven und sehr konstruktiven Gespräche. Ich habe seine Fähigkeit, sich in eine Thematik innerhalb kürzester Zeit hineinzudenken und wertvolle Anregungen zu geben, immer sehr geschätzt. Nach seinem Ruf zum Ordinarius hat André Borrmann die Betreuung komplett übernommen. Bereits als Betreuer stand Prof. Borrmann mir jeder Zeit mit Rat und Tat zur Seite. In vielen intensiven und konstruktiven Diskussionen entstand letztendlich diese Arbeit, die in dieser Art ohne Professor Borrmann nicht möglich gewesen wäre. Zudem möchte ich mich noch einmal ausdrücklich dafür bedanken, dass er immer Zeit und ein offenes Ohr hatte, wenn ich zu ihm kam.

Ich möchte außerdem Herrn Prof. Dr. rer.-nat. Michael Bader für die Übernahme des Zweitprüfers sehr danken. Für sein Interesse an diesem Thema und die intensive Auseinandersetzung und wertvollen Tipps bin ich sehr dankbar.

Prof. Dr. rer.-nat. Gerta Köster gilt ebenfalls ein großer Dank. Bereits in ihrer Rolle als Projektleiterin bei der Siemens AG und später als Professorein an der Hochschule München hat sie mich sowohl fachlich als auch menschlich immer bestens unterstützt. Ich freue mich sehr, dass sie meiner Anfrage, sie als Drittprüferin zu gewinnen, nachgekommen ist und mir zu dieser Arbeit viel wertvolles Feedback gegeben hat.

Ohne die Siemens Crew, aus der ich ganz besonders Herrn Dr. Dirk Hartmann und Dr. Wolfram Klein herausheben möchte, wäre diese Arbeit nicht zustande gekommen. Mit der zur Verfügungstellung des von Siemens entwickelten Codes konnten die Ansätze in dieser Arbeit umgesetzt werden und sowohl Herr Hartmann als auch Herr Klein haben viel kreatives Feedback und Input geliefert.

Natürlich wäre das ganze Promotionsvorhaben ohne die Unterstützung von netten Kollegen nicht möglich gewesen. Für das tolle kollegiale Klima am Lehrstuhl möchte ich mich bei allen ganz herzlich bedanken. Ganz besonderer Dank gilt Vasco Varduhn, der mich von der ersten Minute an begleitet hat und immer ein offenes Ohr für meine Fragen hatte und mir stets geholfen hat.

Tanja Scherzl, Jessica Tilly, Christina Kreis und vor allem Monika Kneidl danke ich noch einmal ausdrücklich dafür, dass sie sich die Zeit und die Mühe gemacht haben, diese Arbeit Korrektur zu lesen.

Ein ganz besonderer Dank gilt Christian Hohenegger, der mit viel Geduld, vor allem in den letzten Wochen, immer hinter mir stand und mich unterstützt hat. Last but not least möchte ich meiner Familie für alles danken, die in all meinen Entscheidungen immer hinter mir steht und immer für mich da ist.



# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Ausgangspunkt und Zielsetzung . . . . .	2
1.3	Aufbau der Arbeit . . . . .	4
<b>2</b>	<b>Modelle zur Simulation von Fußgängerströmen</b>	<b>7</b>
2.1	Modellierungsebenen zur Abbildung von Fußgängerverhalten . . . . .	7
2.2	Kategorisierung von Simulationsmodellen . . . . .	8
2.2.1	Makroskopische Modelle . . . . .	9
2.2.2	Mikroskopische Modelle . . . . .	10
2.2.3	Kontinuierliche vs. diskrete Modelle . . . . .	13
2.2.4	Einordnung des Modells bezüglich weiterer Kategorien . . . . .	14
2.3	Bewertung der Simulationsmodelle bezüglich der Modellierungsebenen . . . . .	14
<b>3</b>	<b>Lokomotionsebene des Simulationsmodells</b>	<b>17</b>
3.1	Elemente eines Szenarios . . . . .	18
3.2	Modellierung der Lokomotionsebene . . . . .	19
3.2.1	Diskretisierung des Raums und der Zeit: Zellulärer Automat . . . . .	19
3.2.2	Modellierung des Bewegungsverhaltens: Navigationsfelder . . . . .	21
3.3	Zusammenfassung . . . . .	29
<b>4</b>	<b>Taktische Ebene des Simulationsmodells</b>	<b>31</b>
4.1	Bekannte Ansätze zur Erzeugung von Navigationsgraphen . . . . .	31
4.2	Kegelbasierter Ansatz zur Generierung von Navigationsgraphen . . . . .	34
4.2.1	Erzeugung der Knotenmenge $V$ . . . . .	34
4.2.2	Erzeugung der Kantenmenge $E$ . . . . .	37
4.3	Qualitativer Vergleich des ausgedünnten Sichtbarkeitsgraphen mit einem vollständigen Sichtbarkeitsgraphen . . . . .	49
4.4	Zusammenfassung . . . . .	60
<b>5</b>	<b>Der Navigationsgraph als Schnittstelle zwischen mikroskopischen und makroskopischen Simulationsmodellen</b>	<b>61</b>
5.1	Motivation für das Kopplungsverfahren . . . . .	61
5.2	Makroskopische Komponente: Dynamisches Netzwerkflussmodell . . . . .	63
5.2.1	Quickest Flow Problem . . . . .	63
5.2.2	Aufbau des Netzwerks für die Kopplung . . . . .	64

5.3	Mikroskopische Komponente: Erweiterung des mikroskopischen Modells für die Kopplung . . . . .	65
5.4	Aufbau und Parameter des Kopplungsverfahrens . . . . .	65
5.4.1	Parameter für das Kopplungsverfahren . . . . .	66
5.4.2	Ablauf des Kontrollzyklus . . . . .	67
5.5	Tests . . . . .	70
5.5.1	Schematische Testfälle . . . . .	70
5.5.2	Anwendungsbeispiel: Minimale Kopplung für komplexe Szenarien . . . . .	82
5.6	Zusammenfassung . . . . .	85
<b>6</b>	<b>Wegefindung:</b>	
	<b>Routingalgorithmen zur Abbildung menschlichen Navigationsverhaltens</b>	<b>87</b>
6.1	Kognitive Aspekte bei der Wegefindung von Menschen . . . . .	87
6.2	Algorithmen zur Abbildung unterschiedlicher Strategien bei der Wegefindung . . . . .	91
6.2.1	Strategie „Schnellster Weg“: Fastest Path Algorithmus . . . . .	92
6.2.2	Strategie „Möglichst nahe entlang der Luftlinie zum Ziel“: Heuristic Path Algorithmus . . . . .	101
6.2.3	Strategie „Orientierung anhand der Luftlinie, aber ortsfremd“: Greedy Heuristic Path Algorithmus . . . . .	107
6.2.4	Strategie „Möglichst wenige Richtungswechsel“: Straight and Long Legs (SALL) Algorithmus . . . . .	110
6.2.5	Strategie „Anderen Personen folgen“: Probabilistic Choice Algorithmus . . . . .	113
6.2.6	Evakuierung – zum nächstgelegenen Ziel laufen . . . . .	120
6.3	Einbinden von Landmarks . . . . .	121
6.4	Zusammenfassung . . . . .	124
<b>7</b>	<b>Validierung</b>	<b>125</b>
7.1	Einführung . . . . .	125
7.2	State of the Art: Validierung . . . . .	127
7.3	Validierung der taktischen Ebene . . . . .	128
7.3.1	Aufbau des Experiments . . . . .	128
7.3.2	Ergebnisse des Experiments . . . . .	129
7.3.3	Simulationsaufbau . . . . .	133
7.3.4	Definition der Vergleichsmetriken . . . . .	136
7.3.5	Ergebnisse des Vergleichs . . . . .	140
7.4	Zusammenfassung . . . . .	169
<b>8</b>	<b>Zusammenfassung und Ausblick</b>	<b>171</b>
8.1	Zusammenfassung der Arbeit . . . . .	171
8.2	Ausblick . . . . .	174
	<b>Literaturverzeichnis</b>	<b>175</b>



# Kapitel 1

## Einführung

### 1.1 Motivation

Großveranstaltungen wie Volksfeste, Sportveranstaltungen, Demonstrationen, Konzerte oder Messen spielen im gesellschaftlichen Leben eine wichtige Rolle. Die Anzahl der Besucher dieser Veranstaltungen nimmt stetig zu, zum Einen durch die erhöhten Kapazitäten der Austragungsstätten (größere Stadien, größere Konzerthallen), zum Anderen durch die größere Mobilität der Menschen. Eine übergeordnete Rolle bei der Planung und Durchführung solcher Veranstaltung spielt die Sicherheit. Durch die große Anzahl von Besuchern stellt der reibungslose Ablauf bei der Durchführung dieser Veranstaltungen und die Aufrechterhaltung der Sicherheit bei unvorhergesehenen Ereignissen eine sehr große Herausforderung für die Planer und Sicherheitskräfte dar.

In Deutschland gibt es unterschiedliche Verordnungen, wie die Versammlungsstättenverordnung (VStättV) sowie bauliche Verordnungen oder Brandschutzverordnungen, um den Planern Richtwerte zur Dimensionierung solcher Veranstaltungen an die Hand zu geben.

Diese Verordnungen stützen sich auf Erkenntnisse aus der Wissenschaft, die sich seit Anfang der 1970er Jahre mit der Dimensionierung von Fußgängeranlagen beschäftigt. Die Dimensionierung fordert gewisse Mindestmaße, um die Sicherheit von Fußgängern zu gewährleisten. Weidmann (1993) erhob Anfang der 1990er Jahre empirische Daten über die Eigenschaften von Fußgängern wie Größe, Geschlecht, Alter, Platzbedarf, Geschwindigkeit, Dichten usw. und leitete daraus Parameter für die Dimensionierung von Fußgängeranlagen ab.

Allein eine auf diese statischen Größen gestützte Dimensionierung kann jedoch nicht die Dynamik von Menschenmassen erfassen.

Daher spielen zusätzlich zu Verordnungen Erfahrungswerte der Einsatzkräfte eine wichtige Rolle.

Dennoch kann bei unvorhergesehenen Vorfällen eine Dynamik innerhalb von Menschenmassen auftreten, die weder durch Verordnungen noch durch langjährige Erfahrung vorhergesehen werden kann.

Aufgrund der höheren Rechenleistung und der daraus resultierenden Möglichkeit der computergestützten Simulation wurden in den letzten Jahrzehnten Modelle entwickelt, die die Dynamik von Fußgängerströmen abbilden können. Sie können dazu verwendet werden, in der Planungsphase „Was-Wäre-Wenn“ Szenarien zu simulieren und so kritische Situationen und mögliche Konfliktpunkte zu identifizieren und den Planern vorab die Möglichkeit zu geben, durch entsprechende Maßnahmen diese Situationen zu entschärfen.

Diese Simulationsmodelle helfen nicht nur bei der Planung und Durchführung von Großveranstaltungen, sondern finden ihren Einsatz in vielen weiteren Anwendungsgebieten.

Im Bauwesen werden Fußgängersimulationen verwendet, um mögliches Verkehrsaufkommen und dessen Auswirkung innerhalb und außerhalb von Gebäuden zu simulieren. Bei der Planung von Gebäuden kann eine Simulation dazu eingesetzt werden, Engstellen und mögliche Konfliktpunkte vorab zu erkennen und das Gebäude so zu dimensionieren, dass solche Engstellen nicht auftreten. Auch beim Umbau oder Modernisierung von Gebäuden kann eine Simulation von unterschiedlichen Szenarien a priori Alternativen von Umbaumaßnahmen bewerten.

Desweiteren werden Simulationen zur Betrachtung von Evakuierungsszenarien verwendet. Mit ihrer Hilfe können minimale Evakuierungszeiten für Gebäude, Schiffe, Flugzeuge oder auch ganzen Regionen ermittelt und daraus optimale Evakuierungswege abgeleitet werden.

In einem ganz anderen Einsatzgebiet unterstützen die Ergebnisse von Fußgängersimulationen den Planungsprozess von Freizeitparks. Durch die Anwendung solcher Simulationen können Wartezeiten an Attraktionen sowie Besucherströme berechnet werden und damit den Parkdesignern wichtige Größen zum Design und zur Platzierung von Attraktionen liefern.

Auch in Kombination mit Videotechnik steht die Fußgängersimulationen seit kurzem als wichtiges Instrument zur Verfügung. Mithilfe solcher Simulationen können ausgehend von aktuellen Videodaten Kurzzeitprognosen berechnet werden und als Unterstützung für das Personal von Leitzentralen in Bahnhöfen, Stadien oder öffentlichen Veranstaltungen Verwendung finden.

## 1.2 Ausgangspunkt und Zielsetzung

In den letzten Jahrzehnten wurden eine Reihe verschiedener Ansätze entwickelt, diese breitgefächerten Einsatzgebiete zu erfassen und Simulationsmodelle zu erstellen. Aufbau und Ziele der Modelle unterscheiden sich dabei erheblich.

Makroskopische Modelle betrachten ein Szenario gesamtheitlich und verwenden als Berechnungsgrundlage aggregierte Werte wie beispielsweise Mittelwerte der Eingangsgrößen wie Geschwindigkeit oder Dichten. Sie werden beispielsweise zur Abschätzung unterer Schranken von Evakuierungszeiten eingesetzt.

Mikroskopische Modelle dagegen modellieren Fußgänger als Individuen mit dem Ziel, Interaktionen zwischen Fußgängern und das Verhalten in Menschenmengen abzubilden. Dies wird durch sehr unterschiedliche Ansätze umgesetzt.

Als Beispiel seien kräftebasierte Modelle genannt, die sich gut für die Modellierung von hohen Dichten und deren Auswirkungen in großen Menschenmengen eignen.

Agentenbasierte Modelle auf der anderen Seite eignen sich gut, um Aktivitäten abzubilden, die in einer bestimmten Reihenfolge von Fußgängern ausgeführt werden. Solche Abläufe existieren beispielsweise an Bahnhöfen oder Flughäfen. Sie stützen sich dabei auf den Ansatz, dass Personen unterschiedliche Verhaltensweisen an den Tag legen und bilden verschiedene Charaktere ab. Die Ergebnisse der agentenbasierten Simulationsmodelle liefern Aussagen über mögliche Kapazitätsauslastungen und helfen beim Betrieb von Infrastrukturen; beispielsweise das Öffnen von Schaltern, das Platzieren von Kiosken, das Einrichten von Wartezonen etc..

Auch wenn die Ansätze sehr unterschiedliche Ausprägungen annehmen, müssen dieselben Aspekte zur Modellierung von Fußgängerverhalten betrachtet werden. Dazu zählt die lokale Kollisionsvermeidung, die Modellierung von Lokomotionsverhalten (z.B. Laufen, Gehen, Schlendern, Stoppen), das Abbilden typischer Bewegungsmuster, die vor allem in dichten Menschenmengen zu beobachten sind, sowie globale Wegefindungsstrategien.

Diese Aspekte können in lokales Verhalten auf der Lokomotionsebene und globale Wegefindungsstrategien auf der taktischen Ebene eingeordnet werden. Lokales Verhalten ist vorwiegend durch Ereignisse innerhalb des Sichtfelds geprägt, wird aber von globalen Wegefindungsstrategien beeinflusst, die auch nicht einsehbare Bereiche mit einbezieht.

Bei den meisten mikroskopischen Modellen liegt der Fokus auf einer möglichst realistischen Abbildung des lokalen Verhaltens. Für die globalen Wegefindungsstrategien werden einfache Modelle verwendet, die in der Regel nur unzureichend menschliches Verhalten abbilden.

Bei der Entwicklung makroskopischer Modelle hingegen liegt der Schwerpunkt eher auf der globalen Planung, lokale Interaktion zwischen den Personen wird dabei vernachlässigt. Die globalen Wegefindungsstrategien werden abgebildet, indem ein auf globaler Ebene optimales Ergebnis (wie beispielsweise die minimale Evakuierungszeit) berechnet wird. Diese so ermittelten Ergebnisse liegen in der Regel unterhalb der real benötigten Zeit, da zum Einen Fußgänger global nicht optimal planen können, denn es fehlt ihnen die Kenntnis über andere Fußgänger und eventuelle Gegebenheiten wie beispielsweise Staus. Zum Anderen wird lokales Verhalten und dessen Einfluss auf die globalen Strategien nicht betrachtet.

Bisher existieren keine Modelle, die bei der Modellierung sowohl mikroskopische als auch makroskopische Aspekte berücksichtigen. Während bei der Entwicklung mikroskopischer Modelle der Schwerpunkt auf der Abbildung von lokalem Verhalten liegt und als globale Wegefindungsstrategien meist einfache Routingalgorithmen eingesetzt werden, konzentrieren sich makroskopische Netzwerkmodelle

auf die Ermittlung von unteren Schranken für Evakuierungszeiten ohne mikroskopische Aspekte zu betrachten.

In dieser Arbeit soll diese Lücke geschlossen werden. Dazu werden Methoden zur realitätsnahen Abbildung globaler Wegefindungsstrategien vorgestellt, die in ein mikroskopisches Modell integriert werden können. Diese Methoden basieren auf einem Navigationsgraph, um den das Modell erweitert wird. Er bildet die Grundlage für die Wegefindung. Zudem wird er als Schnittstelle zu makroskopischen Modellen eingesetzt. Damit werden diese Modelle um den Einfluss von lokalem Verhalten erweitert.

### 1.3 Aufbau der Arbeit

Zunächst wird in **Kapitel 2** ein Überblick über den aktuellen Stand der Wissenschaft im Bereich der Simulation von Fußgängerströmen gegeben.

In **Kapitel 3** wird das mikroskopische Modell beschrieben, das als Grundlage zur Untersuchung in dieser Arbeit verwendet wird.

In den Kapiteln 4 bis 7 werden die miteinander vernetzten Schwerpunkte dieser Arbeit beschrieben, die die Lücke zwischen mikroskopischen und makroskopischen Aspekten schließen.

Zunächst wird in **Kapitel 4** das mikroskopische Modell um einen Navigationsgraphen erweitert, der automatisiert aus einer Szenariogeometrie erstellt wird. Dazu wird ein neues Verfahren vorgestellt und bewertet, das einerseits einen ausgedünnten Graphen erstellt und gleichzeitig eine größtmögliche Gebietsabdeckung gewährleistet. Damit können globale Wegefindungsstrategien der taktischen Ebene effizient abgebildet werden. Gleichzeitig kann der vorliegende Graph als Schnittstelle zu makroskopischen Simulationsmodellen eingesetzt werden.

In **Kapitel 5** wird ein Verfahren vorgestellt, das diesen Navigationsgraphen als Schnittstelle zu einem makroskopischen Modell einsetzt. Das mikroskopische Modell wird mit einem makroskopischen Modell in Form eines Regelkreises gekoppelt, um zum Einen in das makroskopische Modell lokales Verhalten mit einfließen zu lassen und damit die Genauigkeit der Vorhersagen zu erhöhen. Zum Anderen können in das mikroskopische Modell globale Strategien integriert werden. Diese globale Strategien können den positiven Einfluss einer ordnenden Instanz (Polizei, Sicherheitskräfte) abbilden, die in der Regel einen Gesamtüberblick haben und Personen global optimal leiten. Das Verfahren wird schematisch an kleinen realistischen Testfällen demonstriert. Aus den Ergebnissen der Tests lässt sich ein leicht abgeändertes Verfahren ableiten; durch die bereits große Annäherung der Evakuierungszeiten beider Verfahren in der ersten Runde der Kopplung wird von dieser Kenntnis Gebrauch gemacht, eine „minimale Kopplung“ durchzuführen, die nur aus einer Iteration besteht. Das spart zum Einen Rechenzeit, garantiert aber gleichzeitig eine erhebliche Annäherung der Ergebnisse beider Verfahren, ohne dass sich Oszillationen potenzieren und damit das Erreichen eines stationären

Zustands verhindern. Dieses abgeänderte Verfahren wird an einem komplexen Realbeispiel demonstriert.

**Kapitel 6** widmet sich der Abbildung von globalen Wegefindungsstrategien auf Basis des in Kapitel 3 eingeführten Navigationsgraphen. Zur Modellierung unterschiedlicher Wegefindungsstrategien werden kognitive Aspekte mit einbezogen. Die Übertragung dieser Aspekte in Algorithmik, mit dem Ziel, diese mit Computersimulationen abbilden zu können, wird detailliert beschrieben. Es werden unterschiedliche Typen von Fußgängern modelliert, die in der Kombination abhängig des zu simulierenden Grundsachverhalts zu einer realistischeren Abbildung der Wegefindungsstrategien führen.

Zur Validierung der vorgestellten Algorithmen werden in **Kapitel 7** Ergebnisse eines Experiments mit den Ergebnissen des erweiterten Modells verglichen und bewertet. Dazu wird eine neue Vergleichsmetrik „similarity“ eingeführt, die zusätzlich zum visuellem Vergleich aussagekräftige Aussagen über die Qualität der vom Simulator berechneten Routen ermöglicht.

**Kapitel 8** fasst die beschriebene Ansätze zusammen und gibt einen Ausblick auf Weiterentwicklungsmöglichkeiten von Fußgängersimulationsmodellen.



## Kapitel 2

# Modelle zur Simulation von Fußgängerströmen

In den letzten Jahrzehnten wurden weltweit viele unterschiedliche Modelle und Methodiken zur Simulation von Fußgängerströmen entwickelt. Dieses Kapitel gibt einen kurzen Überblick über die unterschiedlichen Aspekte zur Modellierung von Fußgängerströmen und die gängigsten Verfahren, um das hier benutzte Verfahren zu kategorisieren und zu motivieren. Anschließend werden die Modelle in Bezug auf die Abbildung dieser Ebenen bewertet.

### 2.1 Modellierungsebenen zur Abbildung von Fußgängerverhalten

Hoogendoorn & Bovy (2004) definieren drei unterschiedliche Ebenen zur Modellierung von Fußgängerverhalten: die strategische Ebene, die taktische Ebene und die operationelle Ebene. Eine ähnliche Klassifizierung verwendet auch Reynolds (1999), der bei seinem Modell zwischen strategischer, taktischer und Lokomotionsebene unterscheidet. Diese Ebenen dienen dazu, unterschiedliche Skalen der Wahrnehmung (Perzeption) und daraus resultierende (Re-)aktionen zu modellieren. Es wird zwischen drei Skalen unterschieden, der unmittelbaren Umgebung (Mikroskala), den Sichtbereich (Mesoskala) sowie Bereiche, die nicht einsehbar sind (Makroskala).

Auf der strategischen Ebene wird die Absicht der Bewegung beschrieben (Reynolds, 1999). Hier wird unterschieden, ob es sich bei dem zu simulierenden Grundsachverhalt beispielsweise um ein alltägliches Szenario (z.B. der tägliche Arbeitsweg an Bahnhöfen, Flughäfen usw.), eine Freizeitveranstaltung wie ein Konzert, Fußballspiel oder Demonstration oder ein Evakuierungsszenario handelt. Diese Ebene kann im Allgemeinen nicht modelliert werden, sondern dient als Grundlage für die Wahl des Simulationsmodells (vgl. Abschnitt 2.3) und der Parameter. Mit unterschiedlichen Parametern wie Geschwindigkeit, Zielstrebigkeit oder Gruppengröße kann hier auf verschiedene Grundsachverhalte eingegangen werden.

Die taktische Ebene modelliert den Wegefindungsprozess der Fußgänger. Von seinem aktuellen Ort möchte der Fußgänger in Richtung eines Ziels gehen, das im Allgemeinen nicht einsehbar ist. Diese Ebene formt die Makroskala der Wahrnehmung und bildet die unterschiedlichen Strategien der Fußgänger bei der Wegefindung ab.

Die Lokomotionsebene beschreibt die Art der Fortbewegung. Unterschiedliche Arten wie Laufen, Gehen, Anhalten, Schlendern, das Schieben eines Kinderwagens, Rollstuhl fahren etc. werden hier abgebildet. Zudem spielt die Kollisionsvermeidung eine Rolle. Kollisionsvermeidung beschreibt die Art, wie Menschen einander ausweichen. Dabei wird die menschliche Wahrnehmung auf Mikro- und Mesoskala mit einbezogen.

Zusätzlich zu den in der Literatur beschriebenen Ebenen muss die Verhaltensebene über alle drei Schichten hinweg betrachtet werden. Die Verhaltensebene beschreibt die Formationen sowie Zusammensetzung von Gruppen, das Verhalten beim Anstehen (engl. *queueing*), Bewegungsmuster innerhalb von dichten Menschenmassen, das Warten auf Gruppenmitglieder oder andere vom Verhalten der Menschen ausgelöste Ereignisse.

Eine Übersicht über die unterschiedlichen Ebenen und deren Zusammenspiel ist in Abbildung 2.1 gegeben. In der Literatur finden sich mehrere Synonyme für die

Strategische Ebene: Grund der Bewegung	Verhaltensebene: Gruppenverhalten, Anstehverhalten etc.
Taktische Ebene: Navigation zum Ziel	
Lokomotionsebene: Bewegungsart	

Abbildung 2.1: Modellierungsebenen

hier eingeführten Ebenen. So wird die Lokomotionsebene auch als mikroskopische Schicht bezeichnet, die taktische Ebene als Navigationsebene oder mesoskopische Ebene. Die strategische Ebene wird entsprechend als makroskopische Ebene vorgestellt. In dieser Arbeit kommen die Begriffe wie in Abbildung 2.1 dargestellt zur Anwendung.

## 2.2 Kategorisierung von Simulationsmodellen

In Schadschneider *et al.* (2009) ist eine ausführliche Übersicht über die unterschiedlichen Simulationsmodelle zu finden. Hierin unterscheiden die Autoren zwischen

- mikroskopischen und makroskopischen



- diskreten und kontinuierlichen
- deterministischen und stochastischen
- regelbasierten und kraftbasierten

Ansätzen, sowie Ansätze, die einen hohen bzw. geringen Genauigkeitsgrad in der Modellierung aufweisen.

Im Folgenden wird auf die ersten beiden Arten der Kategorisierung genauer eingegangen, da das hier zugrundeliegende Modell anhand der genannten Attribute am Besten einzuordnen ist.

### 2.2.1 Makroskopische Modelle

Makroskopische Modelle umfassen Modelle, die eine gesamtheitliche Sicht auf das zu modellierende Szenario haben und somit „allwissend“ sind. Allwissend bezogen auf die Anzahl, die Position und die Bewegung der Personen, die innerhalb eines Szenarios betrachtet werden. Der Fokus solcher Modelle liegt darin, Kenngrößen eines Szenarios im Gegensatz zu Interaktionen einzelner Personen zu bestimmen. Daher dienen aggregierte Werte als Parameter, wie zum Beispiel Durchschnittsdichten oder -geschwindigkeiten. Bekannte Vertreter solcher makroskopischer Simulationsansätze sind beispielsweise netzwerkbasierte Modelle, strömungsbasierte Modelle oder gaskinetische Modelle.

**Netzwerkflussmodelle** werden dazu verwendet, Optimierungsprobleme wie das *Quickest Flow Problem* (z.B. Burkard *et al.*, 1993, Tjandra, 2003) an Fußgängerströme zu adaptieren und zu lösen. Sie werden dazu verwendet, Evakuierungsszenarien zu modellieren. Die Ergebnisse entsprechen minimalen Evakuierungszeiten unter Annahme von bestimmten Eingangsparametern. Eine genauere Beschreibung dieser Art von Modellen erfolgt in Kapitel 5.

Treuille *et al.* (2006) und Hughes (2000, 2003) befassen sich mit **Kontinuumsmodellen**, die ein globales Potentialfeld generieren, anhand dessen die Bewegungen der Fußgänger definiert sind. Die Potentialfelder werden entweder auf Basis der Minimierung von Reisezeiten (Hughes, 2000, 2003) oder auf Basis der Minimierung von Unbehagen (engl. *discomfort*) gebildet (Treuille *et al.*, 2006).

**Strömungsbasierte Modelle sowie gaskinetische Modelle** basieren auf der Analogie zwischen Strömungsphänomenen bei Fluiden bzw. Ausbreitung von Gasen im Raum, und Menschen, die sich in Massen bewegen oder sich in einem Raum befinden. Eingeführt von Henderson (1971, 1974), modellieren sie die Interaktionen zwischen Personen mit der Boltzmann Transport Gleichung (Henderson, 1974). Henderson geht von den konventionellen Gleichungen aus, in denen der Erhalt von Energie und Impuls angenommen werden. Interaktionen zwischen Fußgängern werden als Kollisionsprozess beschrieben, in dem Energie und Impuls ausgetauscht werden.

Menschliches Verhalten entspricht nicht ausschließlich physikalischen Gesetzen, vielmehr spielen auch kognitive, psychologische und soziologische Aspekte eine Rolle. Der Erhalt von Energie und Impuls ist bei Fußgängerbewegungen nicht ausschlaggebend. Dadurch entstehen Probleme bei der Abbildung von menschlichem Verhalten: Beispielsweise werden die Anisotropie bei Kollisionen von Men-

schen oder die Vorzugsrichtung von Fußgängern in Richtung Ziel nicht präzise genug abgebildet. Eine genauere Analyse der Schwachstellen sowie ein Vorschlag für ein verbessertes Modell ist in Helbing (1992) gegeben. Aber auch bei diesem verbesserten Modell treten Schwachstellen bei geringen Dichten auf. Es entstehen Oszillationen, die dem menschlichen Verhalten nicht entsprechen. Aufgrund dieser Erkenntnisse haben Helbing & Molnár (1995) ein mikroskopisches Modell entwickelt, das die einzelnen Partikel (Fußgänger) individuell modelliert.

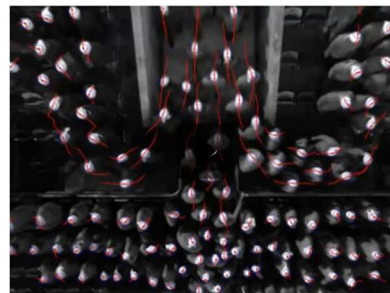
### 2.2.2 Mikroskopische Modelle

Mikroskopische Simulationsmodelle betrachten die zu simulierenden Personen als Individuen und haben zum Ziel, Interaktionen zwischen den Fußgängern möglichst realistisch zu beschreiben. Dabei sollen die Modelle lokale Phänomene wie Bahnenbildung (siehe Abbildung 2.2a), Stau vor Engstellen (siehe Abbildung 2.2b) oder Stoßwellen (engl. *stop-and-go waves*) in Menschenmengen abbilden (Schadschneider *et al.*, 2009).

Bahnen innerhalb von Menschenmengen entstehen dadurch, dass Menschen dazu tendieren, anderen Menschen zu folgen, die in dieselbe Richtung laufen. Dies vermeidet Kollisionen mit anderen Personen und führt so zu Bahnenbildung (Schadschneider *et al.*, 2009).

Staus vor Engstellen können je nach Geometrie unterschiedliche Ausprägungen annehmen. Formieren sich Staus vor Engstellen, die aus einem breiten Korridor oder Raum herausführen, eher in Form eines Halbkreises, sind Staus vor Verengungen, die in schmaleren Gängen auftreten, eher trichterförmig (Seyfried *et al.*, 2009).

Stoßwellen entstehen in unidirektionalen dichten Fußgängerströmen (z.B. an Bahnsteigen, wenn alle Personen diese in Richtung Ausgang verlassen möchten). Sie sind quasi-periodische Wellen, die durch unterschiedliche Dichten und Geschwindigkeiten entstehen (Portz & Seyfried, 2011).



- (a) Hier sieht man eine typische Bahnenbildung, in dem Menschen versuchen, möglichst hinter ihren Vorgängern zu bleiben, um schnell durch eine Menschenmasse zu kommen. (Urheber des Fotos: Simon Garnier, entnommen von <http://www.mehdimoussaid.com/?p=119>)
- (b) Dieses Bild wurde innerhalb eines Experiments im Projekt Hermes<sup>1</sup> aufgenommen.

Abbildung 2.2: Mikroskopische Phänomene bei Fußgängerströmen

<sup>1</sup><http://www2.fz-juelich.de/jsc/appliedmath/ped/projects/hermes>

Ein Vorreiter der mikroskopischen Modelle ist das **Social Force Model** von Helbing & Molnár (1995). Dieses Modell kommt ursprünglich aus der makroskopischen Modellierung und nimmt als Grundidee an, dass jeder Fußgänger von anziehenden und abstoßenden Kräften beeinflusst wird. Die daraus resultierenden Kraftfelder werden in einem gemeinsamen Kraftfeld überlagert, entlang dem die Fußgänger in Richtung Ziel laufen. Jeder Fußgänger wird somit einzeln modelliert, hat jedoch keinen eigenen Willen, sondern wird von den auf ihn wirkenden Kräften gesteuert. Daher ist es sehr schwierig, mit dieser Art von Modell kognitive Prozesse oder individuelles Verhalten zu modellieren. Eine Möglichkeit, Verhaltensmuster zu integrieren, ist die Erweiterung des Modells um Anziehungskräfte, die den Fußgänger zusätzlich beeinflussen (z.B. Anziehung zu Schaufenstern oder zu anderen Gruppenmitgliedern). Je mehr solcher Kräfte modelliert werden, desto schwieriger ist eine Kalibrierung der einzelnen Kräfte und damit die Anpassung des Modells auf beliebige Situationen. Bei falscher Kalibrierung können lokale Minima auftreten, aus denen Fußgänger nicht wieder herausfinden.

Aufgrund des kräftebasierten Ansatzes tendieren die Trajektorien der Fußgänger dazu, bei niedrigen Dichten zu oszillieren (Chraïbi *et al.*, 2011). Das passiert immer dann, wenn mehrere Kräfte auf die Fußgänger wirken und je nach Position des Fußgängers die unterschiedlichen Kräfte alternierenden Einfluss haben. Dies ist zum Beispiel an Engstellen der Fall. Durch die abstoßende Wirkung der Wände kann es passieren, dass ein Fußgänger von einer zur anderen Wand abgestoßen wird und daraus ein unnatürliches Laufverhalten resultiert. Wenn dagegen hohe Dichten auftreten, ist das Modell gut geeignet: In dichten Mengen kann der Fußgänger seinen eigenen Willen nicht mehr gegen die physikalischen Kräfte durchsetzen. Daher können Phänomene wie z.B. Stoßwellen oder Turbulenzen (Helbing *et al.*, 2007) gut abgebildet werden.

Chraïbi (2012) entwickelte ebenfalls ein kräftebasiertes Modell mit dem Fokus, die Schwachstellen des bisherigen Social Force Modells zu analysieren und daraus ein verbessertes Modell abzuleiten. Er konzentriert sich dabei unter anderem auf die zweidimensionale Projektion der geometrischen Form bei der Abbildung von Fußgängern (Ellipse vs. Kreis) und den Einfluss von Richtungssteuerung innerhalb des Systems.

In Moussaïd & Helbing (2011) wird ein kognitiver Ansatz zur Modellierung von menschlichem Verhalten in Kombination mit einem Kräftemodell vorgeschlagen: Der Hauptfokus liegt darin, in dem Modell abzubilden, was Fußgänger (visuell) wahrnehmen und wie sie daraufhin ihr Laufverhalten verändern. Dies soll die Nachteile des reinen Social Force Modells eliminieren und so ein realistischeres Laufverhalten abbilden.

Langston *et al.* (2006) stellen ebenfalls ein auf Differentialgleichungen basierendes Kräftemodell vor. Zur Modellierung der Fußgängerbewegungen wird eine *Discrete Element Method* angewendet. Neben Newtonscher Mechanik zur Abbildung der Bewegungen werden zusätzlich psychologische Kräfte sowie Antriebsimpuls und -kraft modelliert.

All diese hier beschriebenen Modelle haben gemein, dass sie auf der Lösung von Differentialgleichungen basieren und durch Einstellung unterschiedlicher Parameter wie beispielsweise die Form der Partikel oder unterschiedlich stark be-

einflussende Kräfte variieren. Im Folgenden werden diese Art von Modellen als *Differential Equations Force Models (DEFM)* bezeichnet.

Eine andere Art mikroskopischer Modelle basieren auf **zellulären Automaten** und werden in Unterabschnitt 3.2.1 ausführlich beschrieben. Diese Modelle definieren eine Zerlegung des Raums in Zellen, auf denen sich Fußgänger bewegen können. Die Bewegung der Fußgänger wird dabei durch Zellwerte und -zustände sowie ein Regelset bestimmt. Dieses Regelset liefert die nächste Zelle für jeden Fußgänger, der sich auf diesem Zellgitter bewegt.

Klassische Ansätze haben ihren Ursprung in der Verkehrssimulation und wurden an Fußgänger angepasst (z.B. Blue *et al.*, 1997, Burstedde *et al.*, 2001, Fukui & Ishibashi, 1999, Varas *et al.*, 2007, Yamamoto *et al.*, 2007). Hier gibt es zufallsgesteuerte Automaten (z.B. Burstedde *et al.*, 2001, Kirik *et al.*, 2009) sowie deterministische Automaten (z.B. Blue *et al.*, 1997, Fukui & Ishibashi, 1999). Eine ausführliche Beschreibung unterschiedlicher Automaten angewandt auf Fußgängersimulation befindet sich in der Studienarbeit von Kinkeldey (2003).

Der Nachteil solch simpler zellulärer Automaten ist, dass nur sehr lokale Einflüsse modelliert werden können und zudem alle Fußgänger gleichgestellt sind, also denselben Regeln unterliegen. Die räumliche Diskretisierung birgt den Nachteil, dass nur eine begrenzte Anzahl von Bewegungsrichtungen abgebildet werden kann und durch die Zellgröße eine maximale Dichte vorbestimmt ist.

Eine Kombination von kräftebasierten Modellen und zellulären Automaten hat sich in der Folge entwickelt (z.B. Huang & Guo, 2008, Kretz, 2009, Köster *et al.*, 2010). Hier werden die Zellen des Automaten mit Werten belegt, die aus der Berechnung unterschiedlicher Potenziale resultieren. Dabei werden die Kräfte als so genannte Navigationsfelder interpretiert, die durch Flutung des Raums, analog einer Wellenausbreitung, die Zellwerte bestimmen. Dieses Verfahren wird in Unterabschnitt 3.2.2 detailliert beschrieben. In dieser Art von Verfahren werden keine Differentialgleichungen gelöst wie bei den im vorherigen Absatz beschriebenen *DEFM* Ansätzen, sondern Potenziale an bestimmten Punkten im Raum ausgewertet.

Die Flutung kann statisch (also einmal) (Huang & Guo, 2008) oder dynamisch (mehrmals während der Simulation) (Kretz, 2009, Hartmann, 2010b) mit bestimmten Potentialwerten erfolgen. Ein Überblick über unterschiedliche Möglichkeiten, Potentialwerte zu definieren, ist in (Kretz *et al.*, 2008) gegeben.

Ein ganz neues Modell wird von Seitz & Köster (2012) vorgeschlagen: Hier sollen die Vorteile von zellulären Automaten mit denen von *DEFM* vereint werden. Statt der räumlichen Diskretisierung des Automaten werden die Kräfte, die aus Navigationsfeldern ermittelt werden, an beliebigen Stellen ausgewertet. Dies führt zu einem natürlicheren Bewegungsmuster im Vergleich zu zellulären Automaten, mit gleichzeitig verbesserter Rechenzeit gegenüber kontinuierlichen Modellen.

Ein dritter Typ mikroskopischer Modelle hat seinen Ursprung in der künstlichen Intelligenz und modelliert das Verhalten der Fußgänger mithilfe von **agentenbasierten Modellen bzw. Verhaltensmodellen** (engl. *behavioral models*). Die Idee hierbei ist, möglichst realistisch autonom agierende Charaktere zu modellieren. Damit kann individuelles Verhalten sehr gut modelliert werden. Dies ist ein

Vorteil gegenüber Kräftenmodellen, die in ihrer ursprünglichen Art alle Fußgänger gleichbehandeln.

Eine Reihe von Wissenschaftlern haben agentenbasierte Modelle zur Simulation von Fußgängerverhalten entwickelt (z.B. Dijkstra *et al.*, 2006, Durupinar *et al.*, 2008, Funge *et al.*, 1999, Teknomo & Millonig, 2007, Lerner *et al.*, 2007, Rindsfuser & Klügl, 2007, Rodriguez & Amato, 2010, Sud *et al.*, 2008, Teknomo, 2008). Die zugrundeliegende Idee ist es, individuelle Charaktere anhand von Verhaltenssätzen und -regeln zu generieren und jedem Charakter einen Aktionsplan mitzugeben.

Hoogendoorn & Bovy (2004) stellen ein Modell vor, dass ein normatives Verhaltensmodell beschreibt. Alle Aktionen, die ein Fußgänger ausführt, verursachen Kosten. Diese Kosten kennt der Fußgänger im Voraus und optimiert sie. Fußgänger werden so modelliert, dass sie ihre Aktivitäten in einer für sie optimalen Weise abarbeiten.

Agentenbasierte Modelle bzw. Verhaltensmodelle werden vor allem zur Simulation von Städteplanung oder Planung von öffentlichen Einrichtungen (z.B. Einkaufszentren, Bahnhöfen) verwendet, da sie die Möglichkeit bieten, Abläufe zu modellieren.

Der Vorteil solcher Modelle ist es, dass kognitive Prozesse von Fußgängern sowie deren Verhalten sehr gut abgebildet werden können. Der Nachteil ist die enorme Rechenzeit, die benötigt wird, um alle Agenten zu modellieren sowie die fehlenden empirischen Daten, um Regelsets präzise zu definieren und zu kalibrieren.

### 2.2.3 Kontinuierliche vs. diskrete Modelle

Eine weitere Möglichkeit zur Unterscheidung von Modellen ist, den Grad der räumlichen und zeitlichen Diskretisierung zu betrachten. Diese Unterscheidung bezieht sich ausschließlich auf mikroskopische Modelle. Da es sich bei allen Modellen um computergestützte Simulationsmodelle handelt, ist eine zeitliche Diskretisierung unumgänglich. Die Granularität dieser Diskretisierung ist jedoch unterschiedlich. Sie hängt eng mit der räumlichen Diskretisierung zusammen. Kontinuierliche Modelle (z.B. Chraibi *et al.*, 2011, Durupinar *et al.*, 2008, Funge *et al.*, 1999, Henderson, 1971) diskretisieren den Raum nicht, jeder Punkt im Raum ist grundsätzlich erreichbar. Der Vorteil solcher Modelle ist, dass eine sehr hohe Genauigkeit in der Modellierung erreicht werden kann. Bei einer Nicht-Diskretisierung des Raums muss in jedem Schritt eine Kollisionsvermeidung durchgeführt werden, um zu vermeiden, dass sich zwei Personen auf der selben Position im Raum befinden. Daher sind diese Modelle im Allgemeinen sehr rechenintensiv. Kontinuierliche Räume werden häufig bei agentenbasierten Modellen eingesetzt. Auch *DEFM* Modelle arbeiten mit einem kontinuierlichen Raum.

Dem gegenüber stehen räumlich diskretisierte Modelle. Eine räumliche Diskretisierung erfolgt meist durch eine gleichmäßige Unterteilung des Raums in Zellen. Dabei werden entweder quadratische Zellen (z.B. Blue & Adler, 1999, Burstedde *et al.*, 2001, Fukui & Ishibashi, 1999, Huang & Guo, 2008, Kinkeldey, 2003, Kirik *et al.*, 2009, Pelechano & Malkawi, 2008, Varas *et al.*, 2007, Yamamoto *et al.*, 2007) oder hexagonale Zellen (z.B. Kinkeldey, 2003, Klein *et al.*, 2010) verwendet.

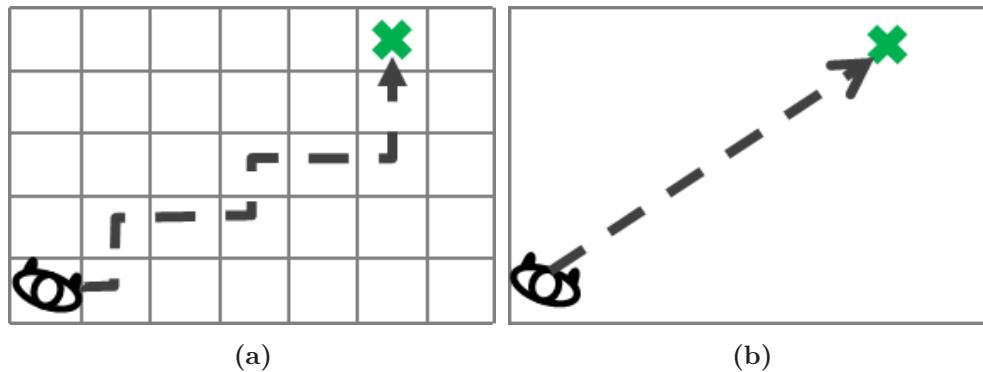


Abbildung 2.3: a) Räumliche Diskretisierung: Hier wurde der Raum in ein quadratisches Gitter unterteilt. Der Fußgänger in der süd-östlichen Ecke kann keinen direkten Weg zum Ziel auf diesem Gitter laufen. b) Im kontinuierlichen Raum ist jeder Punkt erreichbar und damit die direkte Strecke zwischen Fußgänger und Ziel modellierbar.

Ein Beispiel zur Unterscheidung von räumlich-diskretisierten sowie kontinuierlichen Modellen ist in Abbildung 2.3 dargestellt.

Weitere Ansätze diskretisieren den Raum dynamisch, je nach Geschwindigkeit des Fußgängers (Antonini *et al.*, 2006).

Definitionsgemäß verwenden Modelle, die auf zellulären Automaten basieren eine Diskretisierung des Raums. Eine neue, hybride Art wurde von Seitz & Köster (2012) vorgestellt. Hier wird der Raum nicht mehr explizit diskretisiert, vielmehr werden Bewegungen auf einem variablen Schrittkreis um die Fußgängerposition möglich.

#### 2.2.4 Einordnung des Modells bezüglich weiterer Kategorien

Die Unterscheidung zwischen stochastischen und deterministischen Modellen sowie zwischen regelbasierten und kräftebasierten Modellen ist nicht eindeutig. Es gibt zelluläre Automaten (Köster *et al.*, 2011), die als deterministisch einzuordnen sind, da aufgrund der Regelsets und der Navigationsfelder die Schritte eindeutig vorgegeben sind. Dennoch enthält das Modell stochastische Elemente, die beispielsweise ermöglichen, eine nicht optimale Wahl bezüglich des Regelsets und der Potenziale zu treffen. Vielmehr kann der nächste Schritt durch „Würfeln“ bestimmt werden kann (z.B. zweitbeste Option). Ebenso sind Modelle, die zelluläre Automaten mit kräftebasierten Modellen kombinieren (z.B. Huang & Guo, 2008, Kretz, 2009, Köster *et al.*, 2010), sowohl regelbasiert als auch kräftebasiert. Die Regeln basieren auf den zugrundeliegenden Potentialen, die sich aus den Kraftfeldern ergeben.

### 2.3 Bewertung der Simulationsmodelle bezüglich der Modellierungsebenen

Die im letzten Abschnitt vorgestellten Simulationsmodelle umfassen ein weites Spektrum. Dabei werden die in Abbildung 2.1 dargestellten Modellierungsebenen in unterschiedlicher Ausprägung einbezogen.

Makroskopische Modelle fokussieren sich im Allgemeinen auf Gesamtsituationen und finden ihren Einsatz daher vor allem in der Berechnung von unteren Schranken von Evakuierungszeiten sowie zur Identifizierung von durchschnittlichen Dichten und Geschwindigkeiten. Sie haben Vorteile, wenn genaue Angaben über einzelne Parameter, wie die Demographie der zu simulierenden Fußgänger oder Gruppengrößen, nicht bekannt sind. Da nur aggregierte Werte angenommen werden, können mit diesen Modellen zumindest Trends wie z.B. Staus vor Engstellen angenähert werden. Die Lokomotionsebene sowie die taktische Ebene werden nur rudimentär modelliert, da Personen nicht individuell betrachtet werden. Bei Netzwerkflussmodellen kann die taktische Ebene durch Beeinflussung von Kapazitäten auf Kanten grundlegend modelliert werden. Die Verhaltensebene wird in makroskopischen Modellen grundsätzlich nicht betrachtet.

Mikroskopische Modelle können die einzelnen Ebenen der Modellierungsaspekte im Allgemeinen besser und detaillierter abbilden.

Auch wenn bei Kräftenmodellen der Schwerpunkt auf der Lokomotionsebene liegt, können diese durch Navigationsgraphen erweitert werden, um die taktische Ebene abzubilden. Die Verhaltensebene ist durch reine Kräftenmodelle nicht abgedeckt, kann aber durch die Einführung von zusätzlich Kräften bis zu einem gewissen Grad modelliert werden. Da sich Fußgänger in dichten Menschenmassen nicht frei bewegen können und damit auch keine Entscheidungsfreiheit besitzen, eignen sich Kräftenmodelle sehr gut, um dichte Menschenmassen zu simulieren.

Auf zellulären Automaten basierende Modelle bilden ebenfalls die Lokomotionsebene ab und können zudem mit Navigationsgraphen zur Abbildung der taktischen Ebene erweitert werden. Das Abbilden der Verhaltensebene kann durch Regeln widerspiegelt werden, aber durch den Aufbau eines gesamtheitlichen Regelsets kann dieses sehr komplex werden. Diese Art von Modellen kann aufgrund ihrer Geschwindigkeit verwendet werden, um eine große Anzahl von Fußgängern zu simulieren, auch bei geringen Dichten. Bei zu großen Dichten sind die Modelle durch ihre räumliche Diskretisierung nur begrenzt einsetzbar.

Verhaltensbasierte Modelle bilden die Lokomotionsebene meist ebenfalls durch Kräfte ab, die jedoch nicht global, sondern lokal definiert sind. Die taktische Ebene ist in den Verhaltensregeln integriert. Der Fokus bei diesen Modellen liegt auf der Abbildung von Verhalten. Daher wird die Verhaltensebene mit diesen Modellen sehr gut abgebildet. Vor allen Abläufe, wie sie an öffentlichen Plätzen wie Flughäfen, Bahnhöfen oder auch Einkaufszentren vorkommen, können gut modelliert werden.

Zusammenfassend lässt sich sagen, dass es eine breite Variation an Fußgänger-simulationsmodellen gibt, die alle Stärken und Schwächen aufweisen. Es hängt stark vom Zweck der Simulation ab, welches Modell am besten geeignet ist.

In dieser Arbeit liegt der Fokus auf der taktischen Ebene. Es werden Konzepte zur Abbildung der Wegefindung bei Fußgängern vorgestellt. Das Modell basiert auf einem zellulären Automaten in Kombination mit Navigationsfeldern. Es wurde gewählt, da es in der Lage ist, eine große Anzahl von Fußgängern in Realzeit

zu simulieren. Es können große Bereiche modelliert werden, was gerade zur Abbildung von Navigationsverhalten relevant ist. Zudem ist dieses Modell robust, so dass es zuverlässig zur Modellierung der Lokomotionsebene eingesetzt werden kann.

Zur Abbildung der taktischen Ebene wird dieses Modell mit einem Navigationsgraphen erweitert, auf dem verschiedene Arten der Wegefindung implementiert werden. Die Umsetzung der taktischen Ebene ist nicht beschränkt auf dieses Modell, sondern kann auf beliebige andere mikroskopische Simulationsmodelle übertragen werden.

Im nächsten Kapitel wird dieses Modell im Detail beschrieben.



## Kapitel 3

# Lokomotionsebene des Simulationsmodells<sup>1</sup>

Das in dieser Arbeit verwendete Modell ist mikroskopischer Natur und besteht aus zwei Modellierungsebenen. Die Lokomotionsebene wird mittels eines zellulären Automaten in Kombination mit Navigationsfeldern abgebildet. Die taktische Ebene formt ein Navigationsgraph, der die Grundlage für die Wegfindung der Fußgänger bildet. Eine Übersicht des Gesamtmodells ist in Abbildung 3.1 gegeben. Die einzelnen Schichten werden in Abschnitt 3.2 und Kapitel 4 beschrieben. Dabei liegt ein besonderer Fokus auf der taktischen Ebene, da diese den Schwerpunkt dieser Arbeit bildet.

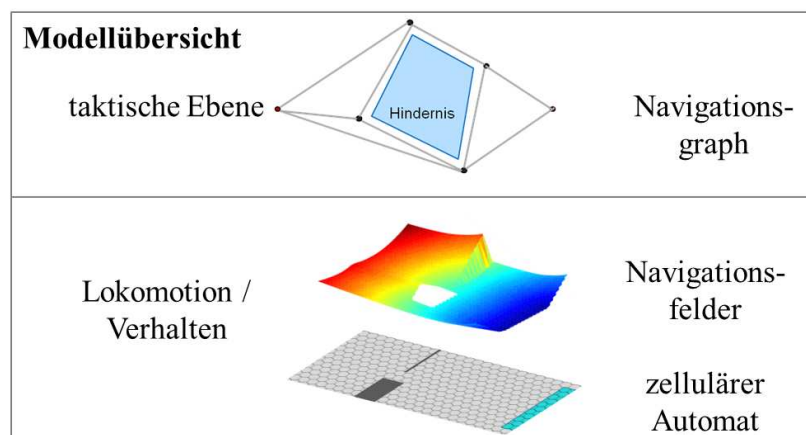


Abbildung 3.1: Die zwei Modellierungsebenen des Simulationsmodells

<sup>1</sup>Die Lokomotionsebene dieses Modells wurde von der Siemens AG entwickelt und im Rahmen des BMBF-Projekts REPKA zur Verfügung gestellt

### 3.1 Elemente eines Szenarios

Bevor auf die Beschreibung der einzelnen Modellierungsebenen eingegangen wird, werden zunächst die Grundelemente eines Simulationslaufs vorgestellt. Jeder Simulationslauf besteht aus einem Szenario sowie einem Parameterset, die zusammen den zu simulierenden Grundsachverhalt abbilden.

Ein Szenario besteht aus folgenden Grundelementen:

- eine oder mehrere Quellen
- ein oder mehrere Ziele
- kein oder mehrere Hindernisse

Die Geometrie eines zu simulierenden Gebiets wird durch Hindernisse abgebildet. Hierbei werden offene (z.B. Wände) oder geschlossene Polygonzüge (Gegenstände, bauliche Elemente) verwendet. In den Quellen werden die Fußgänger erzeugt und in den Zielen werden die Fußgänger aus der Simulation entfernt. Jeder Quelle des Szenarios wird mindestens ein Ziel zugewiesen. Werden einer Quelle mehrere Ziele zugewiesen, so werden diese mit vorgegebenen Wahrscheinlichkeiten gewichtet. Damit laufen alle Fußgänger während der Simulation zielorientiert, also in Richtung eines ihnen zugewiesenen Ziels. Zudem kann bestimmt werden, wie viele Fußgänger pro Sekunde in einer Quelle generiert werden. Für Ziele kann ebenfalls die Kapazität bestimmt werden, d.h. wie viele Fußgänger pro Sekunde aus der Simulation innerhalb welchen Zeitraums entfernt werden können.

Zusätzlich zum Szenario wird ein Set von Simulationsparametern definiert. Die wichtigsten Parameter umfassen:

- Geschwindigkeitsverteilung
- Platzbedarf einer Person
- maximale Dichten
- Simulationszeit
- Gruppengrößenverteilung

Die generierten Fußgänger laufen ohne Beeinträchtigung mit einer vorgegebenen *free-flow* Geschwindigkeit, die stochastisch über die Fußgänger verteilt ist. Den Untersuchungen von Weidmann (1993, S.43) folgend werden die Fußgänger im Standardfall mit einer Durchschnittsgeschwindigkeit von 1.34 m/s erzeugt. Die Geschwindigkeitsverteilung entspricht einer Gaußverteilung, deren Mittelwert mit Hilfe des Parameters eingestellt werden kann. Die Standardabweichung beträgt standardmäßig 0.26 und kann ebenfalls durch einen Parameter verändert werden. Abhängig von der Dichte werden die Fußgänger entsprechend einer Geschwindigkeits-Dichte-Relation abgebremst, die in Weidmann (1993, S.62) definiert ist. Daraus resultiert ein Fundamentaldiagramm, welches in der Verkehrstechnik sehr oft eingesetzt wird. Dieses Diagramm setzt den Fluss in Relation zur Dichte. Fluss bezeichnet hierbei die Anzahl der Personen [P], die pro Sekunde [s] eine Messlinie mit definierter Breite [m] überschreiten und wird in [P/ms] angegeben. In Abbildung 3.2 ist das Diagramm nach Weidmann, der das Fundamentaldiagramm an Fußgänger angepasst hat, dargestellt.

Der Platzbedarf einer Person bestimmt die Größe des darunterliegenden Zellgit-

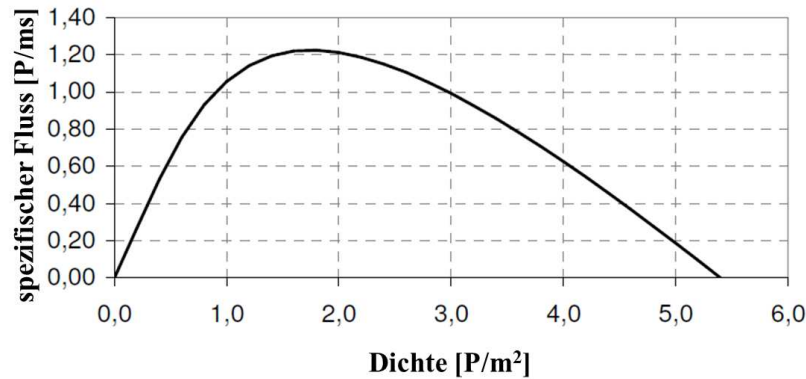


Abbildung 3.2: Fundamentaldiagramm nach Weidmann

ters des zellulären Automaten, der in Unterabschnitt 3.2.1 genauer beschrieben wird. Ebenfalls aus Weidmann (1993, S.25) entnommen, wird der minimale Platzbedarf mit  $0.15 m^2/P$  beschrieben. Aus dem Platzbedarf einer Person leitet sich direkt die Dichte ab, die sich hier als maximal  $6.6 P/m^2$  ergibt. Auch hier können Anpassungen entsprechend des zu simulierenden Szenarios vorgenommen werden. Da bei Europäern eine Maximaldichte von  $5.4 P/m^2$  angenommen wird (Weidmann, 1993, S.87), ist diese Einstellung standardmäßig gegeben.

Die Gesamtsimulationszeit kann für jedes Szenario individuell angegeben werden.

Die Verteilung der Gruppengrößen wird ebenfalls szenariospezifisch eingestellt. Da die Gruppengröße und -verteilung abhängig des zu simulierenden Grund Sachverhalts sehr stark variieren kann, müssen diese Parameter pro Szenario ermittelt und eingestellt werden. Standardmäßig wird von Einzelpersonen ausgegangen.

Zusätzlich zu den hier aufgeführten Parametern gibt es eine Reihe weiterer Größen, die der Anwender justieren kann. Da diese für das Verstehen des Modells nicht relevant sind, wird hier nicht weiter darauf eingegangen. Eine Auflistung ist in Hartmann (2010b) und Köster *et al.* (2010) zu finden.

## 3.2 Modellierung der Lokomotionsebene

Die Lokomotionsebene kann in zwei Komponenten unterteilt werden, einerseits in die räumliche und zeitliche Diskretisierung, andererseits in Navigationsfelder, die die Werte für das räumliche Zellgitter liefern. Im Folgenden werden die beiden Schichten detailliert beschrieben.

### 3.2.1 Diskretisierung des Raums und der Zeit: Zellulärer Automat

In dieser Arbeit wird ein reguläres Gitter zur Diskretisierung des Raums verwendet. In Kombination mit der zeitlichen Diskretisierung und einem Regelset formt diese Schicht ein zellulärer Automat. Folgende Definition ist an die in der Litera-

tur gegebenen Definitionen aus Bungartz *et al.* (2009), Malcher (2004), Mainzer & Chua (2012) sowie die im Kontext von Fußgängersimulation gegebene Definition (Pelechano *et al.*, 2008) angelehnt.

**Definition 3.1** *Ein zellulärer Automat ist definiert durch:*

- einen diskreten Zellraum  $R^n$  ( $n = \text{Dimensionalität des Raums}$ ), unterteilt in Zellen gleicher Geometrie
- eine Zustandsmenge  $Q$
- eine endliche Nachbarschaft  $N$
- diskrete Zeitschritte  $t$
- und eine lokale Überföhrungsfunktion  $\delta : Q^N \mapsto Q$

Hier wird der zweidimensionale Zellraum  $R^2$  betrachtet. Jede Zelle besitzt eine endliche Anzahl von Nachbarzellen. Der Automat nimmt zu jedem Zeitpunkt einen Zustand aus der Zustandsmenge  $Q$  an. Zustände werden erzeugt, in dem Überföhrungsfunktionen in jedem Zeitschritt auf die Zellen angewendet werden. Dabei ist die Besonderheit, dass der Zustand einer Zelle nicht nur von ihrem eigenen aktuellen Zustand abhängt, sondern auch von den Zuständen der Nachbarzellen in der Nachbarschaft  $N$ . Die Überföhrungsfunktionen werden auf jede Zelle in  $R$  im Zeitschritt  $t - 1$  angewendet, um einen neuen Gesamtzustand aus  $Q$  in Zeitschritt  $t$  zu generieren.

Übertragen auf das Gebiet der Fußgängersimulation wird der zelluläre Automat wie folgt konfiguriert: Die Geometrie des Szenarios wird in ein zelluläres Gitter unterteilt. Es existieren unterschiedliche Geometrien für die Zellform. Die Größe einer Zelle wird so gewählt, dass zu jedem Zeitpunkt eine Zelle von genau einer Person belegt wird. Am häufigsten werden quadratische Zellen (z.B. Blue & Adler, 1999, Burstedde *et al.*, 2001, Varas *et al.*, 2007, Yamamoto *et al.*, 2007) oder hexagonale Zellen (z.B. Kinkeldey, 2003, Klein *et al.*, 2010) für die Diskretisierung verwendet. Die Größe der Nachbarschaft  $N$  hängt von der Geometrie der Zellen ab (siehe Abbildung 3.3). Bei quadratischen Zellen sind zwei unterschiedliche Nachbarschaften definiert. In der von-Neumann-Nachbarschaft gibt es vier Nachbarzellen, die alle den gleichen Abstand zur Zelle haben. Die Moore-Nachbarschaft definiert acht Nachbarn, deren Abstände unterschiedlich groß sind. Bei hexagonalen Zellen gibt es sechs Nachbarzellen, die alle den gleichen Abstand besitzen.

In dem hier beschriebenen Modell werden hexagonale Zellen verwendet, um eine größtmögliche Anzahl von Bewegungsrichtungen abbilden zu können, ohne eine Unterscheidung in der Abstandsberechnung der Nachbarzellen machen zu müssen. Zudem nähert ein Hexagon die Form des menschlichen Körpers eher an als ein Quadrat. Jede Zelle  $c$  des Automaten kann zwei Zustände  $z_c$  annehmen, frei oder belegt; belegt durch einen Fußgänger oder ein Hindernis. Zudem wird jeder Zelle  $c$  ein Wert  $v_c$  zugewiesen, der aus Navigationsfeldern ermittelt wird (siehe Unterabschnitt 3.2.2). Die Überföhrungsfunktionen  $\delta$  des Automaten sind sehr einfach gehalten: Abhängig vom Wert der Nachbarzelle sowie vom Zustand der

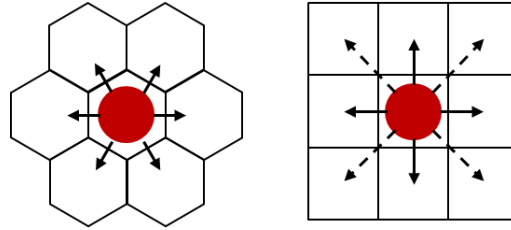


Abbildung 3.3: Links: Hexagonale Zellgeometrie: Die Nachbarschaft einer Zelle besteht aus 6 Zellen. Rechts: Quadratische Zellgeometrie: Die durchgezogenen Linien zeigen die von-Neumann-Nachbarschaft, die durchgezogenen Linien sowie die gestrichelten Linien bilden die Moore-Nachbarschaft.

Nachbarzelle (frei/belegt), wird folgende Schaltregel angewandt. Sei  $z_c \in Z$ ,  $Z = \{\text{'frei'}, \text{'belegt'}\}$  der Zustand einer Zelle  $c$  und  $v_c$  der Wert der Zelle  $c$ . Sei weiterhin  $c_i \in N_f \subseteq N$  die Menge aller Nachbarn mit Zustand  $v_{c_i} = \text{'frei'}$ . So gilt:

$$\text{wenn } v_{c_i} = \min_{\forall c_n \in N_f} (v_{c_n}) \Rightarrow z_{c_i} = \text{'belegt'} \wedge z_c = \text{'frei'}.$$

Die Anwendung der Schaltregeln erfolgt sequentiell über die Liste der Fußgänger: Jeder Fußgänger kennt die Zelle, auf der er sich befindet und somit wird auf jede durch einen Fußgänger besetzte Zelle obige Regel angewandt. Dies ist anders als bei konventionellen zellulären Automaten; dort wird über alle Zellen iteriert. Begonnen wird mit dem „ältesten“ Fußgänger, also demjenigen, der als erstes erzeugt wurde.

Die Größe einer Zelle ist definiert mit einem Umkreisradius  $r$  von 0.267 m. Daraus resultiert eine maximale Dichte von 5.4 Pers/ $m^2$ , womit an die Vorgaben von Weidmann (1993, S.87) bei der Dimensionierung von Fußgängeranlagen angelehnt wird, der dies als maximale Dichte angibt. Dies kann – wie in Abschnitt 3.1 erläutert – umgestellt werden.

Die zeitliche Diskretisierung wird so gewählt, dass abhängig von den eingestellten Geschwindigkeiten die Maximalgeschwindigkeit eines Fußgängers abgebildet werden kann. Als einzustellende Parameter dienen eine Durchschnittsgeschwindigkeit und die Standardabweichung der Geschwindigkeit, aus der die maximale Geschwindigkeit  $v_{max}$  berechnet wird. Die Länge eines Zeitschritts  $t$  wird somit wie folgt berechnet:  $t = 2 \cdot r / v_{max}$ . Dabei bezeichnet  $2 \cdot r$  den Abstand zwischen zwei Zellen.

Eine weitergehende Beschreibung dieser Ebene ist in Köster *et al.* (2010) zu finden.

### 3.2.2 Modellierung des Bewegungsverhaltens: Navigationsfelder

Ziel der Lokomotionsebene ist die Abbildung von möglichst realistischem Laufverhalten auf dem Zellgitter des zellulären Automaten. Dazu werden in dem hier beschriebenen Verfahren Navigationsfelder eingesetzt. Diese liefern die Zellwerte

$v_c$  für jede Zelle des Automaten. Zur Ermittlung dieser Werte wird ein Potentialmodell implementiert. Eine Modellierung mittels Potentialen ist einer Modellierung mittels Kraftfeldern äquivalent. Die entsprechenden Kraftfelder können über Gradientenbildung gewonnen werden (Meschede, 2006). Es wirken anziehende Kräfte in Richtung Ziel des Fußgängers  $FG$ , sowie abstoßende Kräfte, die von Hindernissen und von anderen Fußgängern ausgehen, die sich in der näheren Umgebung eines Fußgängers  $FG$  befinden. Die Ermittlung dieser Kräfte erfolgt ähnlich des Social Force Models nach Helbing & Molnár (1995). Die folgenden Beschreibungen der einzelnen Potentiale/Kräfte sind an Hartmann (2010b) angelehnt und dort in ausführlicher Form zu finden.

Pro Zelle wird ein Gesamtpotential  $\Phi_{ges}$  berechnet:

$$\Phi_{ges} = \Phi_{obs} + \Phi_{dest} + \Phi_{ped}$$

Im Folgenden werden die einzelnen Potentiale beschrieben.

#### Hindernispotential $\Phi_{obs}$

Die Abstoßung von Hindernissen wird als eine Kraft mit kurzer Reichweite modelliert. Das Potential hat eine einzustellende Reichweite, die zwischen einem und zwei Metern gewählt werden sollte und fällt linear ab. Das Potential an der Position  $(x, y)$  wird wie folgt ermittelt:

$$\Phi_{obs}(x, y) = \max\left(0, \Phi_{obs_{max}} \cdot \left(1 - \frac{d_{obs}(x, y)}{d_{max}}\right)\right)$$

Dabei bezeichnet  $d_{obs}(x, y)$  den Abstand vom Hindernis,  $d_{max}$  die maximale Reichweite des Potentials und  $\Phi_{obs_{max}}$  den maximalen Abstoßungswert des Hindernisses.

#### Zielpotential $\Phi_{dest}$

Die Anziehungskraft zum Ziel wird abgebildet, indem den Zellen Abstandswerte zum Ziel gegeben werden. Im einfachsten Fall kann auch hier ein einfaches Potential angewandt werden. Dies führt allerdings zu Problemen bei Geometrien, in denen keine Sichtverbindung zwischen Quelle und Ziel gegeben ist. Dort können lokale Minima auftreten, wie in Abbildung 3.4 beispielhaft dargestellt. Um das Potential auch für nicht-sichtbare Ziele zu ermitteln, werden sogenannte Flutungsalgorithmen (auch Flutfüllungsverfahren (engl. *flood filling*) genannt) eingesetzt. Diese zeichnen sich dadurch aus, dass ausgehend vom Ziel ein Gebiet (in diesem Fall das Zellgitter) geflutet wird in Analogie zu einer Welle, die sich in einem Raum ausbreitet.

Die Zellwerte können zum Beispiel durch Anwendung des Dijkstra Algorithmus (Dijkstra, 1959) ermittelt werden: Gegeben sei eine Menge  $N$  von Zellen, die  $n$  Schritte vom Ziel entfernt sind. Die Abstände der Nachbarn werden berechnet, in dem ein Schritt auf den bisherigen Wert addiert wird. (Abbildung 3.5). Mit diesem Verfahren wird jedoch nicht der euklidische Abstand zwischen zwei Zellzentren betrachtet, sondern der Abstand in Bezug auf die Anzahl der Schritte zwischen zwei Zellen. Dies kann zu ungewünschtem Laufverhalten führen. In Abbildung 3.5 ist zu sehen, dass der südliche Weg bezüglich dieser Metrik genauso

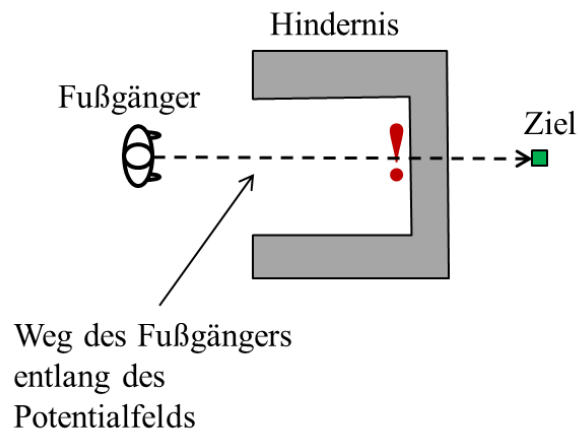


Abbildung 3.4: Bei einem einfachen Potentialmodell kann es passieren, dass ein lokales Minimum auftritt, aus dem der Fußgänger nicht herausfindet. Bei Verwendung von euklidischer Metrik entsteht ein lokales Minimum innerhalb des u-förmigen Hindernisses.

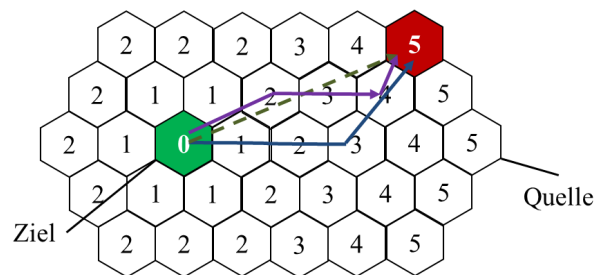


Abbildung 3.5: Wenn die Anzahl der Schritte als Abstandswerte genommen werden, so kann es passieren, dass ein vermeintlich kürzester Weg bezogen auf diese Metrik, eine erhebliche Abweichung zum kürzesten Weg nach euklidischer Metrik (Luftlinie) aufweist. Der untere Weg weicht erheblich vom kürzesten Weg entlang der Luftlinie (gestrichelte Linie) ab, auch wenn er bezüglich der Schrittlänge gleich ist.

gut ist wie der Weg, der näher entlang der Luftlinie zum Ziel verläuft. Dies führt zu „eckigem“ Laufverhalten und dazu, dass unnatürliche Wege gelaufen werden, auf denen Fußgänger mehr Zeit benötigen, um zum Ziel zu gelangen.

Um dies zu umgehen, wird die Ausbreitungsgeschwindigkeit der Welle betrachtet. Diese bewegt sich mit einer Normalengeschwindigkeit von  $F(x)$  fort. Dies bietet den Vorteil gegenüber der Anzahl der Schritte, die im oben beschriebenen Verfahren ermittelt werden, da direkt die Reisezeit miteinfließt. In Abbildung 3.6 ist dies schematisch dargestellt. Es werden Ankunftszeiten statt Abstände berech-

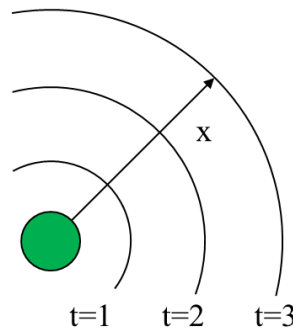


Abbildung 3.6: Ausbreitung einer Welle mit konstanter Geschwindigkeit zu unterschiedlichen Zeitschritten

net. Diese Ankunftszeiten  $T(x)$  können mathematisch mit der Eikonalgleichung (Gleichung 3.1) beschrieben werden:

$$\begin{aligned} F(x)|\nabla T(x)| &= 1 \text{ in } \Omega, \\ T(x) &= 0 \text{ in } \Gamma. \end{aligned} \quad (3.1)$$

Die allgemeine Eikonalgleichung legt mit  $\Omega$  das Gebiet der Wellenausbreitung sowie mit  $\Gamma \in \Omega$  den Reisezeit von 0 im Ursprung und eine von 0 verschiedene Reisezeit im Ausbreitungsraum.

Die Eikonalgleichung angewandt auf den 2-dimensionalen Raum wird wie in Gleichung 3.2 beschrieben:

$$F(x, y)^2 \cdot \left[ \left( \frac{\partial T(x, y)}{\partial x} \right)^2 + \left( \frac{\partial T(x, y)}{\partial y} \right)^2 \right] = 1 \quad (3.2)$$

mit  $F(x, y) = 0$  für  $(x, y) \in \Gamma$ .  $\Gamma$  bezeichnet das Ziel, von dem aus die Welle propagiert wird und beinhaltet alle Hindernisse des Szenarios. Für das restliche Gebiet  $\Omega$  wird  $F(x, y) = 1$  gesetzt. Die Ankunftszeit der Welle am Punkt  $(x, y)$  entspricht dem kürzesten Abstand dieses Punktes bezüglich der euklidischen Metrik. Die Richtung des kürzesten Wegs ist gegeben durch den Gradienten

$$\left( \frac{\partial T(x, y)}{\partial x} \frac{\partial T(x, y)}{\partial y} \right). \quad (3.3)$$

Mit einer gegebenen räumlichen Diskretisierung von  $\Omega$ , die in diesem Fall durch das Zellgitter des hexagonalen Automaten gegeben ist, bietet die Fast Marching



Methode (Sethian, 1999) eine sehr effiziente Möglichkeit, die Eikonalgleichung numerisch zu lösen. Dazu werden die Gitterpunkte  $\vec{p}_i$  des Zellraums in drei verschiedene Zustände unterteilt: weit entfernt (keine Reisezeit  $T(\vec{p}_i)$  bisher zugewiesen), vorläufig ( $T(\vec{p}_i)$  ist zugewiesen, aber noch nicht fixiert) und fixiert ( $T(\vec{p}_i)$  zugewiesen und unveränderlich). Alle vorläufig berechneten Gitterpunkte werden in einer Kandidatenliste gespeichert. Beginnend mit der Diskretisierung der initialen Front wird der Punkt  $\vec{p}_i$  mit kleinster Reisezeit  $T(\vec{p}_i)$  aus der Kandidatenliste gewählt und fixiert (siehe Abbildung 3.7a). Alle Nachbarn der Punkte  $\vec{p}_i$  aus der Kandidatenliste werden aktualisiert, d.h.  $T$  wird neu berechnet. Zudem werden alle Nachbarn initialisiert, die weiter entfernt sind, d.h.  $T$  wird ermittelt und die Knoten wandern in die Kandidatenliste (siehe Abbildung 3.7b). Diese Schritte werden für den nächsten Punkt aus der Kandidatenliste mit geringster Reisezeit  $T$  wiederholt (siehe Abbildung 3.7c und Abbildung 3.7d). Der Algorithmus stoppt, wenn die Kandidatenliste leer ist oder ein Stoppkriterium erfüllt ist, z.B. die Zielzelle erreicht wurde (siehe Abbildung 3.7e). Dem Prinzip nach entspricht der Algorithmus einem klassischen Sortieralgorithmus, der die Kandidatenpunkte mit kleinster Reisezeit  $T$  iterativ findet, diese in die Menge der fixierten Punkte verschiebt und die Nachbarn aktualisiert. Die Verwendung eines effizienten Sortieralgorithmus vorausgesetzt (z.B. Heap Sort), beträgt die Rechenzeit  $n \log n$  (Sethian, 1999), wobei  $n$  die Anzahl der Gitterpunkte bezeichnet.

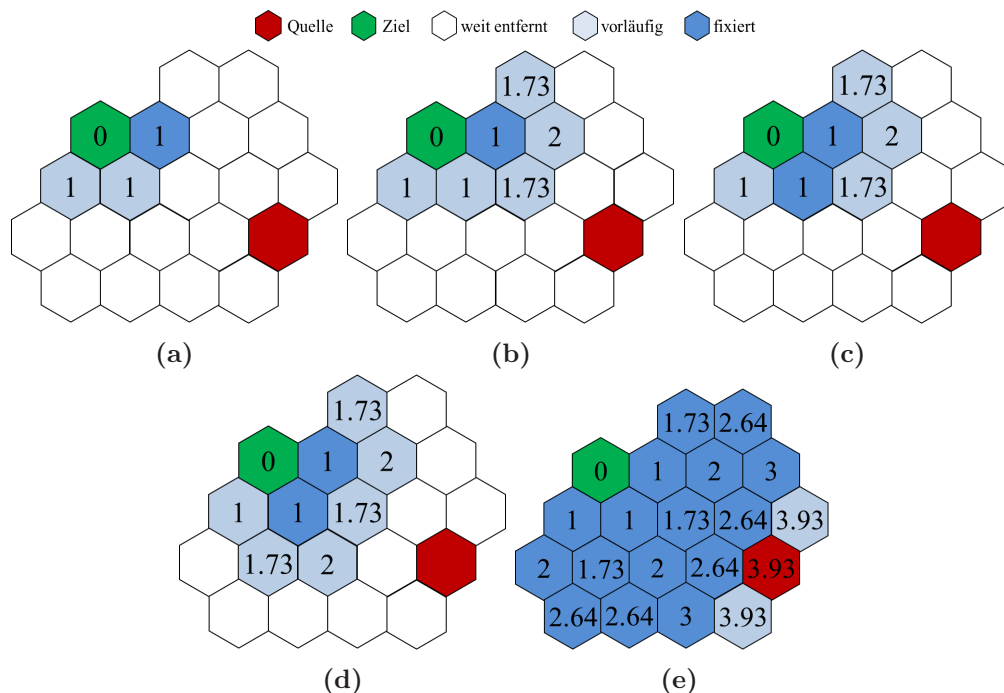


Abbildung 3.7: Schematischer Ablauf des Flutungsalgorithmus: Ausgehend vom Ziel wird in Richtung Quelle geflutet. Da zu Beginn alle Nachbarzellen des Ziels den gleichen Wert besitzen, wird die obere Zelle exemplarisch gewählt. Je nach Reihenfolge des Durchlaufens der Zellen kann auch eine andere Zelle gewählt werden.

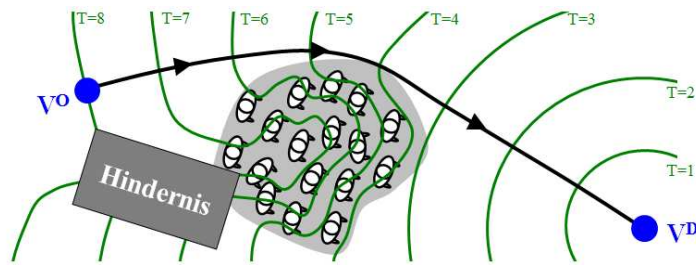


Abbildung 3.8: Fortbewegung der Welle von  $V^D$  zu  $V^O$ . Die Ankunftszeiten der Welle geben den schnellsten Weg vor. (Urheber: Dr. Dirk Hartmann)

Eine ausführliche Beschreibung der Abstandsberechnung sowie das Miteinbeziehen einer Richtungskorrektur und das Variieren von Geschwindigkeiten ist in Hartmann (2010a) zu finden.

Damit betrachtet die Welle ausschließlich nicht passierbare (statische) Hindernisse, die sich auf dem Ausbreitungsgebiet befinden. Um auch Fußgänger, die sich auf dem Gebiet bewegen, einzubeziehen, werden diese bei der Berechnung der Ausbreitungsgeschwindigkeit mit in Betracht gezogen. Auf allen Zellen, auf denen sich ein Fußgänger befindet, wird eine geringere Reisegeschwindigkeit angenommen, also  $F(x, y) \leq 1$ . Da sich Fußgänger bewegen, muss dieses Navigationsfeld ständig aktualisiert werden. Es wird deshalb in regelmäßigen Abständen neu berechnet. Ein Beispiel für die Fortbewegung der Welle unter Miteinbeziehung der Fußgänger ist in Abbildung 3.8 dargestellt. Die Welle wird von den Fußgängern verlangsamt. Deshalb ist der Weg um den Pulk schneller. Für jede freie Zelle wird  $F(\vec{x}) = 1$  gewählt, für Zellen, auf denen sich ein Fußgänger befindet, wird  $F(\vec{x}) = 1/(1 + \delta)$  mit dem penalty Faktor  $\delta > 0$  gesetzt. Ein Vergleich zwischen statischen und dynamischen Navigationsfeldern ist in Abbildung 3.9 zu sehen. Hier erkennt man deutlich, dass sich ohne das Einbeziehen von Fußgängern ein Stau auf der westlichen Seite des Hindernisses bildet und die Breite des Ganges nicht ausgenutzt wird. Mit dem Einbeziehen der Fußgänger ändert sich das Bewegungsmuster dahingehend, dass die Breite des Ganges besser genutzt wird, was einem natürlicheren Muster entspricht.

Eine weitere Möglichkeit besteht darin – statt der Addition des Hindernispotentials auf das Gesamtpotential – auch das Hindernispotential direkt mit in die Flutung einfließen zu lassen. Dabei kann analog zur Einbeziehung der Fußgänger vorgegangen werden. Die Methodik ist in Hartmann (2010b) detailliert beschrieben.

Es hat sich gezeigt, dass diese Methode anderen Methoden überlegen ist. Zum Einen, da keine Sichtverbindung zwischen Quelle und Ziel vorhanden sein muss, und zum Anderen durch die exaktere Annäherung der euklidischen Metrik bei der Zellwertberechnung. Zudem kann dieses Verfahren in Kombination mit einem Navigationsgraphen sehr effizient eingesetzt werden. Statt ein globales Feld zu generieren, kann pro Kante ein Feld generiert werden, wodurch die Aktualisierung

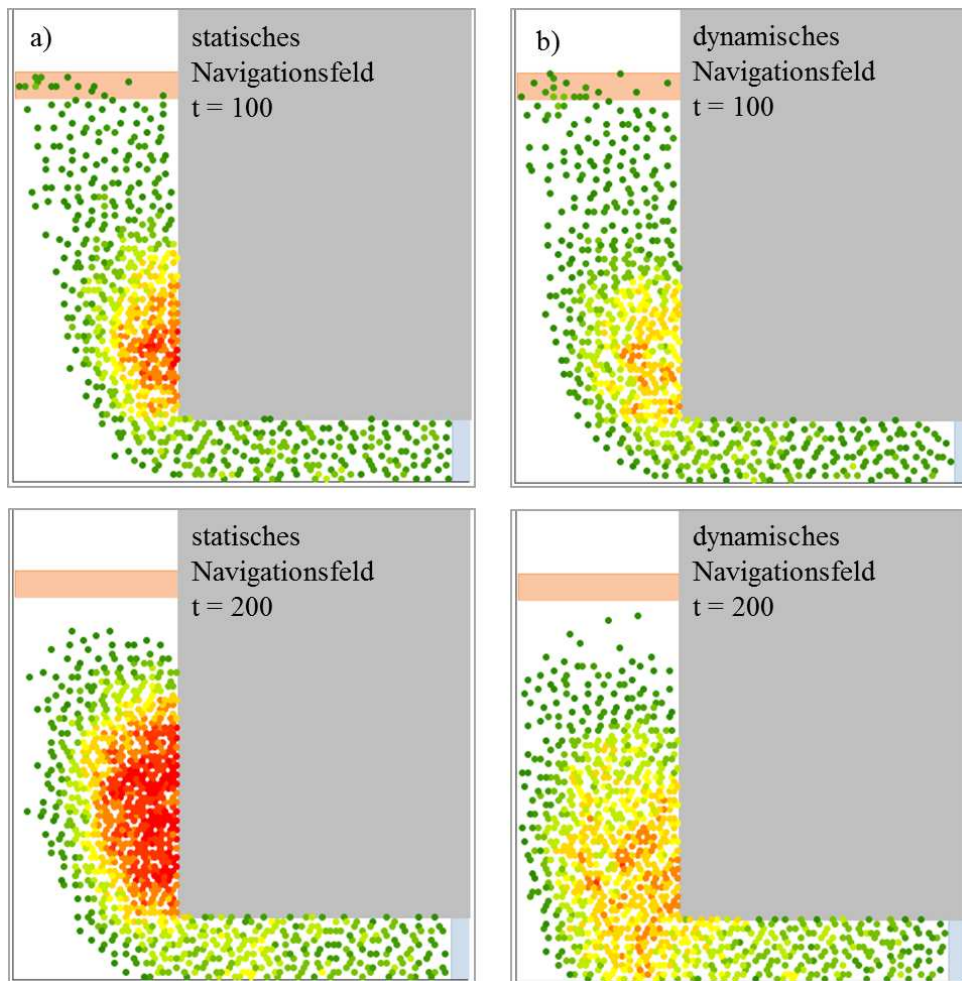


Abbildung 3.9: Screenshots der Simulation nach 100 und 200 Sekunden. Links mit statischen Navigationsfeldern, rechts mit dynamischen Navigationsfeldern

der dynamischen Felder modularer durchgeführt werden kann; es müssen nur diejenigen Felder aktualisiert werden, auf denen sich Fußgänger befinden. Detailliert wird darauf in Unterabschnitt 6.2.1 eingegangen.

### Personenpotential $\Phi_{ped}$

Das Personenpotential dient dazu, Kollisionen von Fußgängern zu vermeiden. Daher fließen die Fußgänger in unmittelbarer Nähe in die Bewertung der Nachbarzellen mittels des Personenpotentials ein. Auf jede Zelle, auf der sich ein Fußgänger innerhalb eines festzulegenden Radius befindet, wird Abstoßungspotential addiert. Zu beachten ist, dass dieses Personenpotential von den berücksichtigten Fußgängern im Zielpotential zu unterscheiden ist. Während es sich beim Zielpotential um eine globale Flutung des Gebiets unter Miteinbeziehung von weiter entfernten Fußgängern handelt, werden mit dem Personenpotential nur die Personen in unmittelbarer Nähe in Betracht gezogen, da nur diese den nächsten Schritt eines Fußgängers beeinflussen.

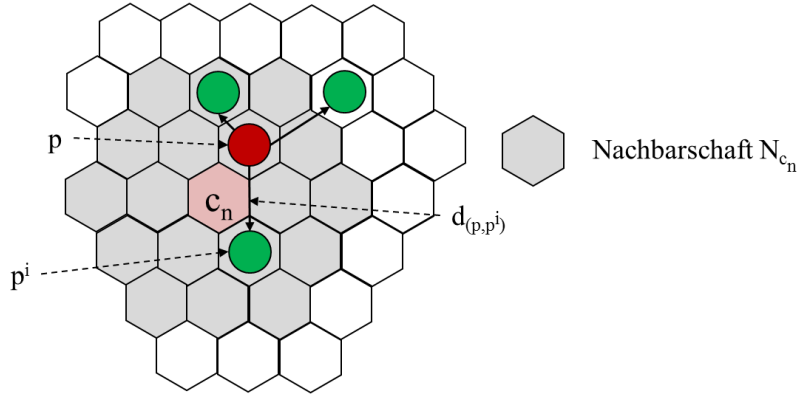


Abbildung 3.10: Beispiel für die Berechnung des Personenpotentials: Die grau markierten Zellen entsprechen  $N_{c_n}$ , exemplarisch ist hier der Abstand für die Potentialberechnung für Zelle  $c_n$  von Fußgänger  $p$  zu  $p_i$  eingezeichnet

Fußgänger, die in dieselbe Richtung laufen, beeinflussen die Bewegung des betrachteten Fußgängers in einem geringeren Maße als entgegenkommende oder kreuzende Fußgänger. Daher werden alle Fußgänger, die sich in dieselbe Richtung wie der betrachtete Fußgänger bewegen, geringer gewichtet als Fußgänger mit unterschiedlicher Bewegungsrichtung.

Die Berechnung pro benachbartem Fußgänger mit der Position  $(p_x^i, p_y^i)$  in der Nachbarschaft der Zelle  $c_n$ ,  $p^i \in N_{c_n}$  für den Fußgänger mit der Position  $(p_x, p_y)$  wird wie folgt durchgeführt:

$$\phi_p^{c_n}((p_x, p_y), (p_x^i, p_y^i)) = c_{amplitude} \cdot e^{\left(-\frac{d((p_x, p_y), (p_x^i, p_y^i))}{c_{width}}\right)}$$

Dabei gibt  $c_{amplitude}$  die Amplitude und  $c_{width}$  die Reichweite des Potentials vor. Beide Parameter sind entsprechend des Platzbedarfs und der Schrittweite einzustellen.

Das Gesamtpotential einer Zelle  $c_n$  berechnet sich dann aus der Summe aller Einzelpotentiale der Nachbarn  $(p_x^i, p_y^i)$ :

$$\Phi_{p_{ges}}^{c_n} = \sum_{\text{alle Nachbarn } p^i \text{ der Zelle } c_n} \phi_p((p_x, p_y), (p_x^i, p_y^i))$$

Zur Veranschaulichung ist ein Beispiel in Abbildung 3.10 gegeben.

Der zelluläre Automat in Kombination mit den unterschiedlichen Kräften formt die Lokomotionsebene. Damit können lokale Aspekte des Bewegungsverhaltens von Fußgängern wie die Lokomotion oder Kollisionsvermeidung und davon abhängige Entscheidungen gut modelliert werden. In Kapitel 4 wird genauer auf die taktische Ebene eingegangen, die globale Aspekte und deren Einfluss auf die Wegefindung bei Fußgängern berücksichtigt.

### 3.3 Zusammenfassung

In diesem Kapitel wurde der Aufbau des Simulationsmodells, das die Grundlage für diese Arbeit bildet, erläutert. Es besteht aus zwei Ebenen, der Lokomotionsebene sowie der taktischen Ebene. Die Lokomotionsebene wurde in diesem Kapitel vorgestellt, eine ausführliche Beschreibung der taktischen Ebene folgt in Kapitel 4.

Die Lokomotionsebene ist definiert durch einen zellulären Automaten, dessen Zellen Werte zugewiesen bekommen, die aus Navigationsfeldern ermittelt werden. Navigationsfelder bilden unterschiedliche Kräfte ab, von denen die Fußgänger auf ihrem Weg von einer Quelle zu einem Ziel beeinflusst werden. Dazu zählt die anziehende Kraft zum Ziel sowie abstoßende Kräfte von Hindernissen und anderen Fußgängern. Die Kombination dieser Kräfte werden als Zellwerte den einzelnen Zellen des Automaten zugewiesen. Damit kann das lokale Navigationsverhalten von Fußgängern sehr gut modelliert werden.



## Kapitel 4

# Taktische Ebene des Simulationsmodells

Zur Modellierung der taktischen Ebene wird ein Navigationsgraph eingesetzt. Dieser Graph wird automatisiert aus einer Szenariogeometrie erzeugt und dient als Grundlage für unterschiedliche Wegefindungsalgorithmen (Kapitel 6) zur Abbildung menschlichen Navigationsverhaltens. Er wird gleichzeitig dazu eingesetzt, die mikroskopische Simulation mit einem makroskopischen Netzwerkflussmodell zu koppeln (Kapitel 5).

Um einen Graphen zu erzeugen, der so ausgedünnt wie möglich bei gleichzeitig größtmöglicher Gebietsabdeckung ist, wird ein neues Verfahren vorgestellt. Die Forderung nach einem ausgedünnten Graphen resultiert aus den auf ihm implementierten Routingalgorithmen, deren Performanz von der Anzahl der Kanten und Knoten des Graphen abhängt.

Nach einem kurzen Überblick über bestehende Verfahren zur Generierung von Navigationsgraphen wird dieses Verfahren detailliert beschrieben. Zudem wird die Qualität des auf diese Weise generierten Graphen überprüft und bewertet.

### 4.1 Bekannte Ansätze zur Erzeugung von Navigationsgraphen

In der Literatur findet man unterschiedliche Ansätze, um Navigationsgraphen zu generieren. Viele Ansätze kommen aus dem Gebiet der Robotik. Dort liegt der Fokus darauf, einen Weg durch ein unbekanntes Terrain zu finden, der auf der einen Seite effizient (also kurz) ist, auf der anderen Seite möglichst weit um Hindernisse verläuft, um Kollisionen zu vermeiden. Einen detaillierten Überblick über gängige Raumzerlegungsalgorithmen (engl. *space decomposition techniques*) findet sich in Choset (2005). Im Bereich der Fußgängersimulation gibt es verschiedene Ansätze, die weitgehend aus dem Feld der Robotik übernommen wurden:

#### **Voronoidiagramme**

In Sud *et al.* (2008) und Sterlin *et al.* (2010) dienen Voronoi-Diagramme (Auren-

hammer, 1991) als Navigationsgraph. Dabei werden die Linien des Diagramms als Graphkanten und die Schnittpunkte als Graphknoten interpretiert. Charakteristisch für einen solchen Navigationsgraph ist, dass alle Kanten äquidistant zu allen umgebenden Hindernissen sind. Dies hat zur Folge, dass die darauf berechneten Wege nicht den kürzesten Wegen entsprechen, sondern in weitem Bogen um Hindernisse führen. Das reflektiert jedoch nicht menschliches Laufverhalten, da Menschen in der Regel näher um Hindernisse herum gehen (Brogan & Johnson, 2003). Eine Lösung hierzu wird in Sterlin *et al.* (2010) beschrieben. Hier wird ein so genannter „*platter lift*“ Algorithmus eingeführt, in dem sich die Fußgänger wie bei einem Schlepplift an die Kante des Voronoi-Diagramms mit variierendem Abstand hängen.

Ein weiterer Ansatz wurde in Sud *et al.* (2008) vorgestellt: Hier navigieren die Fußgänger entlang kombinierter Voronoi-Diagramme, die nicht nur Hindernisse, sondern auch andere Fußgänger enthalten. Die Überschneidung der Regionen eines Voronoi-Diagramms erster Ordnung mit einem Voronoi-Diagramm zweiter Ordnung bildet den Navigationsgraph. Die Autoren bezeichnen diesen Graph als Multiagent Navigation Graph (MaNG), der maximalen Abstand für jeden Agenten bietet.

Geraerts & Overmars (2007) verwenden das Generalisierte Voronoi-Diagramm (GVD) als Basis, fügen zusätzlich so genannte *event points* ein, die eine explizite minimale *corridor map* erzeugen. Dies ermöglicht die Definition von Regionen mit minimalem Abstand.

### Sichtbarkeitsgraph

Eine andere Möglichkeit Navigationsgraphen zu erstellen, ist der Einsatz von Sichtbarkeitsgraphen (Berg *et al.*, 2000). Dieser ist formal definiert wie folgt:

**Definition 4.1** *Ein Graph  $G(E, V)$ , mit  $E$  als Kantenmenge und  $V$  als Knotenmenge ist genau dann ein Sichtbarkeitsgraph, wenn für alle  $e = (v_1, v_2) \in E$  gilt, dass  $v_1, v_2 \in V$  in Sichtverbindung liegen.*

In Arikan *et al.* (2001) wird ein solcher Sichtbarkeitsgraph verwendet, um die Agenten in einem Szenario zu navigieren. Basierend auf dem Sichtbarkeitsgraph wird eine vorberechnete kürzeste Wegekarte gespeichert. Falls sich andere Agenten auf einem der vorberechneten Pfade befinden, muss eine Neuberechnung des Pfades erfolgen. Da die Neuberechnung sehr rechenintensiv ist, liegt der Fokus der Arbeit auf der Approximation der Positionen der Agenten, indem nur Agenten in Betracht gezogen werden, die innerhalb des sichtbaren Bereichs liegen.

Auch in Ansätzen von Gloor *et al.* (2004) und Höcker (2010) werden Sichtbarkeitsgraphen eingesetzt zur Abbildung des menschlichen Navigationsverhaltens. Der Ansatz, einen Sichtbarkeitsgraphen zu generieren, wird im Rahmen dieser Arbeit ebenfalls verfolgt. Zum Einen resultieren realistische Laufwege aus den Kanten eines Sichtbarkeitsgraphen, zum Anderen ist ein solcher Graph einfach aus der Szenariogeometrie zu erstellen. Durch die gegebene Sichtverbindung zwischen den Knoten können Navigationsalgorithmen sehr effizient angewandt werden.



### Algorithmen zur Generierung eines Sichtbarkeitsgraphen

Um einen Sichtbarkeitsgraphen zu erstellen, gibt es bereits effiziente Algorithmen, wie der *Plane Sweep* Algorithmus, der mithilfe einer Sweep Line die Umgebung scannt und jeweils Knoten verbindet, die auf dieser Linie liegen. Eine detaillierte Beschreibung ist in Choset (2005) gegeben. Dieser Algorithmus nimmt die konvexen Eckpunkte von Polygonen als Graphknoten und findet alle möglichen Kanten in einem gegebenen Gebiet.

Gloor *et al.* (2004) konstruieren einen Sichtbarkeitsgraphen, indem Knoten in einem bestimmten Abstand zu den konvexen Ecken der Hindernisse platziert werden. Pro konvexe Ecke werden drei Knoten platziert. Dieser Ansatz verhindert, dass Fußgänger zu nah um die Ecken laufen. Mit dieser Konstruktionsmethode enthält der Graph jedoch viele überflüssige Knoten.

Höcker (2010) beschreibt in seiner Arbeit einen ausgedünnten Sichtbarkeitsgraphen, der aus der Geometrie eines Szenarios erstellt wird. Die Knoten werden ebenfalls in einem bestimmten Abstand zu konvexen Ecken platziert. Hier wird zusätzlich die Anzahl der Kanten reduziert, in dem Kanten, die einen geringeren Winkel als den halbierten Innenwinkel einer konvexen Ecke eines beliebigen Polygons aufspannen, eliminiert werden.

In dem hier beschriebenen Ansatz liegt der Augenmerk darauf, dass die Graphknoten nicht den Eckpunkten entsprechen, da sonst die Fußgänger direkt an die Ecken laufen würden, was nicht dem realen Laufverhalten entspricht (Brogan & Johnson, 2003). Daher ist der *Plane Sweep* Algorithmus nicht geeignet.

Mit den Algorithmen von Gloor *et al.* (2004) und Höcker (2010) werden die Knoten zwar in einem gewissen Abstand zu den Ecken gesetzt und in Höcker (2010) wird der Graph zudem ausgedünnt, was zu einer Reduzierung der Kantenanzahl führt. Es existieren jedoch immer noch geometrisch nahe gelegene Kanten, die sehr ähnliche Wege beschreiben.

Gewünscht ist ein Graph, der eine möglichst geringe Anzahl an Knoten und Kanten besitzt, ohne Wege zwischen Quellen und Zielen zu zerstören. Ein ausgedünnter Graph ist aus dem Grund erstrebenswert, da die in Kapitel 6 vorgestellten Algorithmen diesen Graph durchsuchen, um Wege zwischen ausgewählten Knoten zu finden. Die Anzahl der Kanten sollte daher so gering wie möglich sein, um eine effiziente Ausführung zu gewährleisten.

Es sollen alle geometrisch nahe gelegenen Kanten eliminiert werden, so dass bei minimaler Kantenanzahl die räumliche Gebietsabdeckung bestehen bleibt.

Ein zweiter Aspekt, auf den ein Augenmerk gelegt wird, ist die Erreichbarkeit von Kanten innerhalb des Graphen. Damit ist gemeint, dass nur Kanten benötigt werden, die auf dem Weg zu den Zielen des Szenarios erreichbar sind. Alle anderen Kanten können vernachlässigt werden. Auch dies wird bei keinem der bisher vorgestellten Algorithmen betrachtet.

Im Folgenden wird ein Ansatz vorgestellt, der die beiden Aspekte enthält und daher den bisherigen Ansätzen überlegen ist. Er ist eine Weiterentwicklung des in Kneidl *et al.* (2010, 2011, 2012) vorgestellten Ansatzes. Dazu wird in Abschnitt 4.2 ein kegelbasierter Ansatz zur Grapherzeugung verwendet, der um eingefügte Kanten kegelförmige Gebiete aus dem Suchgebiet ausschneidet. Dies verhindert das

Einfügen von geometrisch nahegelegenen Kanten. Durch Überprüfung auf Zusammenhangskomponenten können nicht erreichbare Kanten aus dem Graphen gelöscht werden.

## 4.2 Kegelbasierter Ansatz zur Generierung von Navigationsgraphen

Die Generierung des Graphen besteht aus zwei Schritten; in Unterabschnitt 4.2.1 wird die Erzeugung der Knotenmenge beschrieben und Unterabschnitt 4.2.2 wird der Algorithmus zum Einfügen der Kanten vorgestellt.

Zur besseren Veranschaulichung wird ein Beispielszenario betrachtet, anhand dessen die komplette Grapherzeugung erläutert wird. Das Szenario besteht aus zwei Quellen und zwei Zielen, sowie einem Hindernispolygon und einer Wand. Der Abstand zwischen Wand und Hindernispolygon ist so gewählt, das eine Person die Engstelle passieren kann. In Abbildung 4.1 ist das Szenario dargestellt.

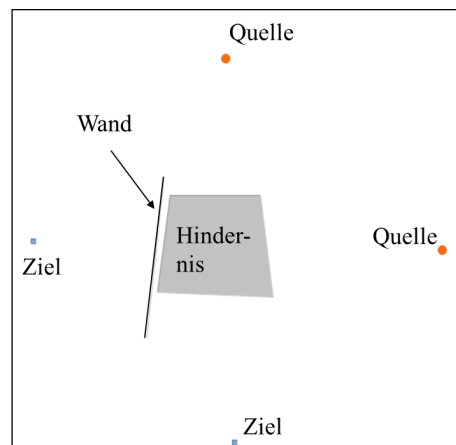


Abbildung 4.1: Beispielszenario bestehend aus zwei Quellen und zwei Zielen. Von jeder Quelle aus sind jeweils beide Ziele erreichbar. Die Sichtverbindung zwischen den gegenüberliegenden Quellen und Zielen wird durch zwei Hindernisse blockiert. Der Abstand zwischen Wand und Hindernis ist so gewählt, dass ein Fußgänger die Engstelle passieren kann.

### 4.2.1 Erzeugung der Knotenmenge $V$

Die Knoten des Sichtbarkeitsgraphen werden in mehreren Schritten erzeugt. Zunächst wird an jeder konvexen Ecke eines Hindernisses ein Knoten platziert. Dieser wird in Verlängerung der Winkelhalbierenden der Ecke in einem einstellbaren Abstand gesetzt (siehe Abbildung 4.2). Dabei werden mehrere Überprüfungen durchgeführt: Zunächst wird die Sichtbarkeit zwischen Knoten und Hindernisecke geprüft. Ist der Knoten von der Ecke aus nicht sichtbar, so gibt es zwei Möglichkeiten:

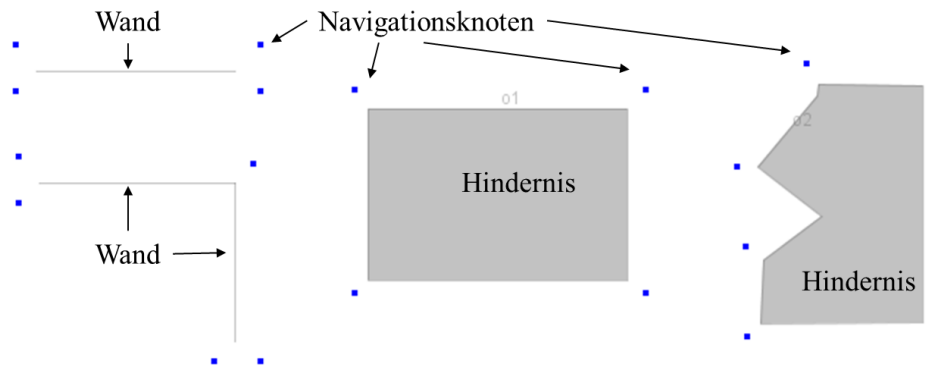


Abbildung 4.2: Platzierung der Navigationsknoten am Beispiel verschiedener Hindernisse

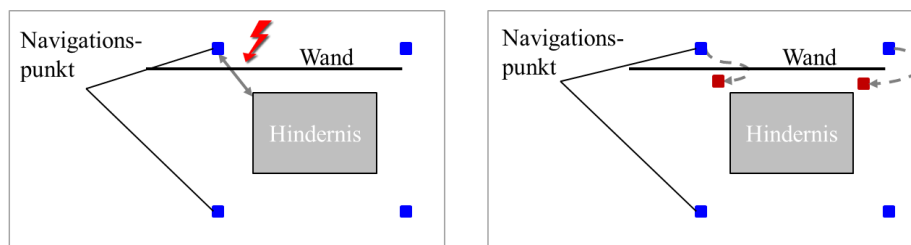


Abbildung 4.3: Umsetzen der Navigationsknoten, so dass eine Sichtverbindung zwischen Ecke und Knoten besteht

- Falls der Abstand zwischen schneidendem Hindernis und Hindernisecke größer als der Inkreisradius einer Zelle ist, wird der Knoten genau auf die Hälfte des Abstands zwischen Hindernisecke und schneidendem Hindernis gesetzt.
- Falls der Abstand kleiner als der Inkreisradius einer Zelle ist, wird kein Knoten platziert, da kein Fußgänger diese Engstelle passieren kann.

Ein Beispiel für das Umsetzen der Navigationsknoten ist in Abbildung 4.3 illustriert.

Die so erhaltene Knotenmenge kann geometrisch nahegelegene Knoten beinhalten, die im nächsten Schritt zusammengefasst werden, da die Anzahl der Knoten so gering wie möglich sein sollte im Hinblick auf die auf dem Graph definierten Wegfindungsalgorithmen. Als geometrisch nahe gelegene Knoten werden diejenigen Knoten bezeichnet, die einen kleineren Abstand als den zu den Hindernissen eingestellte Abstand haben. Formal bedeutet das:  $dist(u, v) < dist_{obs}$ , wobei  $u, v$  Navigationsknoten darstellen und  $dist_{obs}$  den voreingestellten Abstand zu Hindernissen definiert. Um diese Knoten in einem neuen Knoten zusammenfassen zu können, muss die Bedingung erfüllt sein, dass alle Knoten, die bisher in Sichtverbindung zu jedem dieser Knoten lagen, auch nach dem Zusammenfassen sichtbar sind. Dies kann erreicht werden, indem zu jedem Knoten die entsprechende Hindernisecke sowie das zugehörige Hindernis mit abgespeichert werden. Sofern die

Knoten in Sichtverbindung zu dem platzierten Navigationsknoten liegen, muss auch der neu zusammengefasste Knoten eine Sichtverbindung zu diesen Ecken haben. Falls dies gegeben ist, kann ein neuer Knoten generiert werden. In Abbildung 4.4 ist dieser Schritt verdeutlicht. Hier werden zunächst die Knoten  $u$  und  $v$  überprüft. Jeder der beiden Knoten hat eine Sichtverbindung zu zwei weiteren Knoten. Nach der Erzeugung des neuen Knoten  $w$  bleibt die Sichtverbindung erhalten (siehe Abbildung 4.4a). Im zweiten Fall, bei der Überprüfung, ob der Knoten  $w$  mit dem Knoten  $x$  zusammengefasst werden kann, bleiben die Sichtverbindungen nicht erhalten. In diesem Fall darf der Knoten nicht mit seinen Nachbarn verschmolzen werden. Selbiges gilt für den Knoten  $w$  und  $y$  (siehe Abbildung 4.4b).

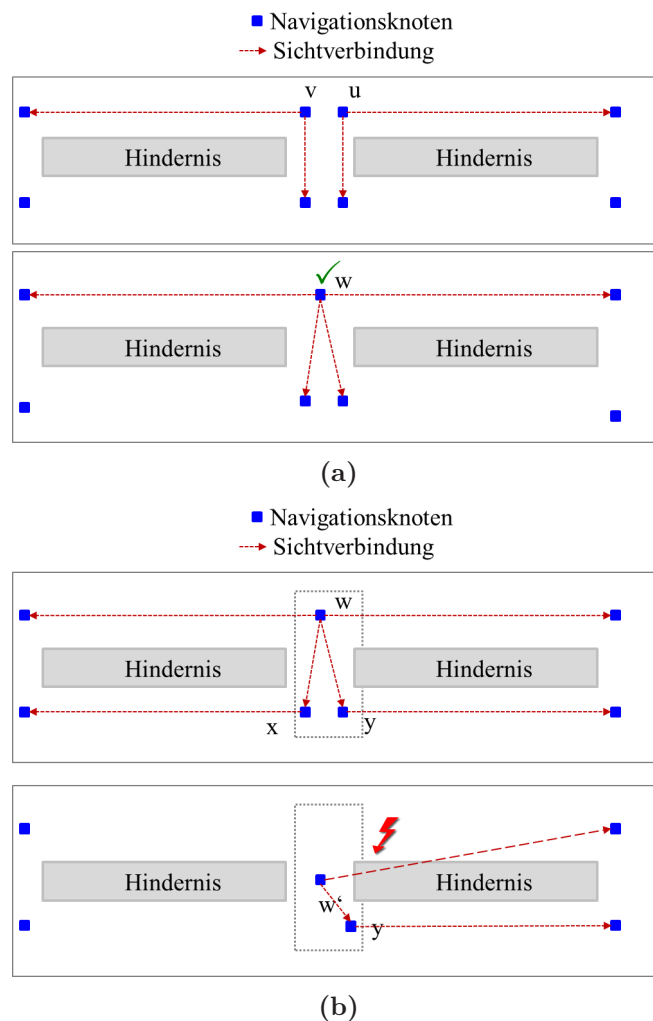


Abbildung 4.4: Überprüfen auf das Zusammenfassen von Navigationsknoten: a) Falls eine Sichtverbindung zu den Knoten der benachbarten Ecken nach dem Zusammenfassen bestehen bleibt, so kann ein neuer Knoten gesetzt werden. b) Falls nach der Verschmelzung die Sichtverbindung verloren geht, müssen die ursprünglichen Knoten erhalten bleiben.

Angewandt auf das Demonstrationsbeispiel aus Abbildung 4.1 werden die Knoten, wie in Abbildung 4.5 dargestellt, platziert.

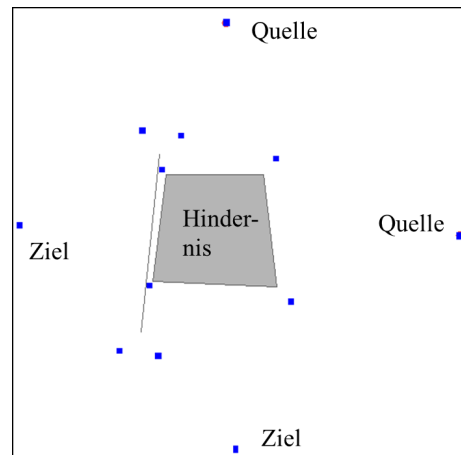


Abbildung 4.5: Beispielszenario mit den erzeugten Navigationsknoten.

## 4.2.2 Erzeugung der Kantenmenge $E$

Ziel ist es, einen Graphen zu erzeugen, mit dem menschliches Navigationsverhalten detailliert abgebildet werden kann. Im einfachsten Fall, wenn alle Knoten in Sichtverbindung miteinander verbunden werden, entsteht ein sehr dichter Graph, der viele redundante Kanten beinhaltet. Zu redundanten Kanten zählen Kanten, die geometrisch sehr nahe liegen (siehe Abbildung 4.6) sowie Kanten, die auf dem Weg zu den jeweiligen Zielen nicht erreichbar sind, also alle Kanten einer Zusammenhangskomponente, in der kein von der Quelle erreichbares Ziel enthalten ist. Diese Redundanz bedingt eine Erhöhung der Laufzeit der darauf angewendeten Algorithmen. Damit auch bei einer großen Anzahl von simulierten Personen die Laufzeit nicht erheblich erhöht wird, sollte der Graph auf der einen Seite die menschlichen Navigationsaspekte gut abdecken, auf der anderen Seite aber so ausgedünnt wie möglich sein. Dies kann erreicht werden, indem redundante Kanten vermieden werden.

Die Idee ist, Knoten unter Berücksichtigung der Kantenwinkel zu verbinden. Begonnen wird mit einem beliebigen Knoten, von dem aus die nächsten Nachbarn gesucht werden. Zwischen dem nächsten Nachbarn und dem betrachteten Knoten wird eine Kante eingefügt. Um diese Kante wird ein kegelförmiges Gebiet aus dem Suchgebiet ausgeschnitten nicht weiter betrachtet. Damit wird der Suchbereich eingeschränkt; Knoten, die in gleicher Richtung liegen aber weiter entfernt vom betrachteten Knoten, werden nicht verbunden.

### 4.2.2.1 Räumliche Datenstrukturen

Um die Identifizierung von nächstgelegenen Nachbarn effizient durchzuführen, wird Gebrauch von einer räumlichen Datenstruktur gemacht. Die Geometrie des

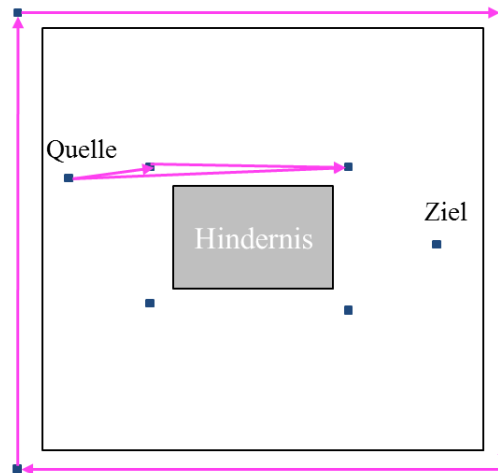


Abbildung 4.6: Beispiel für redundante Kanten: Dazu zählen geometrisch nahe gelegene Kanten, sowie Kanten, die in einer Zusammenhangskomponente ohne Verbindung zu Quelle oder Ziel liegen.

Szenarios sowie die erstellten Navigationsknoten werden dazu in dieser räumlichen Datenstruktur gespeichert. Es handelt sich bei der Geometrie eines Szenarios um zweidimensionale polygonale Objekte unterschiedlicher Größe. Räumliche Datenstrukturen speichern geometrische Objekte in effizienten Strukturen (meist in Bäumen), um die Zugriffe auf die Elemente zu beschleunigen. Zudem unterstützen räumliche Datenstrukturen räumliche Abfragen, die innerhalb des Algorithmus verwendet werden.

Ein Baum ist formal definiert als:

**Definition 4.2** Ein Graph  $G(E, V)$ , mit Kantenmenge  $E$  und Knotenmenge  $V$ , ist genau dann ein Baum, wenn der Graph zusammenhängend und zyklensfrei ist. Bei gerichteten Bäumen wird der Knoten ohne eingehende Kanten als Wurzel bezeichnet. Alle Knoten, die keine ausgehenden Kanten besitzen, werden als Blätter bezeichnet.

Der Algorithmus zur Erstellung der Kantenmenge  $E$  verwendet zum Auffinden von Knoten vor allem Nachbarschaftssuchen innerhalb bestimmter räumlicher Bereiche. Eine Nächste-Nachbarn-Suche ist wie folgt definiert:

**Definition 4.3** Eine Nächste-Nachbarn-Suche (engl. Nearest Neighbor Search) findet zu einem Objekt  $o$  alle Objekte  $o_i$  innerhalb eines definierten Radius oder eines definierten Bereichs, sortiert nach deren Abstand zu  $o$ .

**Quadtree** – Eine bekannte Vertreterin räumlicher Datenstrukturen ist die Familie der Quadtrees in 2D (bzw. Octtree in 3D). Diese Art von Datenstruktur teilt

den Raum rekursiv in gleichgroße Rechtecke auf. Sobald ein Objekt komplett innerhalb eines Quadranten (Rechtecks) liegt, muss nicht weiter verfeinert werden. Der Vorteil des klassischen Quadtree ist die gleichmäßige Unterteilung eines betrachteten Gebiets und die Effizienz bei Anfragen, bei denen überprüft wird, ob Punkte innerhalb oder außerhalb eines Objektes liegen. Für Nachbarschaftsabfragen sind diese Strukturen ungeeignet. Da der Baum nicht höhenbalanciert ist, können die Blätter des Baumes auf unterschiedlichen Ebenen liegen. Dies kann zur Folge haben, dass die Suche innerhalb von Bereichen im schlechtesten Fall linear ist, da ein Zweig des Baums eine sehr hohe Tiefe besitzen kann und somit viele Knoten betrachtet werden müssen.

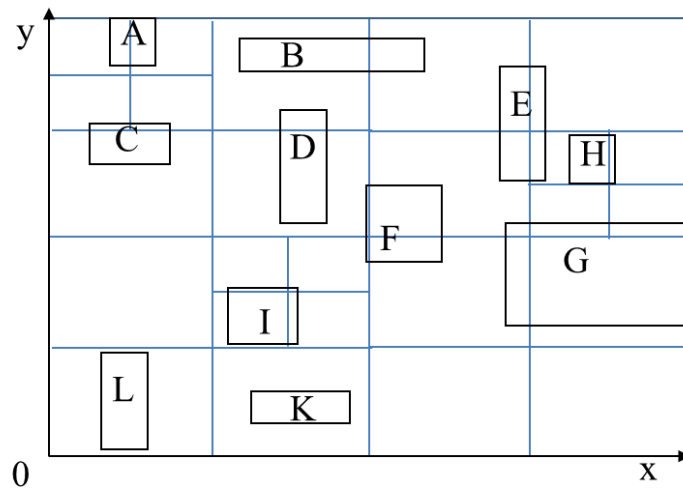
Eine Variante des Quadtree für polygonale Daten wird als MX-CIF Quadtree bezeichnet (Samet, 1990, S.200 ff). Dabei wird der Raum nicht so weit zerlegt, bis ein Objekt komplett innerhalb eines Quadranten liegt, sondern die Objekte werden jeweils in dem Knoten abgespeichert, in dem sie eine der beiden Linien schneiden, die durch den Mittelpunkt eines Quadranten führen. Ein Beispiel hierfür ist in Abbildung 4.7 gegeben. Hier werden die Datenpolygone  $B, F, G$  im Wurzelknoten 1 abgespeichert, da sie entweder die Linie durch den Mittelpunkt in  $y$ -Richtung ( $F, B$ ) und/oder in  $x$ -Richtung schneiden ( $F, G$ ). Der Vorteil gegenüber klassischen Quadtree ist, dass der Baum für Bereichsabfragen oder Nachbarschaftsabfragen nicht so tief durchsucht werden muss. Der Nachteil liegt darin, dass eventuell eine große Anzahl von Rechtecken durchsucht werden muss, da je nach Lage der Rechtecke eine große Anzahl innerhalb eines Quadranten (Knoten) liegen können. Abhilfe schafft der sog. *expanded* MX-CIF Quadtree (Samet, 1990, S.215 ff), bei dem die Rechtecke in mehreren Knoten gespeichert sein können. Dies hat wiederum den Nachteil, dass die Daten redundant gespeichert werden und bei einer großen Anzahl von Objekten dies zu sehr hohem Speicherplatzbedarf führen kann. Da eine Struktur zur effizienten Unterstützung von räumlichen Bereichsanfragen und Nachbarschaftsabfragen bei einer gleichzeitig großen Anzahl an Rechtecken bzw. polygonalen Objekten benötigt wird, scheidet diese Datenstruktur aus.

**$k$ - $d$ -Baum** – Eine Struktur, die räumliche Anfragen effizient unterstützt, ist der  $k$ - $d$ -Baum, wobei  $k$  die Dimension bezeichnet. Das Grundprinzip des Baums ist, dass das zu speichernde Gebiet entlang eines zu bestimmenden Medians alternierend in jeder Dimension geteilt wird. Der Nachteil dieser Struktur ist es, dass sie für Daten, die aus Punkten und Polygonen (im 2D-Fall) bestehen, ungeeignet ist. Das liegt daran, dass Mediane Polygone schneiden können und sich damit Polygone in zwei unterschiedlichen Knoten befinden können.

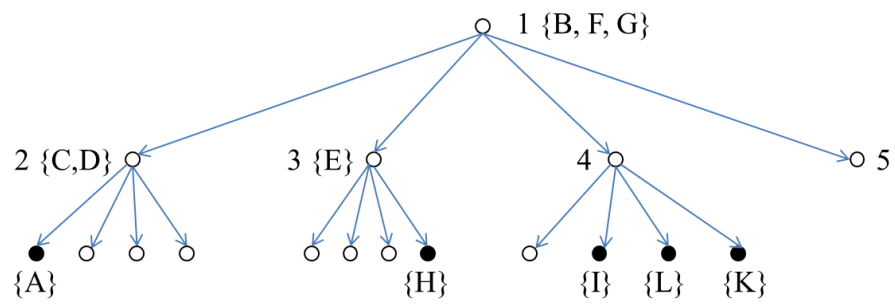
**$R^*$ -Baum** – Da die zu speichernden Daten genau aus Punkten und Polygonen bestehen, wird hier eine alternative Datenstruktur verwendet, der  $R^*$ -Baum. Dieser Baum wird häufig zur Speicherung von geographischen Daten verwendet und unterstützt sehr effizient räumliche Bereichsabfragen (engl. *Range Queries*), innerhalb derer Nächste-Nachbarn-Suchen durchgeführt werden.

Der  $R^*$ -Baum Index, entwickelt von Beckmann *et al.* (1990), ist eine Weiterentwicklung des  $R$ -Baums („Region-Tree“) (Guttman, 1984).

Eine weitere Variante formt der  $R^+$ -Baum. Der Unterschied dieser Strukturen



(a) Datenset



(b) Hierarchie des Quadtree

Abbildung 4.7: Datenset, bestehend aus 11 Rechtecken und zugehöriger MX-CIF Quadtree



liegt in der Verwendung der Regionen, in denen die Daten verwaltet werden. Während der R-Baum keine expliziten Regeln über die Anordnung der Regionen vorgibt, dürfen sich im  $R^+$ -Baum die Regionen nicht überlappen. Im  $R^*$ -Baum wird die Überlappung der Regionen minimiert.

Alle drei Varianten bestehen aus zwei unterschiedlichen Arten von Knoten, Blattknoten und innere Knoten, analog zu B-Bäumen (Bayer & McCreight, 1972), die zur Verwaltung von Indexstrukturen bei Datenbanken eingesetzt werden. Die inneren Knoten speichern die Regionen oder anders genannt, minimal umgebende Rechtecke (*engl. Minimum Bounding Rectangles, MBR*) und einen Verweis auf die Knoten der nächsten Ebene. Die MBRs werden aus den Minima und Maxima der x- und y-Werte (in 2D) der in den Kindknoten enthaltenen Elemente definiert. Blattknoten speichern die eigentlichen Daten. Im Beispiel aus Abbildung 4.8a ist das MBR  $B$  also durch die untere linke Ecke mit  $(p1_x, p3_y)$  und die obere Ecke mit  $(p4_x, p2_y)$  definiert.

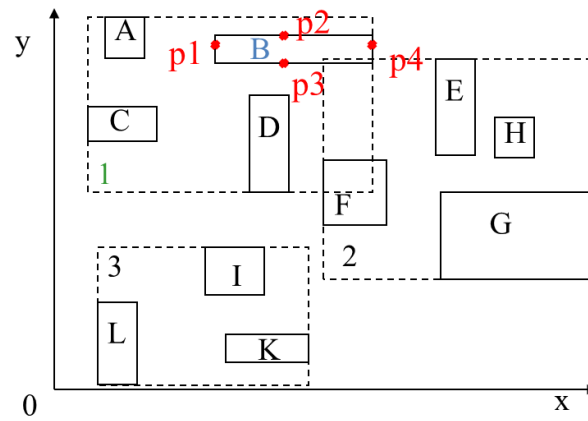
Alle Blattknoten liegen auf der selben Ebene, es handelt sich somit um einen balancierten Baum.

Sei  $m$ ,  $2 < m \leq M/2$  die minimale Anzahl von Einträgen in einem Knoten und  $M$  die maximale Anzahl von Elementen pro Knoten. Für jeden inneren Knoten muss folgende Bedingung jederzeit gelten:  $m \leq \text{Anzahl Kindknoten} \leq M$ . Für Blattknoten gilt:  $m \leq \text{Anzahl Einträge} \leq M$ . Die Wurzel hat mindestens 2 Einträge, außer die Wurzel ist ein Blattknoten.

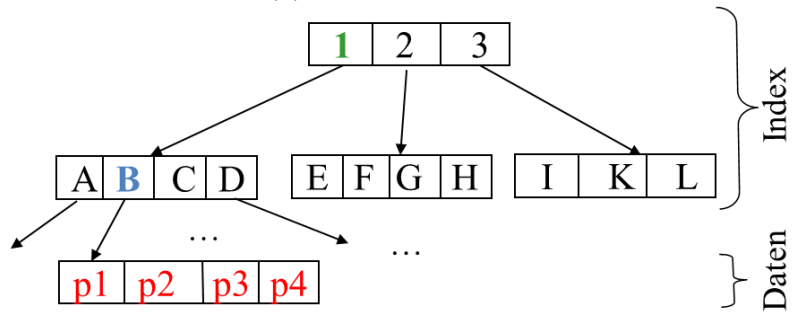
Der Baum wird aufgebaut, in dem nach und nach die zu speichernden Elemente hinzugefügt werden. Zunächst wird die Wurzel erstellt, in der die Elemente direkt gespeichert werden. Wird obige Bedingung verletzt, wird die Wurzel geteilt, und eine neue Ebene wird dem Baum hinzugefügt. Dieser so genannte Split kann nach unterschiedlichen Strategien erfolgen:

- O1: Die Fläche, die von den minimal umgebenden Rechtecken (MBR) aufgespannt wird, soll minimal sein.
- O2: Die Überschneidung der MBRs soll minimal sein
- O3: Der Margin der inneren Knoten soll minimiert werden. Der Margin beschreibt hier die Summe der Längen  $l$  der Kanten eines Rechtecks  $R(b, l)$ . Das soll bewirken, dass die Rechtecke möglichst quadratische Formen annehmen.
- O4: Der Speicherplatz soll optimiert werden

Die Strategien und Auswirkungen auf die Performance von Anfragen werden in Beckmann *et al.* (1990) diskutiert. In Abbildung 4.8 sind beispielhaft dieselben räumlichen Daten wie in Abbildung 4.7a sowie der zugehörige  $R^*$ -Baum dargestellt. Der Baum besitzt die Höhe 3 und exemplarisch für alle Datenpunkte werden  $p1$  bis  $p4$  dargestellt.



(a) Datenset mit MBRs

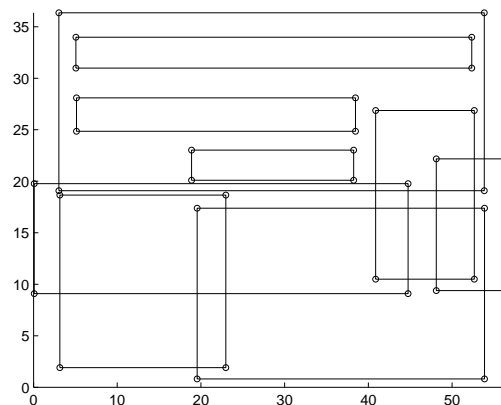


(b) resultierender R\* Baum

Abbildung 4.8: Beispiel für einen R\*-Baum. In a) sind exemplarisch  $p1$  bis  $p4$  als zu speichernde Daten sowie die resultierenden MBRs dargestellt, in b) die Hierarchie des R\*-Baums.



(a) Szenario

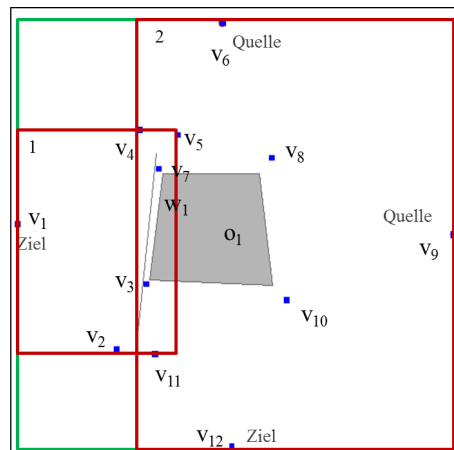


(b) MBRs des resultierenden R\*-Baums mit Tiefe 2

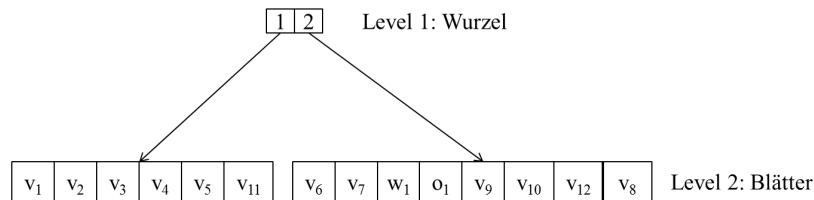
Abbildung 4.9: Ein Bürogebäude als Beispielszenario in a) und zugehöriger R\*-Baum der Tiefe 2 in b)

Eine komplexere Geometrie besitzt das Beispiel in Abbildung 4.9a. Das Szenario besteht aus Büros, in denen jeweils eine Quelle platziert ist. Es befinden sich drei Ausgänge in dem Stockwerk, zwei Treppen an der östlichen und westlichen Seite sowie ein Aufzug in der Mitte des Gebäudes. Der resultierende R\*-Baum der Tiefe 2 ist in Abbildung 4.9b angegeben.

Der R\*-Baum, angewandt auf das oben eingeführte Demonstrationsbeispiel (Abbildung 4.1), ist in Abbildung 4.10 dargestellt. Unter Verwendung dieser Datenstruktur können effizient Bereichsabfragen durchgeführt werden und nach nächsten Nachbarn innerhalb solcher Bereiche gesucht werden. Dies ist für den im nächsten Abschnitt vorgestellten Algorithmus zur Erstellung der Kantenmenge von Bedeutung.



(a) Szenario: Rot eingerahmt ist die Blattebene, Grün eingerahmt die Wurzel



(b) R\* Baum des Szenarios

Abbildung 4.10: R\*-Baum für das Demonstrationsbeispiel aus Abbildung 4.1

#### 4.2.2.2 Kegelbasierter Algorithmus zur Konstruktion der Kantenmenge $E$

Zur Vermeidung von redundanten Kanten wird ein kegelbasierter Algorithmus angewandt. Die Grundidee des Algorithmus ist es, durch das Ausschneiden von kegelförmigen Bereichen aus dem Suchgebiet geometrisch nahe gelegenen Kanten zu vermeiden. Im Detail funktioniert der Algorithmus wie folgt:

Zunächst werden alle Hindernisse, Quellen und Ziele sowie die erstellten Navigationsknoten zum Index hinzugefügt. Diese Elemente können damit effizient durchsucht werden. Der Algorithmus startet mit einem beliebigen Knoten  $v_i$ . Für diesen Knoten  $v_i$  wird ein rechteckiger Suchbereich definiert, in dem die gesamte Geometrie des Szenarios enthalten ist (vgl. Abbildung 4.12a). Innerhalb des Bereichs wird eine Nächste-Nachbarn-Suche für alle Knoten in Sichtverbindung zu  $v_i$  durchgeführt. Die Knoten werden in aufsteigender Entfernung zu Knoten  $v_i$  sortiert zurückgeliefert (vgl. Abbildung 4.12b). Zwischen  $v_i$  und dem nächstgelegenen Knoten  $v_j$  wird eine Kante  $e_{ij}$  eingefügt. Dann wird aus dem Suchbereich ein Kegel ausgeschnitten, der von  $v_i$  mit einem vorgegebenen Winkel  $\alpha$  um die gefundene Kante  $e_{ij}$  erzeugt wird (siehe Abbildung 4.11). Alle Knoten, die innerhalb dieses Bereichs liegen, werden nicht weiter betrachtet, so dass innerhalb dieses Kegels keine weitere Kante mehr eingefügt wird. Jetzt wird der nächstgelegene Knoten außerhalb des Kegels betrachtet und durch eine weitere Kante mit

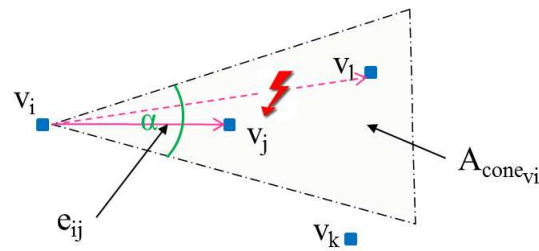


Abbildung 4.11: Erzeugung des Kegels, der aus dem Suchbereich ausgeschnitten wird. Knoten  $v_j$  ist der nächste Nachbar von  $v_i$ . Da  $v_l$  innerhalb des Kegels liegt, wird keine Kante zwischen  $v_i$  und  $v_l$  eingefügt.

Knoten  $v_i$  verbunden. Dann wird ausgehend von  $v_i$  wieder ein Kegel konstruiert, um so weitere Knoten aus dem Suchgebiet auszuschließen (vgl. Abbildung 4.12c). Die ausgeschnittenen Bereiche können sich dabei überlappen.

Dies wird so lange wiederholt, bis keine Knoten mehr existieren, die außerhalb der ausgeschnittenen Kegel liegen. Damit ist der Knoten  $v_i$  komplett untersucht und alle Kanten eingefügt (vgl. Abbildung 4.12d). Der Algorithmus wird für alle Knoten des Graphen wiederholt. Der Pseudocode des Algorithmus ist in Algorithmus 1 aufgelistet.

Der Winkel  $\alpha$  bestimmt die Anzahl der resultierenden Graphkanten. Je größer der Winkel, desto geringer die Kantenanzahl, da größere Bereiche aus dem Suchgebiet herausgeschnitten werden. Ziel ist es also, ein zulässiges  $\alpha_{zul}$  zu finden, so dass möglichst viele redundante Kanten eliminiert werden, gleichzeitig aber keine Wege zwischen Quelle und Ziel entfernt werden.

Die Bestimmung von  $\alpha$  geschieht durch eine binäre Suche. Dazu wird zunächst eine Obergrenze  $\alpha_{max}$  und eine Untergrenze  $\alpha_{min}$  bestimmt.  $\alpha_{min}$  wird als minimaler Winkel zwischen allen ausgehenden Kanten im vollständigen Sichtbarkeitsgraphen gewählt,  $\alpha_{max}$  als maximaler Winkel zwischen zwei ausgehenden Kanten im Sichtbarkeitsgraphen.  $\alpha$  wird der Wert  $(\alpha_{max} - \alpha_{min})/2$  zugewiesen. Bleibt Ziel  $t$  von Start  $s$  aus für dieses  $\alpha$  erreichbar, wird  $\alpha$  um die Hälfte erhöht. Existiert kein Weg mehr für  $\alpha$ , so wird  $\alpha$  halbiert. Dieser Schritt wird so lange wiederholt, bis eine Abbruchbedingung erfüllt ist. Diese Abbruchbedingung kann unterschiedlich definiert sein. Es kann ein bestimmtes  $\epsilon$  gewählt werden, so dass  $\alpha_{max} - \alpha_{min} < \epsilon$  erfüllt ist. Die Abbruchbedingung kann sich auch nach der prozentualen Verlängerung des Wegs von  $s$  nach  $t$  richten. In Algorithmus 2 ist der Pseudocode zur Bestimmung eines maximalen  $\alpha$  dargestellt.

In Abbildung 4.13 sind die resultierenden Graphen des Demonstrationsbeispiels für unterschiedliche Werte von  $\alpha$  zu sehen. In Abbildung 4.13d ist der Graph mit dem maximalen  $\alpha_{zul}$  zu sehen. Hier ist zu erkennen, dass ein zu großes  $\alpha$  die Wege erheblich verlängert. Dem Benutzer sollte deshalb die Möglichkeit gegeben werden, eine Obergrenze für  $\alpha$  zu festzulegen.



**Algorithmus 1** Graphgenerierung

---

```

1: Input: Knoten  $V$ ; Alle Ziele des Szenarios  $D \subseteq V$ ; Hindernisse  $O$ ; Winkel  $\alpha$ 
2: Ergebnis: Graph( $V,E$ )
3: Füge alle Knoten als  $V$  zum Graph  $G$  hinzu
4: for Jeden Knoten  $v_i$  in  $V$  do
5:   Definiere Suchbereich  $A_{search}$ , so dass gilt:  $v_i \in A_{search}$  und  $O \in A_{search}$ 
6:   Suche innerhalb  $A_{search}$  nach allen Knoten in Sichtverbindung
7:   und sortiere sie anhand ihres Abstands zu  $v_i$ 
8:   for jeden Nachbarn  $n$  do
9:     if  $n$  innerhalb  $A_{search}$  then
10:      füge Kante  $e(v_i, n)$  in  $G$  ein
11:      Schneide kegelförmigen Bereich  $A_{cone_{v_i}}$  mit Winkel  $\alpha$  um  $e$  aus:
12:       $A_{search} = A_{search} \setminus A_{cone_{v_i}}$ 
13:     end if
14:   end for
15: end for

```

---

**Algorithmus 2** Bestimmung des maximalen  $\alpha$ 


---

```

1: Gegeben:  $\alpha_{min}, \alpha_{max}$ 
2:  $\alpha = (\alpha_{max} - \alpha)/2$ 
3: Generiere Graph  $G$  mit  $\alpha$ 
4: while Abbruchbedingung nicht erfüllt do
5:   if  $t$  von  $s$  aus in  $G$  erreichbar then
6:      $temp = \alpha$ 
7:      $\alpha = (\alpha_{max} - \alpha)/2 + \alpha$ 
8:      $\alpha_{min} = temp$ 
9:   else
10:     $temp = \alpha$ 
11:     $\alpha = \alpha - (\alpha_{max} - \alpha)/2$ 
12:     $\alpha_{max} = temp$ 
13:   end if
14:   Generiere Graph  $G$  mit  $\alpha$ 
15: end while

```

---

Bisher wurde davon ausgegangen, dass das Szenario aus nur einem Startknoten und einem Zielknoten besteht. Dies ist nicht immer der Fall. Die Erweiterung auf mehrere Quell- und Zielknoten kann direkt umgesetzt werden. Statt ein Quelle-Ziel Paar zu betrachten, müssen nun alle Quelle-Ziel-Kombinationen untersucht werden. Falls ein Ziel von einem Quellknoten nicht mehr erreichbar ist, muss  $\alpha$  entsprechend angepasst werden.

**Entfernen von nicht-erreichbaren Knoten**

Der so erzeugte Graph kann Knoten enthalten, die mit keiner Quelle und keinem Ziel verbunden sind (siehe Abbildung 4.6). Ziel ist es, diese Knoten aus dem Graphen zu entfernen. Dies kann umgesetzt werden, indem nach Zusam-

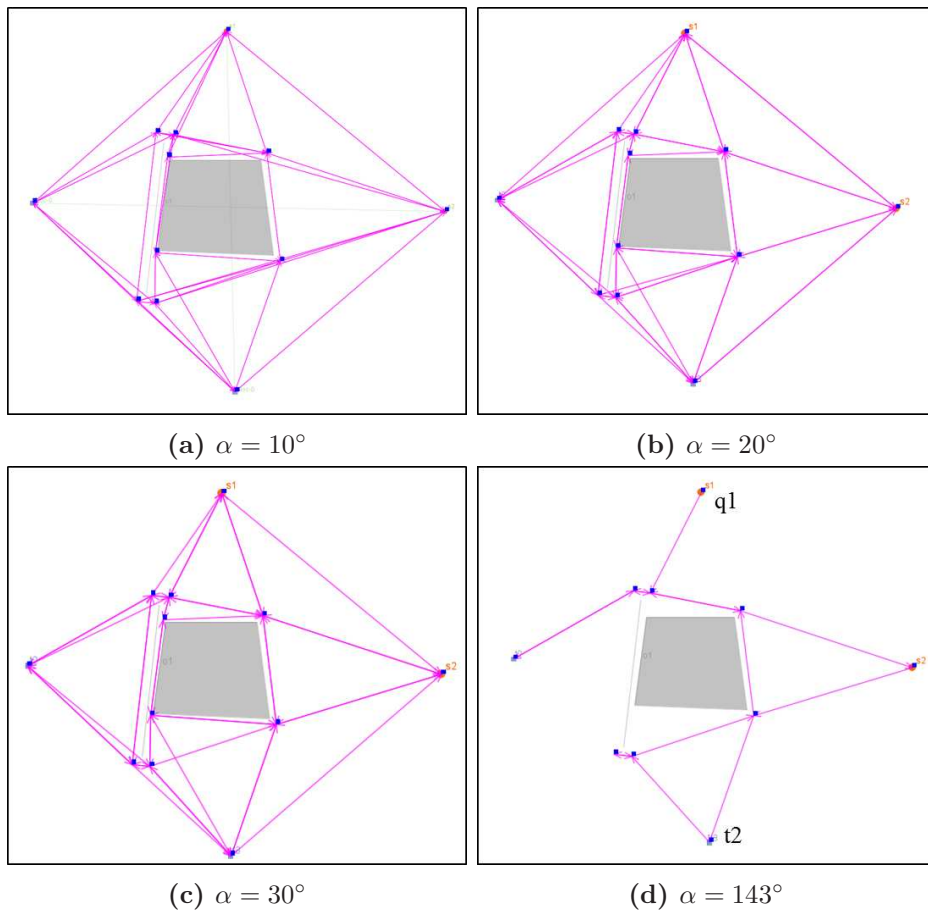


Abbildung 4.13: Graphen mit unterschiedlichen Werten für  $\alpha$ . In d) entspricht  $\alpha = \alpha_{zul}$ , so dass ein Weg zwischen allen Quellen und Zielen existiert.



menhangskomponenten im Graphen gesucht wird (Gibbons, 1999). Um eine Zusammenhangskomponente zu finden, wird eine Breitensuche von jeder Quelle aus ausgeführt. Ist in dem entstandenen Suchbaum (Zusammenhangskomponente) mindestens ein Ziel vorhanden, so werden alle Knoten dieser Komponente behalten. Alle Knoten, die in keiner so erstellten Zusammenhangskomponente enthalten sind, werden aus dem Graph entfernt.

### 4.3 Qualitativer Vergleich des ausgedünnten Sichtbarkeitsgraphen mit einem vollständigen Sichtbarkeitsgraphen

Um den Einfluss von  $\alpha$  auf den resultierenden Graph zu untersuchen, werden Szenarien unterschiedlicher Größe und Geometrie für verschiedene Werte von  $\alpha$  herangezogen. Zur besseren Vergleichbarkeit werden zwei Metriken eingeführt, die eine qualitative Aussage bezüglich der Güte der Graphen ermöglichen. Die erste Metrik stützt sich auf die **Dilatation** eines Graphen. Die (geometrische) Dilatation eines Graphen ist ein Maß dafür, wie viel Umweg beim Traversieren der Kanten im Graphen von einem Knoten  $u \in V$  zu einem Knoten  $v \in V$  in Kauf genommen werden muss im Vergleich zur direkten euklidischen Strecke. Sie definiert das Verhältnis der Entfernung im Graphen zur euklidischen Strecke (Dickmeiß, 2005).

Die Dilatation kann hier zur Angabe des Verhältnisses des kürzesten Wegs in dem ausgedünnten Graphen im Vergleich zum vollständigen Graphen verwendet werden. Hierbei gilt folgende Formel:

$$dil(G_a, G_v) = \max_{\forall (s,t) \in S \times T} \left( \frac{length(s_{G_a}, t_{G_a})}{length(s_{G_v}, t_{G_v})} \right)$$

Dabei bezeichnet  $G_a$  den ausgedünnten Graph und  $G_v$  den vollständigen Sichtbarkeitsgraph.  $S \subset V$  bezeichnet die Teilmenge aller Startknoten und  $T \subset V$  enthält alle Endknoten der Graphen  $G_a$  und  $G_v$ . Hier wird vorausgesetzt, dass für jedes Paar  $(s, t) \in S \times T$  ein Weg existiert.

**Kantenverhältnis** Zusätzlich zur Dilatation ist die Anzahl der Kanten von Bedeutung. Dabei wird in Relation gesetzt, wie viele Kanten jeweils im vollständigen sowie ausgedünnten Graphen vorhanden sind.  $\delta(edge_{G_a}, edge_{G_v})$  wird wie folgt definiert:

$$\delta_{edge}(G_a, G_v) = \frac{|E_{G_a}|}{|E_{G_v}|}$$

Im Folgenden werden vier unterschiedliche Szenarien mit unterschiedlicher Skalierung und Eigenschaften betrachtet, anhand derer jeweils der vollständige Sichtbarkeitsgraph sowie der ausgedünnte Graph miteinander verglichen werden.

**Szenario 1: obiges Beispielszenario** Das hier bereits vorgestellte Beispielszenario repräsentiert das kleinskaligste Beispiel für die  $\alpha$ -Studie. Es sind insgesamt vier Quelle-Ziel Relationen vorhanden, wobei der Weg von  $q1$  nach  $t2$  (siehe

Abbildung 4.13d) die größte Dilatation aufweist und damit dieser Weg als Vergleichswert für unterschiedliche Winkel herangezogen wird.

**Szenario 2: Bürogebäude** Dieses Szenario stellt ein Stockwerk in einem Bürogebäude dar und besteht ausschließlich aus Wänden. In jedem Büro ist eine Quelle platziert, in der eine unterschiedliche Anzahl von Personen generiert werden, abhängig von der Größe des Raums. Es gibt insgesamt drei Ziele, die beiden Treppen östlich und westlich des Stockwerks sowie ein Aufzug in der Mitte des Gebäudes. Der Grundriss beträgt 60 x 40 Meter. Der Weg mit der größten Dilatation führt von  $q1$  nach  $z2$ . Das Szenario ist in Abbildung 4.14 dargestellt, der generierte vollständige Sichtbarkeitsgraph ist in Abbildung 4.15, und der ausgedünnte Graph mit maximal zulässigem  $\alpha_{zul}$  in Abbildung 4.16 abgebildet.



Abbildung 4.14: Szenario 2: Bürogebäude

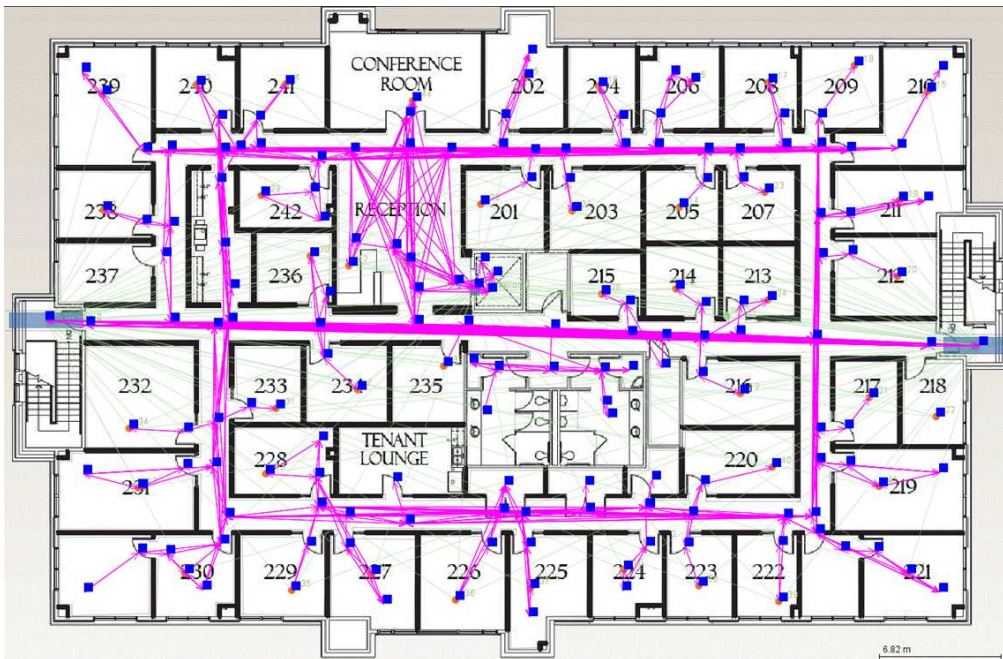


Abbildung 4.15:  $G_v$  für das Szenario Bürogebäude

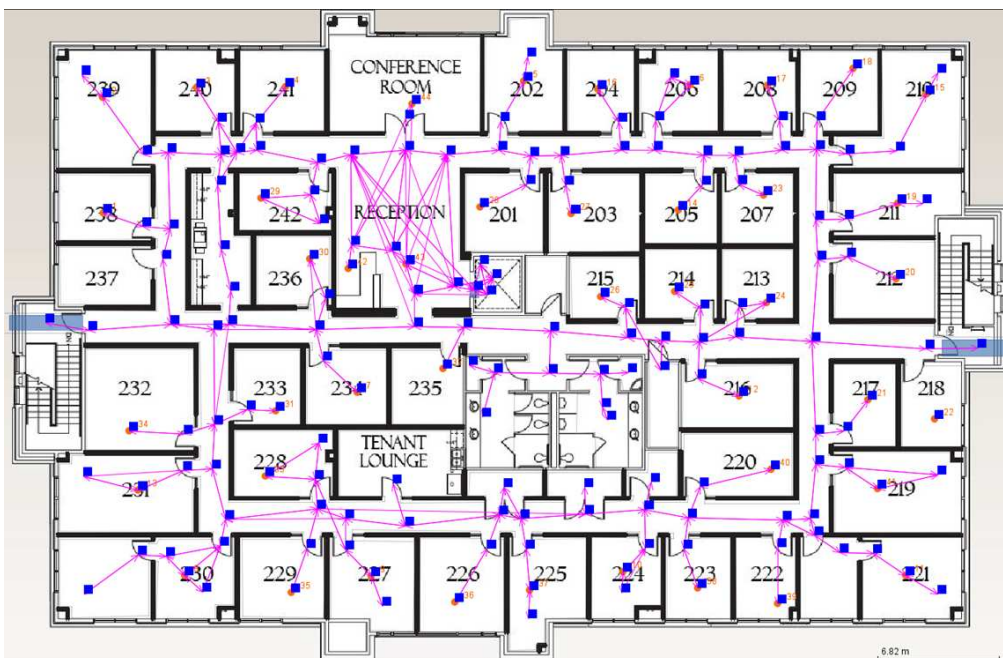


Abbildung 4.16:  $G_a$  mit  $\alpha_{zul} = 58.61^\circ$  für das Szenario Bürogebäude

**Szenario 3: Universitätsgebäude** Das dritte Szenario beschreibt ein Gebäude der TU München, das aus Hörsälen, einer großen Magistrale sowie vielen Büros besteht (Driesel, 2012). Modelliert wird das Szenario aus Wänden und Hindernissen, so dass die Kombination von unterschiedlichen geometrischen Objekten mit einfließt. Zudem ist dieses Szenario geometrisch sehr komplex. Insgesamt hat das Szenario einen Grundriss von 180 x 300 Metern. Das Szenario ist in Abbildung 4.17 dargestellt, der erstellte vollständige Graph ist in in Abbildung 4.18 und der ausgedünnte Graph mit maximal zulässigem  $\alpha_{zul}$  in Abbildung 4.19 abgebildet.

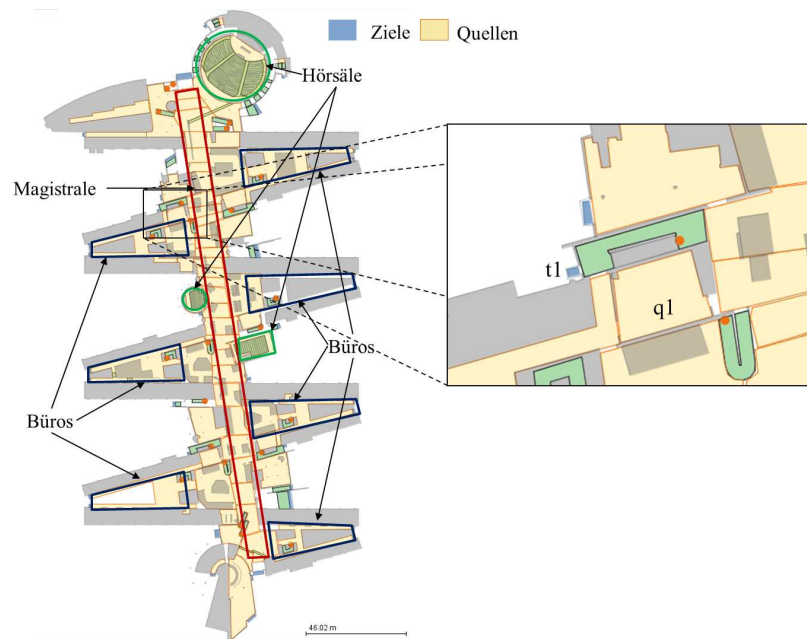


Abbildung 4.17: Szenario 3: Universitätsgebäude

In den einzelnen Flügeln des Gebäudes, in denen die Büros angesiedelt sind, sowie in der Magistrale und in den Hörsälen wurden jeweils Quellen platziert. Zusätzlich gibt es noch Quellen auf allen Treppen, die Personen abbilden sollen, die aus höheren Stockwerken kommen. Die Ziele sind an allen Ausgängen des Gebäudes platziert. Der Weg mit der größten Dilatation führt von  $q1$  nach  $t1$ .

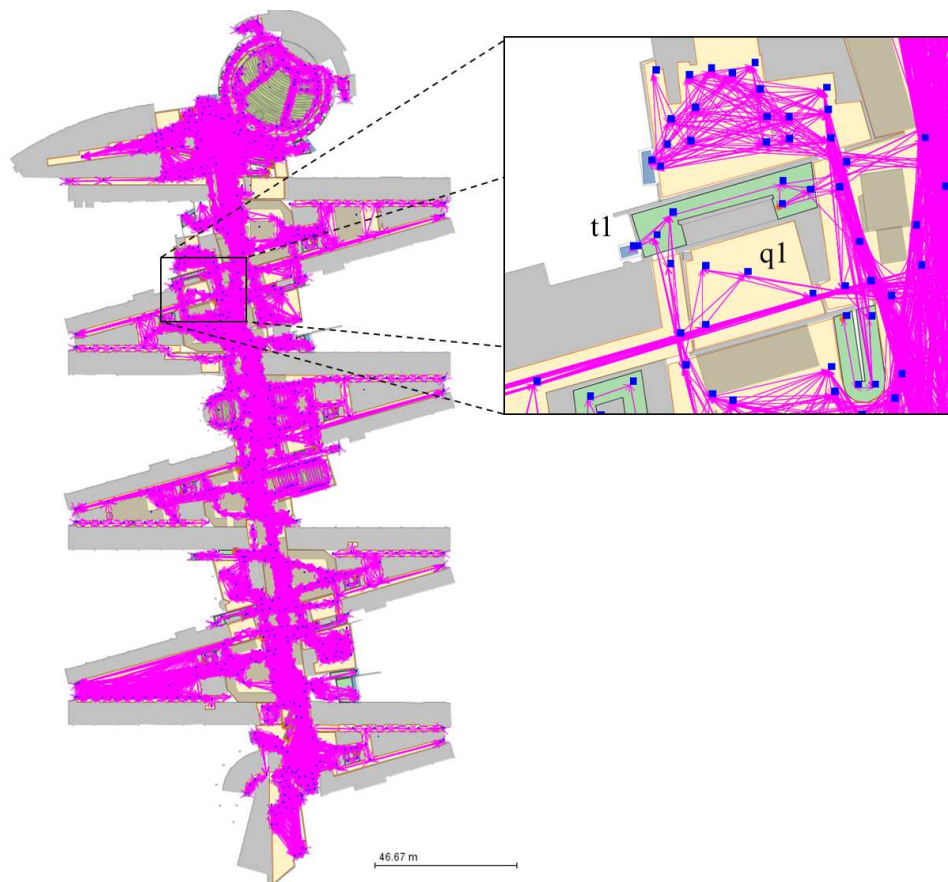


Abbildung 4.18:  $G_v$  für das Szenario Universitätsgebäude

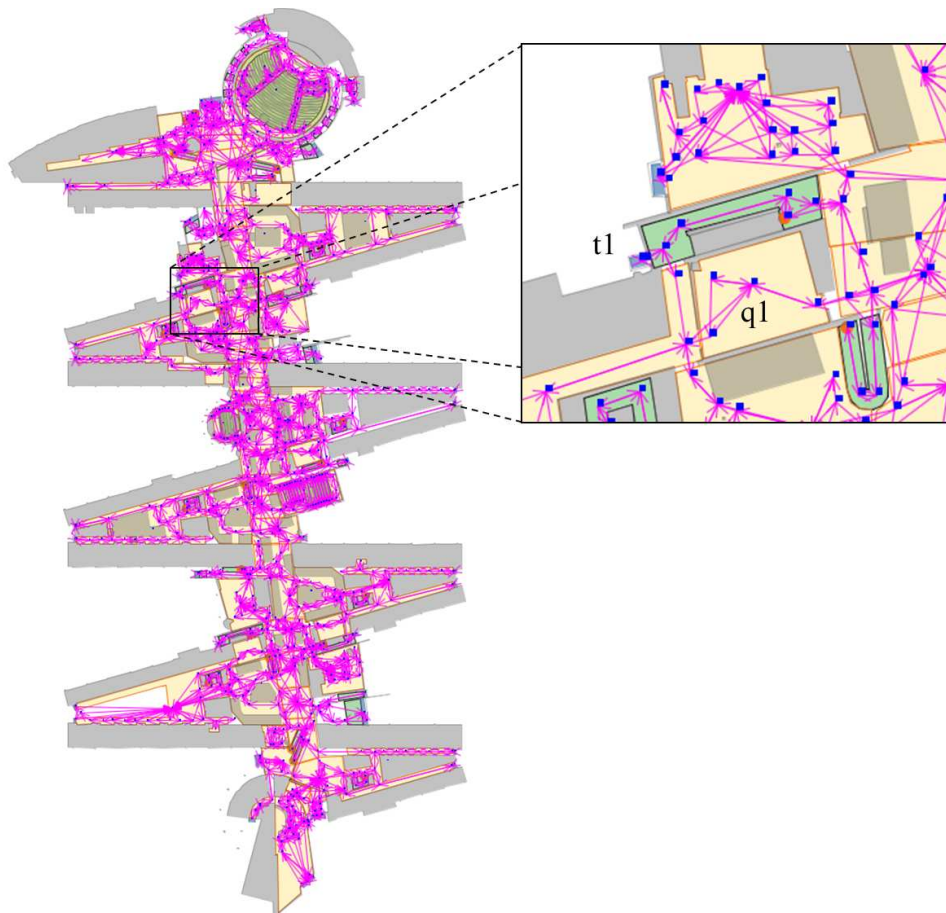


Abbildung 4.19:  $G_a$  mit  $\alpha_{zul} = 125.66^\circ$  für das Universitätsgebäude

**Szenario 4: Stadt** Das letzte Szenario bildet einen Teil einer Stadt ab. Das Gebiet umfasst insgesamt 240 x 180 Meter und besteht ausschließlich aus Polygonen, die die Gebäude der Stadt darstellen. Es sind ebenfalls mehrere Quellen und Ziele enthalten, und hier wurde der Weg von  $q_4$  nach  $z_4$  als der Weg mit der größten Dilatation identifiziert. Das Szenario ist in Abbildung 4.20 dargestellt, der erstellte vollständige Graph in Abbildung 4.21 und der ausgedünnte Graph mit maximal zulässigem  $\alpha_{zul}$  ist in Abbildung 4.22 abgebildet.



Abbildung 4.20: Szenario 4: Stadt



Abbildung 4.21:  $G_v$  für das Szenario Stadt

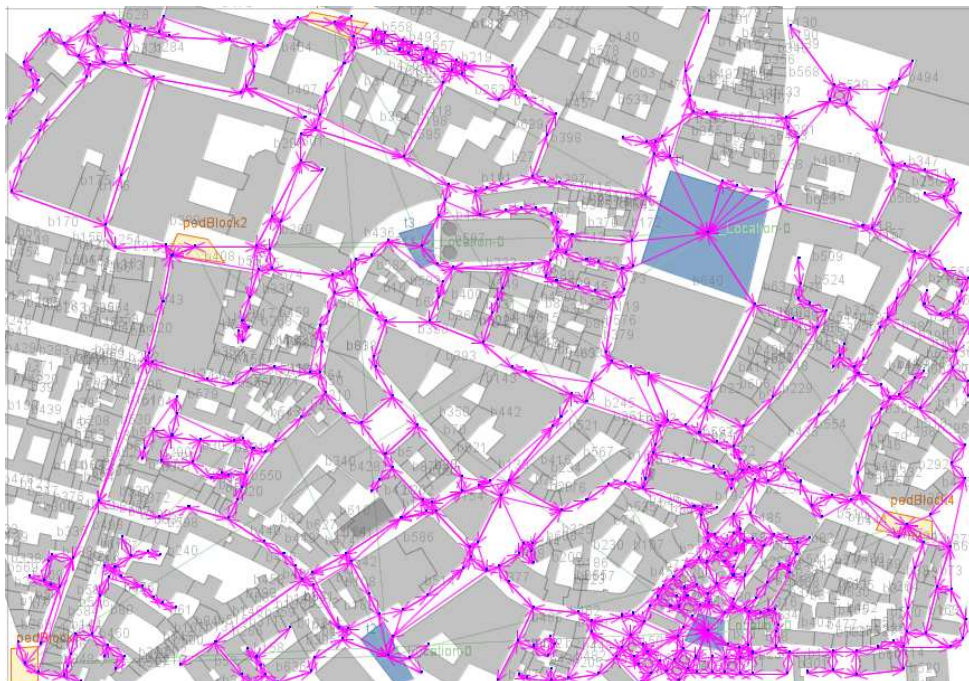


Abbildung 4.22:  $G_a$  mit  $\alpha_{zul} = 143.44^\circ$  für das Szenario Stadt



Für jedes dieser Beispielszenarien wird das Kantenverhältnis sowie die Dilatation in Relation zu  $\alpha$  gesetzt. Bei der Dilatation wird zudem der maximal zulässige Wert von  $\alpha$  markiert, für den der Graph mindestens einen Weg von der Quelle zum Ziel enthält. Für das Kantenverhältnis wird – falls mehrere Quellen und Ziele im Szenario existieren – der Weg als Referenz genommen, der die größte Dilatation zwischen  $G_v$  und  $G_a$  mit maximal zulässigem  $\alpha$  aufweist.

Die Ergebnisse sind in Abbildung 4.23 und Abbildung 4.24 dargestellt.

Je größer das Szenario ist, desto schneller verringert sich die Kantenanzahl für steigende  $\alpha$ -Werte. Während der Winkel bei dem kleinsten Szenario noch sehr groß gewählt werden kann, pendelt sich der Winkel bei den größeren Szenarien um  $100^\circ$  ein. Eine Ausnahme bildet das Bürogebäude. Da dieses Szenario hauptsächlich aus rechtwinkligen Ecken besteht, verwundert es nicht, dass der Wert des maximal zulässigen  $\alpha$  entsprechend gering ist. Gleichzeitig ist zu beobachten, dass die Anzahl der Kanten bei einer weiteren Erhöhung von  $\alpha$  nicht mehr signifikant reduziert wird. Dies ist bei allen vier Beispielszenarien zu beobachten. Der maximal zulässige  $\alpha$ -Wert hat bereits eine erhebliche Reduzierung der Kanten bewirkt. Mehr noch, selbst für kleine  $\alpha$ -Werte verringert sich die Kantenanzahl bereits signifikant.

In Tabelle 4.1 ist die Anzahl der Kanten des vollständigen und ausgedünnten Sichtbarkeitsgraphen aller Szenarien für  $\alpha_{zul}$  gegenübergestellt. Betrachtet man

	<b>Beispiel- szenario</b>	<b>Büro- gebäude</b>	<b>Universitäts- gebäude</b>	<b>Stadt</b>
$ E_{G_v} $	70	1019	17827	8804
$ E_{G_a} $	19	479	3337	1801

Tabelle 4.1: Anzahl der Kanten für die unterschiedlichen Szenarien

die Dilatation der jeweils kürzesten Wege, so erkennt man einen Anstieg in der Länge für größere Winkel. Während die Erhöhung der relativen Verschlechterung beim kleinsten Beispiel sehr sprunghaft verläuft, verläuft der Anstieg gemäßigt bei den anderen Szenarien. Erst bei Winkeln von mehr als  $100^\circ$  sind Sprünge zu beobachten. Da diese Werte für  $\alpha$  jedoch nicht mehr erlaubt sind (zwischen mindestens einem Quelle-Ziel Paar existiert kein Weg mehr) können diese Sprünge vernachlässigt werden.

Die Tatsache, warum höhere  $\alpha$  hier auftauchen, ist wie folgt erklärt: Der maximal zulässige Winkel ist so definiert, dass zwischen allen Quellen und Zielen Wege existieren. Da dieses  $\alpha$  nicht zwangsläufig für den Weg mit der größten Dilatation als Obergrenze gilt, kann es durchaus sein, dass der jeweils untersuchte Weg auch bei größeren Werten für  $\alpha$  noch existiert.

Zusammenfassend kann also gesagt werden, dass der im Rahmen dieser Arbeit entwickelte Algorithmus selbst schon für kleine  $\alpha$  eine erhebliche Reduzierung der Kanten bei gleichzeitig geringer Dilatation bewirkt. Nur für sehr große  $\alpha$  wächst die Dilatation teilweise sprunghaft, so dass solche Werte vermieden werden soll-

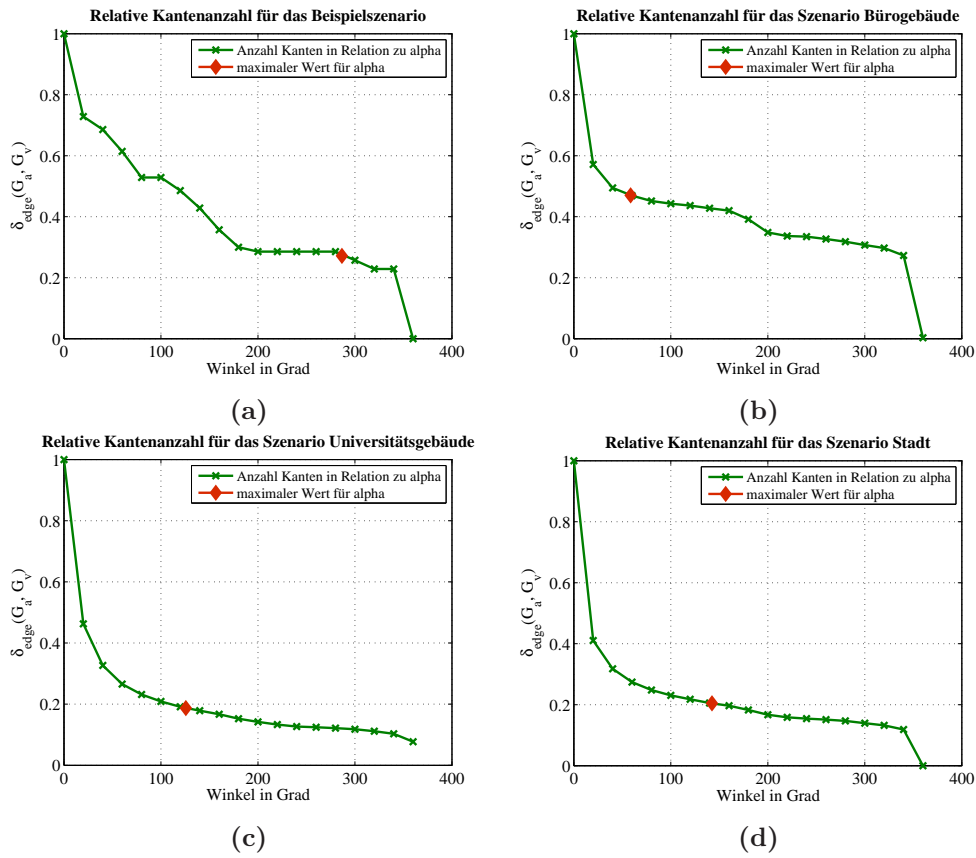


Abbildung 4.23: Anzahl der Kanten für die vier Testszenarien in Relation zu  $\alpha$ . Das maximale  $\alpha_{zul}$ , so dass mindestens ein Weg zwischen Quelle und Ziel erhalten bleibt, ist mit einer Raute markiert.

ten. Aufgrund des verwendeten Zellgitters aus der Lokomotionsebene, welches die Bewegung auf sechs Bewegungsrichtungen beschränkt, ist daher ein Maximalwert von  $\alpha = 60^\circ$  eine sinnvolle Wahl.

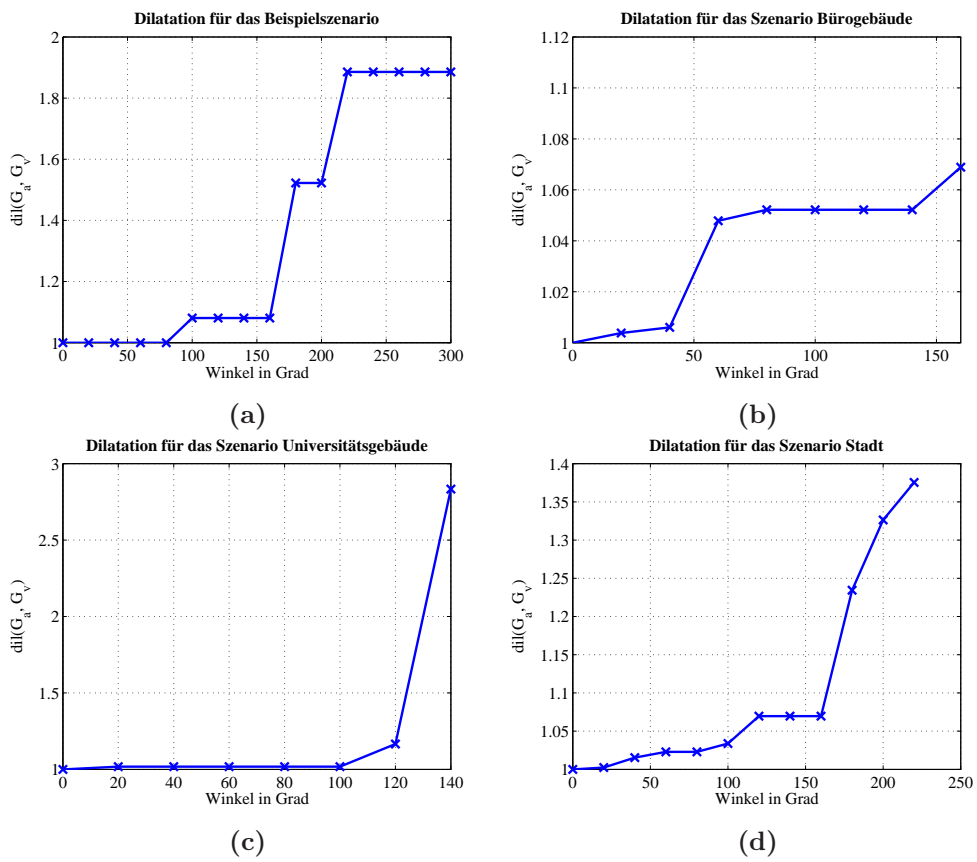


Abbildung 4.24: Dilatation für die vier Testszenarien in Relation zu  $\alpha$ .

## 4.4 Zusammenfassung

In diesem Kapitel wurde der Aufbau der taktischen Ebene des Simulationsmodells erläutert, die die Grundlage für die Untersuchungen dieser Arbeit bildet. Es wird ein ausgedünnter Navigationsgraph verwendet, der durch einen kegelbasierten Kantenerzeugungsalgorithmus effizient erstellt werden kann. In einer Studie wurde die Qualität des so erzeugten Graphen mit einem vollständigen Sichtbarkeitsgraph verglichen. Damit konnte gezeigt werden, dass der ausgedünnte Graph eine gleichwertige räumliche Abdeckung des Gebiets wie ein vollständiger Sichtbarkeitsgraph erzielt bei gleichzeitig erheblicher Reduktion der Kantenanzahl. Dieser Graph wird in Kapitel 5 als Schnittstelle zu einem makroskopischen Netzwerkmodell verwendet, um mikroskopische Aspekte in das Modell zu übertragen. In Kapitel 6 dient dieser Graph als Grundlage zur Entwicklung unterschiedlicher Navigationsalgorithmen zur Abbildung menschlichen Verhaltens bei der Wegfindung.

## Kapitel 5

# Der Navigationsgraph als Schnittstelle zwischen mikroskopischen und makroskopischen Simulationsmodellen<sup>1</sup>

### 5.1 Motivation für das Kopplungsverfahren

Wie in Kapitel 2 bereits beschrieben, haben mikroskopische und makroskopische Simulationsmodelle unterschiedliche Foki und damit einhergehend ihre Vorteile aber auch Schwachstellen. Ein Ziel von Fußgängersimulationen ist es, untere Grenzen für Evakuierungszeiten für ein betrachtetes Szenario zu ermitteln. Dazu können makroskopische Netzwerkflussmodelle (Chalmet *et al.*, 1982, Hamacher & Tjandra, 2002) eingesetzt werden. Diese Modelle finden optimale Lösungen unter der Annahme, dass jeder Fußgänger Routingentscheidungen fällt, die zu einer optimalen Benutzung der Fluchtwege und daher zu einer minimalen Gesamtevakuiierungszeit führen. Mikroskopische Aspekte des Fußgängerverhaltens wie Interaktion werden nicht beachtet. Die daraus resultierende Evakuierungszeit liegt somit in den meisten Fällen unter der realen Evakuierungszeit. Dennoch kann diese Zeit als valide Untergrenze für Evakuierungszeiten dienen.

Auf der anderen Seite betrachten mikroskopische Modelle die Interaktion und lokalen Phänomene zwischen den einzelnen Individuen, wie beispielsweise Bahnenbildung, Staus vor Engstellen oder das Abbremsen in dichten Menschenmengen. Die so berechneten Evakuierungszeiten können nicht als untere Grenze dienen, da die simulierten Fußgänger wenn überhaupt nur lokal optimale Entscheidungen treffen.

---

<sup>1</sup>Dieses Kapitel ist in Zusammenarbeit mit Markus Thiemann entstanden und ist an Anleh-

Um die Vorteile beider Verfahren nutzen zu können, wird hier ein neues methodisches Verfahren vorgeschlagen: die Kopplung des makroskopischen Ansatzes mit dem mikroskopischen Ansatz. Hierbei wird das makroskopische Modell, das den schnellsten Fluss in einem dynamischen Netzwerkflussmodell berechnet, dazu verwendet, optimale Routingstrategien zu ermitteln. Im Folgenden wird diese Komponente als Optimierung bezeichnet. Die Simulation liefert die mikroskopischen Aspekte des Fußgängerverhaltens und ermittelt eine präzisere Vorhersage zur Reisedauer auf den Kanten. In der Literatur sind ähnliche Ansätze kaum zu finden. Peng (2006) schlägt ein hybrides Modell vor, das Daten aus einem dynamischen Netzwerk verwendet. Dressler *et al.* (2010) berechnet frühest ankommende Flüsse (earliest arrival flow) in einem dynamischen Netzwerkflussmodell, um den zu Evakuierenden optimale Ausgänge zuzuweisen. Hamacher *et al.* (2011) verfolgen einen Sandwich-Ansatz, der die eigentlichen Evakuierungszeiten mit Hilfe von Obergrenzen und Untergrenzen wie ein Sandwich umschließen soll. Das hier vorgestellte Verfahren ist ebenfalls in Thiemann (2012) beschrieben.

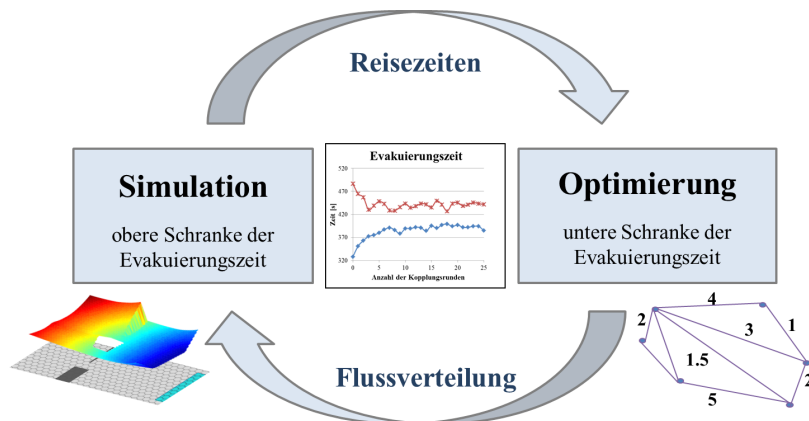


Abbildung 5.1: Aufbau des Kopplungsverfahrens: Auf Basis eines gemeinsamen Netzwerks ermitteln Simulation und Optimierung Evakuierungszeiten und tauschen Reisezeiten auf den Kanten sowie Flussverteilungen aus.

Dieses Kapitel stellt das neu entwickelte Konzept der Kopplung beider unterschiedlichen Ansätze dar, das ähnlich dem Konzept eines Regelkreises ist. In einem klassischen Regelkreis werden Ausgabewerte zurück in einen sog. Regler überführt, der der Abweichung vom Sollwert entgegenwirkt. Ähnlich ist auch hier das Vorgehen: Die Ausgabewerte des einen Verfahrens sollen durch das andere Verfahren (dem Regler) in Richtung einer realistischen Evakuierungszeit (dem Sollwert) überführt werden. Dies geschieht über mehrere Iterationen, bis die Evakuierungszeit innerhalb einer gewissen Schranke liegt oder gegen einen Wert konvergiert. Die Optimierung ermittelt eine untere Grenze der Evakuierungszeit, die im Laufe der Kopplung durch Rückführung der Werte aus der Simulation angehoben werden soll. Mit der Simulation wird eine Obergrenze der Evakuierungszeit ermittelt, die durch die Ausgabewerte der Optimierung nach

nung an Kneidl *et al.* (2011) sowie Borrmann *et al.* (2012) aufgebaut.

unten korrigiert wird.

Die gemeinsame Grundlage dafür bildet der in Kapitel 4 vorgestellte Navigationsgraph. Dieser wird zu einem Netzwerk erweitert, indem den Kanten zusätzliche Attribute zugewiesen werden. Die Optimierung ermittelt Verteilungsraten auf jeder Kante dieses Netzwerks, die dann in der Simulation verwendet werden; die Simulation wiederum ermittelt Reisezeiten auf den Kanten, die an die Optimierung zurückgegeben werden. Der schematische Aufbau des Verfahrens ist in Abbildung 5.1 dargestellt.

In Abschnitt 5.2 wird der Aufbau des makroskopischen Modells beschrieben, in Abschnitt 5.3 wird darauf eingegangen, welche Änderungen am bisherigen mikroskopischen Modell vorgenommen wurden, um dann die eigentliche Kopplung im Detail darzustellen (Abschnitt 5.4). Unterschiedliche Testfälle werden in Abschnitt 5.5 erläutert und demonstrieren die Funktionsweise des Verfahrens. In Unterabschnitt 5.5.2 wird ein Anwendungsbeispiel vorgestellt, bei dem eine minimale Kopplung eingesetzt wird.

## 5.2 Makroskopische Komponente: Dynamisches Netzwerkflussmodell

### 5.2.1 Quickest Flow Problem

Ein gegebenes Szenario wird mit einem diskreten, Zeit-dynamischen Netzwerk  $G = (N, A, T)$  modelliert, bei dem  $N$  die Menge der Knoten bezeichnet,  $A$  für die Kantenmenge steht und  $T$  einen endlichen Zeithorizont beschreibt, diskretisiert in Zeitschritte  $0, \dots, T$ .

Die Knotenmenge enthält einen Startknoten  $s \in N$  sowie einen Zielknoten  $t \in N$ . Für alle Zeitschritte  $\phi = 0, \dots, T$  wird jeder Kante  $(i, j) \in A$  eine zeitabhängige Reisezeit  $\tau_{ij}(\phi) \in \mathbb{Z}_0^+$  sowie eine zeitabhängige Kapazität  $u_{ij}(\phi) \in \mathbb{Z}_0^+$  zugeordnet.  $u_{ij}(\phi)$  limitiert die Anzahl der Flusseinheiten, die sich auf einer Kante  $(i, j)$  in Zeitschritt  $\tau$  befinden können. Flusseinheiten sind diskrete Einheiten des Gesamtflusses, der durch das Netzwerk geschickt wird.

Es wird angenommen, dass die Knotenkapazität zu allen Zeitpunkten 0 beträgt, auf den Knoten folglich keine Wartezeiten auftreten können.

Die Reisezeit  $\tau_{ij}(\phi)$  definiert die benötigte Zeit, um die Kante  $(i, j)$  zu traversieren, wenn die Flusseinheit den Knoten  $i$  zum Zeitpunkt  $\phi$  verlässt und zur Zeit  $\phi + \tau_{ij}$  ankommt.

Der Fluss ist als Funktion  $x : A \times 0, \dots, T \rightarrow \mathbb{Z}_0^+$  definiert, die jeder Kante nicht-negative Zahlen für alle Zeitschritte zuordnet. Das Problem lässt sich wie folgt

formal definieren:

$$\sum_{(j,i) \in A} \sum_{\{\theta': \theta' + \tau_{ji}(\theta') = \theta\}} x_{ji}(\theta') - \sum_{\{(i,j) \in A: \theta + \tau_{ij}(\theta) \leq T\}} x_{ij}(\theta) = 0, \forall i \in N \setminus \{s, t\}, \forall \theta \in \{0, \dots, T\} \quad (5.1)$$

$$x_{ij}(\theta) = 0, \forall \theta : \theta + \tau_{ij}(\theta) > T, \forall (i, j) \in A \quad (5.2)$$

$$0 \leq x_{ij}(\theta) \leq u_{ij}(\theta), \forall \theta : \theta + \tau_{ij} \leq T, \forall (i, j) \in A \quad (5.3)$$

Die Funktion erhält einerseits den Fluss (Gleichung 5.1) und andererseits werden die Kapazitäten der Kanten nie überstiegen (Gleichung 5.3). Zudem ist kein Fluss mehr im Netzwerk nach Ablauf der Zeit  $T$  vorhanden (Gleichung 5.2).

Eine sehr detaillierte Einführung in Netzwerkflussoptimierung ist in Ahuja *et al.* (1993) gegeben. Die Lösung des Quickest Flow Problems findet einen zulässigen dynamischen Fluss  $x$ , der eine gegebene Anzahl von Flusseinheiten  $U \in \mathbb{Z}_0^+$  von  $s$  nach  $t$  in kürzester Zeit  $T_U \leq T$  sendet. Mit oben beschriebenen Aufbau wird dieses Problem als *discrete-time quickest flow problem with time-dependent attributes* bezeichnet (Tjandra, 2003). Dazu seien folgende Definitionen gegeben:

Ein Residualnetzwerk beschreibt ein Netzwerk, in dem Restkapazitäten zu einem Fluss  $x$  angegeben werden. Restkapazitäten werden durch Einfügen von Rückwärtskanten dargestellt, so dass für jede Kante  $a_{ij}$  im Netzwerk  $N$  mit Kapazität  $u_{ij}$  eine Kante  $\overleftarrow{a}_{ij}$  mit Kapazität  $u_{\overleftarrow{a}} = u_{ij} - x_{u_{ij}}$  eingefügt wird. Falls  $x_{u_{ij}} \geq u_{ij}$ , so wird keine Kante eingefügt, da keine Restkapazitäten auf der Kante vorhanden sind.

Ein Pfad in einem solchen Residualnetzwerk wird als augmentierender Pfad bezeichnet. Der von Tjandra beschriebene Algorithmus löst das *discrete-time quickest flow problem with time-dependent attributes*, indem sukzessive augmentierende Pfade mit frühester Ankunftszeit (*earliest arrival time*) in einem dynamischen Residualnetzwerk von der Quelle zum Ziel gefunden werden. Wenn die Gesamtflussmenge entlang der augmentierenden Pfade den initialen Fluss übersteigt, so entspricht der Zeitpunkt des letzten augmentierenden Pfades dem Quickest Flow. Weitere Details sowie eine exakte mathematische Formulierung sind Tjandra (2003, S.83ff) zu entnehmen. Im Kontext von Evakuierungsszenarien bezeichnet dieser schnellste Fluss die minimale Evakuierungszeit für eine bekannte Anzahl von Menschen von einer Quelle  $s$  zu einem Ziel  $t$ . Die Funktion  $x$  gibt hierbei die Lage und den Verlauf der optimalen Evakuierungsrouten im Netzwerk vor.

### 5.2.2 Aufbau des Netzwerks für die Kopplung

Um Fußgängerbewegungen mit Hilfe eines dynamischen Netzwerkflusses zu modellieren, werden Korridore, Wege, Straßen etc. als Kanten dargestellt. Jede Kante  $(i, j) \in A$  hat eine feste Breite  $w_{ij}[m]$  und eine feste Länge  $l_{ij}[m]$ .

Im Kopplungssetup wird die maximale Flussrate pro Einheitsbreite  $M_{ij}[1/ms]$  für jede Kante vordefiniert und im Folgenden als Spezifische Flussrate (*SFR*) der Kante  $(i, j)$  bezeichnet.

Die Zeitschrittlänge im Netzwerk wird auf  $z = 1s$  festgelegt. Basierend auf diesen



Parametern wird die Kapazität der Kante  $(i, j)$  als  $u_{ij} = \lfloor M_{ij} \cdot w_{ij} \cdot 1/z \rfloor$  definiert. Die Kapazität bleibt konstant über die Zeit. Zudem wird angenommen, dass die Durchschnittsgeschwindigkeit  $v_{ij}(\theta)$  auf jeder Kante  $(i, j) \in A$ ,  $\theta = 0, \dots, T$  mit zugehöriger Reisezeit  $\tau(\theta) = \lfloor l_{ij} \cdot v_{ij} \cdot z \rfloor$  bekannt ist.

In Abbildung 5.2 ist ein Beispiel zur Modellierung des Netzwerks illustriert.

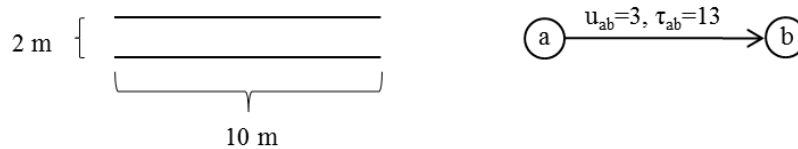


Abbildung 5.2: Aufbau des dynamischen Netzwerks: Beispiel für einen Korridor (links) und die zugehörige Kante im Netzwerk. Hier wird die Geschwindigkeit  $v = 1.3ms^{-1}$  und die Spezifische Flussrate auf  $M = 1.5(ms)^{-1}$  gesetzt

### 5.3 Mikroskopische Komponente: Erweiterung des mikroskopischen Modells für die Kopplung

Um die Kopplung mit dem makroskopischen Netzwerk zu ermöglichen, wird der Navigationsgraph des mikroskopischen Simulationsmodells angepasst: Zusätzlich zu den Knoten und Kanten des automatisch generierten Graphen werden die Länge und Breite der Kante abgespeichert. Da die Breite für Kanten in Szenarien, die keine seitlichen Begrenzungen durch Hindernisse haben, nicht ohne weiteres ermittelt werden kann, werden hier im Folgenden Szenarien betrachtet, deren Wege durch Wände beschränkt sind. Eine Erweiterung auf Szenarien ohne Beschränkung kann durchgeführt werden, die Breite der Kante muss dann aber ggf. per Hand gesetzt werden. Der Graph wird in Kapitel 6 dazu genutzt, um Fußgänger mit unterschiedlichem Navigationsverhalten über Zwischenziele von der Quelle zum Ziel zu navigieren. Hier wird kein Algorithmus zum Finden des Wegs genutzt, es wird vielmehr eine vom makroskopischen Netzwerk gegebene Verteilungsfunktion verwendet, die vorgibt, wie die Fußgänger an jedem Knoten auf die ausgehenden Kanten verteilt werden.

### 5.4 Aufbau und Parameter des Kopplungsverfahrens

Zur Kopplung der beiden Modelle werden zunächst einige Parameter definiert: Eine Kopplungsrunde besteht aus einem Optimierungslauf, gefolgt von einem Simulationslauf und ist in Abbildung 5.3 dargestellt. Es gibt fixierte Parameter, die sich während des gesamten Kopplungsverfahrens nicht verändern und dynamische Parameter, die nach jeder Kopplungsrunde angepasst werden. Das gesamte Kopplungsverfahren, angewandt auf ein Szenario mit  $n$  Iterationen wird als Kontrollzyklus bezeichnet.

### 5.4.1 Parameter für das Kopplungsverfahren

#### Fixierte Parameter

Die fixierten Parameter sind in Tabelle 5.1 zusammengefasst und werden im Folgenden detailliert beschrieben.

Name	Beschreibung
Szenario	Szenario inklusive Netzwerk mit Kantenbreite und -länge
$N$	Anzahl der Fußgänger
$t$	Intervallgröße
$SFR$	Spezifische Flussrate für jede Kante

Tabelle 5.1: Fixierte Parameter für die Kopplung

**Szenario inklusive Netzwerk** – Ein Szenario besteht aus einer Quelle und einem Ziel sowie beliebig vielen Zwischenzielen. Das Netzwerk wird aus der Szenariogeometrie abgeleitet und erstellt.

**Anzahl der Fußgänger** – Die Anzahl der Fußgänger muss groß genug sein, um eine Interaktion zwischen Fußgängern beobachten zu können. Die Dichte auf den Kanten muss folglich so groß sein, dass die Wunschgeschwindigkeit der Fußgänger nicht erreicht werden kann.

**Intervallgröße** – Die Intervallgröße ist die gemeinsame Zeitschrittgröße zum Austausch der Parameter zwischen beiden Modellen. Innerhalb eines Zeitintervalls werden Mittelwerte der entsprechenden Parameter gebildet, die an das andere Modell übergeben werden.

**SFR** – Die Spezifische Flussrate einer Kante beschreibt die maximale Anzahl von Fußgängern, die pro Einheit Breite in einer Sekunde die Kante traversieren können. Die  $SFR$  wird für jede Kante in einem Vorverarbeitungsschritt ermittelt.

#### Dynamische Parameter

Die dynamischen Parameter sind in Tabelle 5.2 aufgelistet und werden im Folgenden beschrieben.

**DR** – Zeitabhängige Verteilungsrate für jeden Knoten und Zeitschritt.  $DRs$  definieren die Flussverteilung an jedem Knoten an die ausgehenden Kanten, d.h. wie der ankommende Fluss am Knoten  $v_i$  auf alle ausgehenden Kanten verteilt wird. Die Verteilungsraten summieren sich für jeden Knoten auf 1 und können in jedem Zeitintervall unterschiedlich sein.  $DRs$  werden in der Optimierung ermittelt und über die Zeitintervalllänge gemittelt. In der Simulation erfolgt die Verteilung der Fußgänger entsprechend der  $DRs$  auf die ausgehenden Kanten.

**SFQ** – Quellflussmenge.  $SFQ$  beschreibt die Anzahl der Flusseinheiten, die in jedem Zeitschritt die Quelle verlassen. Die Quellflussmenge wird vom Optimie-

Name	Beschreibung
$DR$	Verteilungsrate an jedem Knoten pro Zeitschritt
$SFQ$	Quellflussmenge ( <b>S</b> ource <b>F</b> low <b>Q</b> uantity) für jeden Zeitschritt
$v$	Durchschnittsgeschwindigkeit für jede Kante pro Zeitschritt

Tabelle 5.2: Dynamische Parameter für die Kopplung

rungsnetzwerk erzeugt und dient im Simulationslauf dazu, die Anzahl der Personen zu bestimmen, die pro Zeitschritt die Quelle verlassen. In der Simulation können Interaktionen zwischen Fußgängern zu einem Stau führen, so dass die von der Optimierung vorgegebene Anzahl der Fußgänger in einem Zeitschritt nicht erzeugt werden kann. Diese werden dann im nächsten Zeitschritt generiert.

Die Anzahl der effektiv generierten Fußgänger wird als Feedback an die Optimierung zurück gegeben, um als Referenz zu dienen. Die Anpassung der Anzahl der Flusseinheiten innerhalb der Optimierung funktioniert folgendermaßen: Falls die optimale Anzahl der Flusseinheiten innerhalb des vorgegebenen Zeitschritts in der Simulation nicht erreicht wird, so wird für diesen Zeitschritt die Anzahl der Flusseinheiten auf den Wert reduziert, der von der Simulation zurückgegeben wurde.

$v$  – Durchschnittsgeschwindigkeit pro Kante pro Zeitintervall. Die Simulation liefert für jedes Zeitintervall die Durchschnittsgeschwindigkeit pro Kante zurück. In einer Kopplungsrunde  $i \in 1, \dots, n$ ,  $n = \text{Anzahl der Kopplungsrunden}$ , werden die Geschwindigkeiten folgendermaßen eingelesen:

$$v_{regulation}^i = \alpha v^i + (1 - \alpha)v_{regulation}^{i-1}$$

Der Parameter  $v_{regulation}^0$  steht für die Anfangsgeschwindigkeit, die in der ersten Runde von der Optimierung verwendet wird.  $\alpha \in [0, 1]$  gewichtet die Durchschnittsgeschwindigkeiten der aktuellen Runde.

In Abbildung 5.3 ist der Austausch aller Parameter zwischen Optimierung und Simulation veranschaulicht.

### 5.4.2 Ablauf des Kontrollzyklus

Vor Beginn des Kontrollzyklus wird in einem Vorverarbeitungsschritt die Spezifische Flussrate  $SFR$  innerhalb der Simulation ermittelt. Dazu wird eine maximale Anzahl von Personen über jede Kante geschickt, um den Maximalwert der  $SFR$  zu bestimmen. Dies ist notwendig, da die Simulation eine räumliche Diskretisierung vornimmt. Somit können durch die Lage der Zellen eine unterschiedliche Anzahl von Personen innerhalb der Begrenzungen nebeneinander laufen.

Der aus der Szenariogeometrie erzeugte Graph wird zu einem dynamischen Netzwerk erweitert, indem für jede Kante zusätzlich zu den statischen Parametern

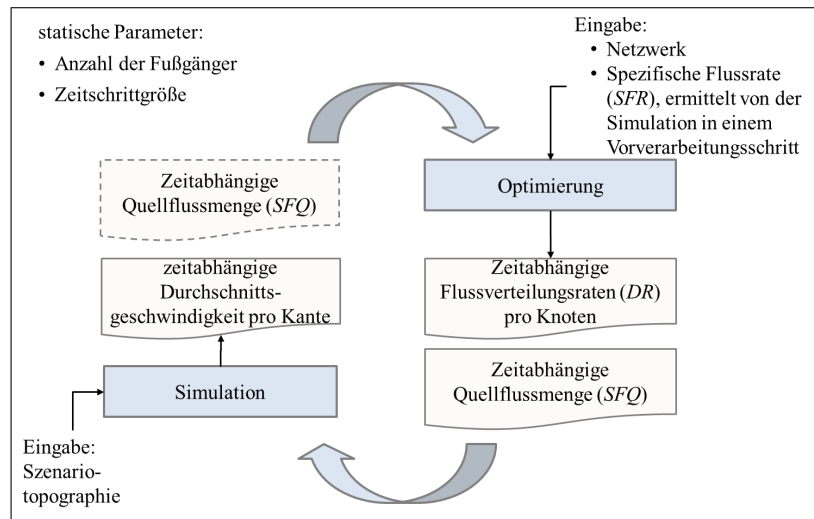


Abbildung 5.3: Kopplungsaufbau

Breite und Länge die Kapazität  $u$  und die Reisezeit  $t$  der Kante berechnet werden.

Die Kapazität wird dabei aus der gegebenen  $SFR$  und der Breite berechnet, die Reisezeit wird aus der Länge der Kante sowie einer gegebenen Gehgeschwindigkeit von 1.34 m/s nach Weidmann (1993) abgeleitet.

Innerhalb dieses dynamischen Netzwerks wird der schnellste Fluss berechnet, mit den zugehörigen zeitabhängigen Verteilungsraten  $DR$  sowie den zeitabhängigen Quellflussmengen  $SFQ$ . Diese beiden Parameter dienen der Simulation als Eingangsparameter. Innerhalb der Simulation werden die Fußgänger in Abhängigkeit der übergebenen  $SFQs$  in der Quelle erzeugt und abhängig der  $DRs$  entlang der Kanten bis zum Ziel geschickt. Die daraus resultierenden zeitabhängigen Durchschnittsgeschwindigkeiten für jede Kante werden an die Optimierung zurückgeliefert.

Anhand dieser neuen Durchschnittsgeschwindigkeiten wird eine Anpassung der Reisezeiten jeder Kante im dynamischen Netzwerk vorgenommen. Zudem werden die effektiven Quellflussmengen  $SFQs$  zurückgeliefert, um die Werte in der Optimierung anzupassen, falls diese von den in der Optimierung erzeugten Quellflussmengen  $SFQs$  abweichen. Die Optimierung berechnet erneut den schnellsten Fluss mit den angepassten Parametern.

Die daraus resultierenden  $SFQs$  und  $DRs$  werden wieder an die Simulation übergeben und ein neuer Simulationslauf mit den angepassten Eingangsparametern wird gestartet.

Ziel ist es, denselben Wert für den von der Optimierung berechneten schnellsten Fluss und der aus der Simulation resultierenden Evakuierungszeit zu erreichen. Da dies aufgrund verschiedener Gründe, die in Unterabschnitt 5.5.1 erläutert werden, nicht möglich ist, wird dieser Zyklus entweder mit einer festen Anzahl von Iterationen durchgeführt oder so lange wiederholt, bis der Unterschied zwischen dem schnellsten Fluss aus der Optimierung und der berechneten Evakuierungs-

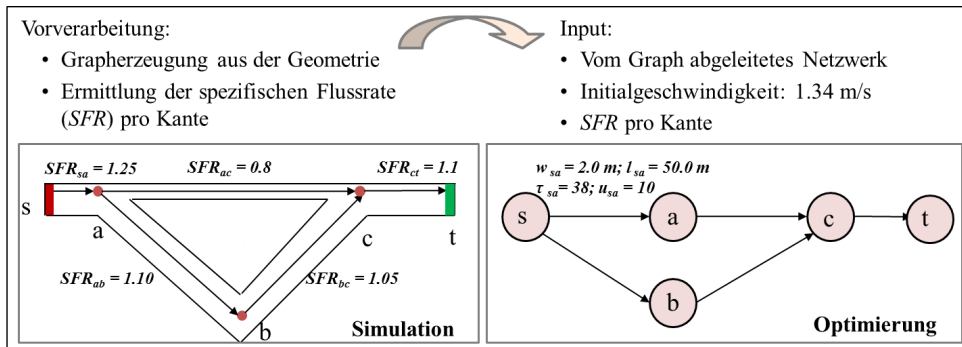


Abbildung 5.4: Vorverarbeitungsschritt innerhalb der Simulation erzeugt die Eingangsparameter für den ersten Optimierungslauf: Der Graph sowie die *SFR* werden generiert.

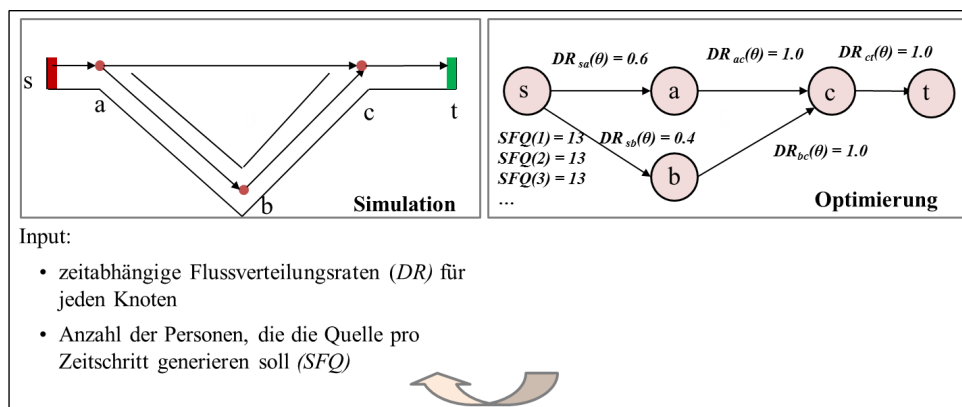


Abbildung 5.5: Nach der Berechnung des schnellsten Flusses werden die zeitabhängigen Parameter *DR* und *SFQ* an die Simulation übergeben

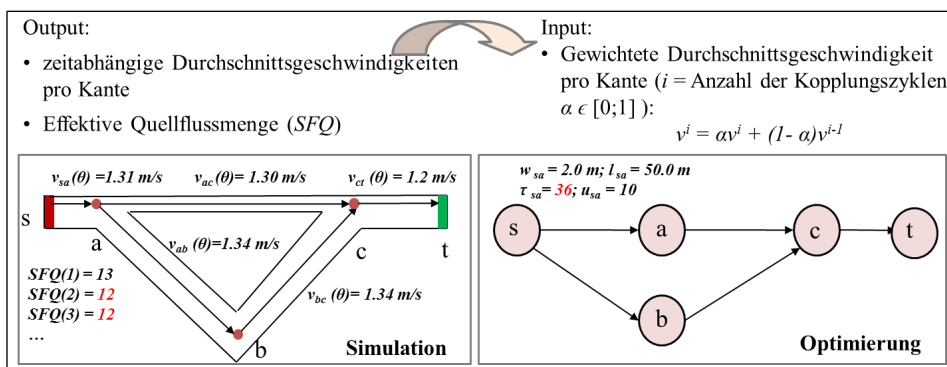


Abbildung 5.6: Die Simulation übergibt die aktualisierten Geschwindigkeiten sowie die effektive Quellflussmenge an die Optimierung; die Optimierung aktualisiert die Reisezeiten für jede Kante; nach dem Optimierungslauf werden die geänderten Parameter *DR* und *SFQ* wieder an die Simulation zurückgegeben. Der Zyklus setzt sich wie in Abbildung 5.5 fort.

zeit aus der Simulation unter einer gewissen Schranke liegt.

Der Ablauf einer Kopplungsrunde ist in den Abbildungen 5.4 bis 5.6 dargestellt. In Abbildung 5.4 wird der Vorverarbeitungsschritt veranschaulicht, Abbildung 5.5 zeigt die Übergabe der Optimierungsergebnisse an die Simulation und Abbildung 5.6 illustriert die Übergabe der Ergebnisse der Simulation an die Optimierung.

## 5.5 Tests

Zur Überprüfung des beschriebenen Verfahrens werden unterschiedliche Testfälle mit unterschiedlichen Konfigurationen betrachtet. Um die Funktionsweise zu testen sowie Parameterstudien durchzuführen, werden in Unterabschnitt 5.5.1 schematische Testfälle konstruiert und diese mit unterschiedlichen Einstellungen getestet. In Unterabschnitt 5.5.2 wird ein größeres Anwendungsbeispiel betrachtet, bei dem das Kopplungsverfahren eingesetzt wird.

### 5.5.1 Schematische Testfälle

#### Testfall 1: Engstelle und Umweg

Ein repräsentatives Testszenario ist in Abbildung 5.7 dargestellt, an dem Parameterstudien durchgeführt werden. Das Szenario besteht aus einem Korridor, der sich bei Knoten  $a$  in einen kürzeren Korridor mit Engstelle und einen längeren Korridor mit konstanter Breite aufteilt. Am Ziel treffen die beiden Korridore wieder aufeinander. Das daraus erstellte Netzwerk ist in Abbildung 5.8 dargestellt. Es wird untersucht, inwiefern der Einfluss der Rückkopplung  $SFQ$  mit der Simulation Auswirkungen auf die Ergebnisse hat. Weiter wird die Länge der Zeitintervalle zur Ermittlung der Durchschnittswerte, also die Häufigkeit des Parameteraustauschs zwischen Optimierung und Simulation, untersucht. Zudem wird der Unterschied in der Verwendung von uniformen Geschwindigkeiten für alle Fußgänger im Gegensatz zu gaußverteilten Gehgeschwindigkeiten untersucht. Eine Auflistung der unterschiedlichen Konfigurationen ist in Tabelle 5.3 gegeben. In

Konfiguration	SFQ Rückkopplung	Zeitintervall[s]	Gaußverteilte Geschwindigkeit
SFQ10UV	ja	10	nein
noSFQ10UV	nein	10	nein
SFQ5UV	ja	5	nein
noSFQ5UV	nein	5	nein
noSFQ10GV	ja	10	ja

Tabelle 5.3: Getestete Konfigurationen

Tabelle 5.4 sind die Parameter aufgelistet, die bei allen Konfigurationen unverän-

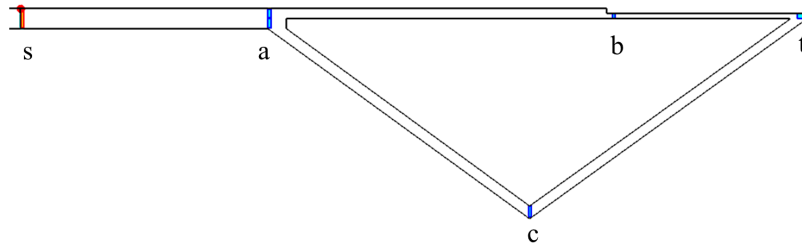


Abbildung 5.7: Das Szenario besteht aus einer Kreuzung bei Knoten *a* sowie einer Engstelle bei Knoten *b*

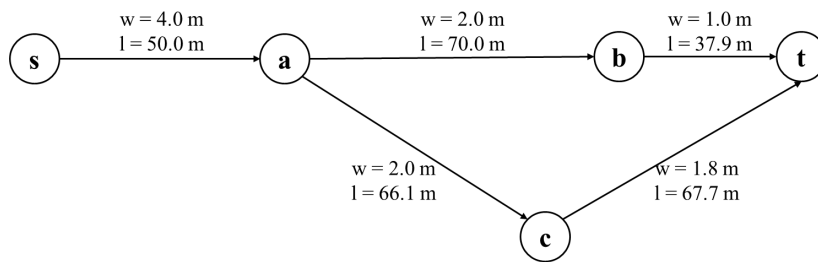


Abbildung 5.8: Das aus dem Szenario abgeleitete Netzwerk

derlich bleiben. Für die Simulation wird die Evakuierungszeit als der Zeitpunkt festgelegt, zu dem 99 % aller Fußgänger im Ziel angekommen sind. Fußgänger können in der Simulation von dem eigentlich zugewiesenen Weg (Kante) abgedrängt werden, falls eine hohe Dichte an einem Knoten (z.B. Engstelle, Kreuzung) auftritt. Da aber in der Kopplung der Fußgänger dem ihm zugewiesenen Pfad unbedingt folgen muss, führt dies dazu, dass der Fußgänger wieder zu seinem ursprünglichen Pfad zurückgeht, sobald sich der Stau aufgelöst hat. Dies hat zur Folge, dass diese Fußgänger sehr spät ins Ziel kommen. Ohne die 99 % Regel verzerren diese Fußgänger das Ergebnis. Es ist wichtig zu erwähnen, dass diese Regel die Schwachstellen der Simulation aufheben soll, nicht aber Fußgänger mit geringen Geschwindigkeiten ignoriert. Im Folgenden wird der Begriff Simulationszeit als Synonym für diese definierte Evakuierungszeit verwendet.

Die Ergebnisse der ersten vier Testkonfigurationen aus Tabelle 5.3 sind in Abbildung 5.9 zusammengefasst. Für jede der 25 Runden ist jeweils die Simulationszeit sowie der schnellste Fluss aufgetragen. Die Graphen zeigen als nullte Kopplungsrunde jeweils die Zeiten, die beide Modelle ohne Kopplung generieren. Bei allen vier Graphen kann festgestellt werden, dass die Zeiten des schnellsten Flusses innerhalb der ersten Runden stetig anwachsen. In den restlichen Zyklen oszilliert der schnellste Fluss um einen stationären Zustand.

Die Simulationszeit wird von dem ursprünglichen Wert Runde für Runde verringert bis sie sich einem stationären Zustand annähert. Auch hier können die typischen Oszillationen eines Regelkreises beobachtet werden. Die Oszillationen sind größer als beim schnellsten Fluss. Der Grund hierfür liegt in der Schwie-

Parameter	Wert
Anzahl der Fußgänger	1000
$\alpha$	0.3
Mittlere Geschwindigkeit zu Beginn der Kopplung	1.34 m/s
Anzahl der Kopplungsrunden	25
SFR	Im Vorverarbeitungsschritt berechnet

Tabelle 5.4: Parameter für alle Konfigurationen

rigkeit, die Fußgänger vor der Kreuzung bei  $a$  zu kontrollieren. Eine geringfügig andere Verteilung an der Kreuzung bewirkt dort einen geringfügig größeren oder kleineren Stau und führt dadurch zu einer niedrigeren oder höheren Evakuierungszeit.

Dennoch bleibt festzuhalten, dass eine Annäherung der Evakuierungszeiten beider Komponenten bei allen vier Konfigurationen zu sehen ist.

Die Erhöhung des schnellsten Flusses resultiert aus den erhöhten Reisezeiten auf den Kanten im Optimierungsnetzwerk, die in den Simulationsläufen generiert werden: Die Simulation bildet Interaktionen zwischen Fußgängern ab, die die Fußgänger abbremsen. Daher gibt die Simulation eine realistischere mittlere Geschwindigkeit auf den Kanten an das Netzwerk zurück, die unterhalb der standardmäßig angesetzten Geschwindigkeit der Optimierung liegt.

Die genauere Simulationszeit resultiert aus der angepassten Quellflussmenge  $SFQ$  sowie der angepassten Verteilungsrate  $DR$  an der Kreuzung bei Knoten  $a$ : In der Simulation ohne Optimierunginput nehmen alle Fußgänger den kürzesten Weg von Knoten  $a$  über  $b$  nach  $t$ . Nach einiger Zeit entsteht ein Stau vor der Engstelle  $b$ , der die Fußgänger abbremst. Die Anpassung der Verteilungsrate  $DR$  stellt sicher, dass für bestimmte Zeitschritte einem Teil der Fußgänger der Weg von  $a$  über  $c$  nach  $t$  zugewiesen wird. Dies führt zu einem kleineren Stau vor dem Flaschenhals bei  $b$ , den die Fußgänger schneller passieren können. Nach einigen Kopplungsrunden erreichen beide Fußgängerströme das Ziel  $t$  fast gleichzeitig. Dies entspricht genau dem Ergebnis, das mithilfe eines Optimierungsnetzwerk erzielt werden soll.



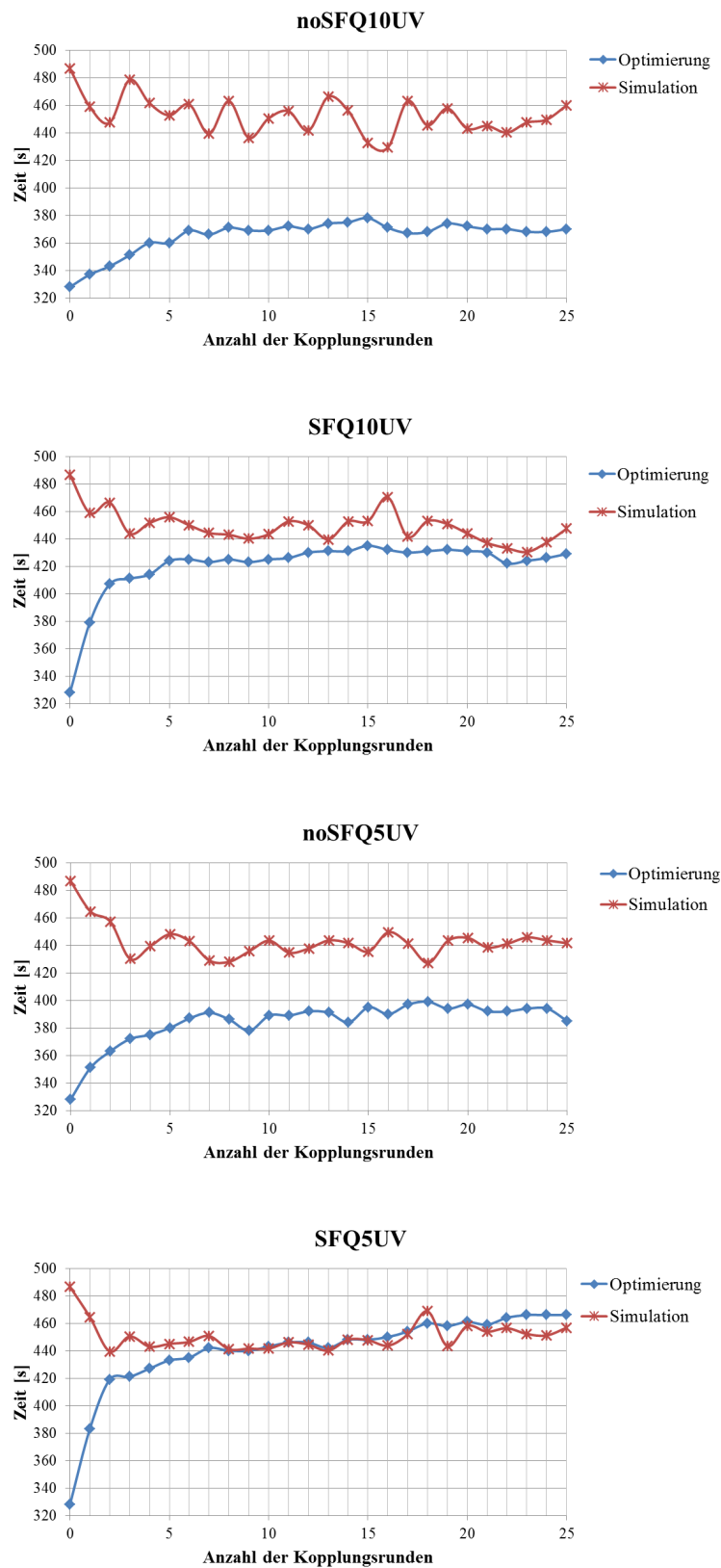


Abbildung 5.9: Ergebnisse der Kopplung des in Abbildung 5.7 dargestellten Szenarios.

Das Feedback der Quellflussmenge  $SFQ$  beeinflusst die Ergebnisse folgendermaßen: Die Kapazitäten der Kanten im Netzwerk sind begrenzt durch die maximale  $SFR$  in der Simulation. Jeder zulässige Fluss in dem dynamischen Netzwerk entspricht maximal der  $SFR$ . Somit entstehen keine Staus, da die maximale Kapazität der Kanten nie überschritten wird. Indem die Werte für die  $SFQ$  in jedem Zeitintervall angepasst werden und die Fußgänger entlang der Pfade geschickt werden, die dem optimalen Fluss entsprechen, erhält auch die Simulation auf ähnliche Weise die  $SFR$  und es entstehen keine Staus (speziell vor Engstellen). Dies wiederum verringert die Gesamtzeit, die benötigt wird, bis alle Fußgänger das Ziel erreicht haben. Auf diese Weise liegen die Ergebnisse der Optimierung und Simulation näher zusammen.

Vergleicht man die Konfiguration  $SFQ10UV$  und  $noSFQ10UV$ , so zeigt sich der positive Einfluss des Feedbacks der effektiven Quellflussmenge  $SFQ$  von der Simulation an die Optimierung auf die Annäherung der Evakuierungszeiten. Bei der Intervallgröße von 10 Sekunden wird beinahe eine Konvergenz der Zeiten zwischen beiden Modellen erreicht. Die effektive Quellflussmenge reduziert die Netzwerkkapazitäten, wann immer die Simulation nicht dazu in der Lage ist, die von der Optimierung vorgegebene Anzahl von Personen zu generieren. Wenn diese Rückmeldung nicht vorhanden ist, beginnen die beiden Modelle sich zu entkoppeln und konvergieren jeweils gegen einen anderen stationären Zustand (siehe Konfiguration  $noSFQ10CV$ ).

Auch bei den beiden Konfigurationen  $SFQ5UV$  und  $noSFQ5UV$  ist derselbe Effekt des  $SFQ$ -Feedbacks zu erkennen. Während ohne Feedback beide Zeiten gegen einen unterschiedlichen Wert konvergieren, liegt mit Feedback die durch die Optimierung ermittelte Evakuierungszeit sogar oberhalb der Simulationszeit. Ein Grund dafür ist, dass Durchschnittsgeschwindigkeiten, die aufgrund von Staus entstehen, aus kurzen Intervallen gemittelt werden und damit sehr gering sind. Angenommen, es hat sich ein kleiner Stau vor der Engstelle gebildet, der sich innerhalb von fünf Sekunden wieder auflöst. So kann es passieren, dass bei einer zu kleinen Intervallwahl die Geschwindigkeit auf 0 absinkt für diese fünf Sekunden. Die gemittelten Werte sind damit zu sensitiv gegenüber kurzzeitigen Veränderungen. Dies hat zur Folge, dass in den nächsten Kopplungsrunden die  $SFQs$  verringert werden. Während die Geschwindigkeiten in der nächsten Kopplungsrunde wieder steigen, da keine Staus auftreten, werden die  $SFQs$  nicht mehr steigen, da die von der Simulation ermittelten  $SFQs$  nur von der Optimierung übernommen werden, falls sie geringer sind.

Das führt letztendlich zu einer strikten Kapazitätsgrenze auf der Optimierungsseite, so dass der Wert des schnellsten Flusses immer weiter steigen wird. Deshalb muss mit der Einstellung der beiden Parameter  $SFQs$  und Intervallgröße sehr vorsichtig umgegangen werden. Beide Parameter beeinflussen sich sehr sensibel und können eine Kettenreaktion auslösen, die zur Folge haben kann, dass der schnellste Fluss oberhalb der Simulationszeit liegt.

Deshalb werden die nächsten beiden Konfigurationen aus Tabelle 5.3 mit einem Zeitintervall von 10 Sekunden und mit  $SFQ$ -Feedback getestet. Diese Einstellung hat sich bei den bisherigen Testfällen als am günstigsten erwiesen. Im Unterschied zu den bisherigen Tests wird jedem Fußgänger eine individuelle Geschwindigkeit

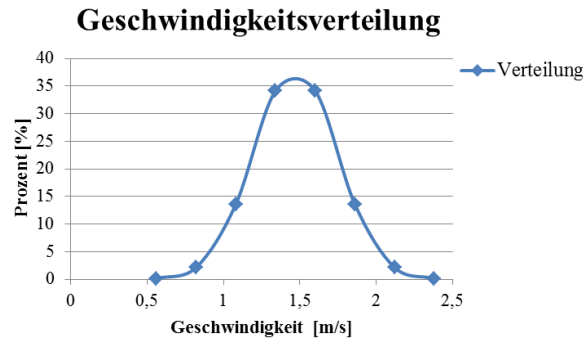


Abbildung 5.10: Geschwindigkeitsverteilung der Fußgänger

zugeteilt. Diese Geschwindigkeit bleibt über alle Kopplungsrunden gleich. Die Geschwindigkeiten sind gaußverteilt in Anlehnung an Weidmann (1993) mit einer Standardabweichung von  $\sigma = 0.26$  und einem Mittelwert von  $1.34$  m/s. Die Verteilung ist in Abbildung 5.10 dargestellt.

Die Ergebnisse der Kopplung, angewandt auf das oben beschriebene Szenario, sind in Abbildung 5.11 zu sehen. Zum besseren Vergleich ist die Konfiguration *SFQ10UV* gegenübergestellt. Die Lücke zwischen den Evakuierungszeiten von Simulation und Optimierung schließt sich auch für gaußverteilte Geschwindigkeiten fast. Für den Simulationslauf ohne Optimierung in Runde 0 ist allerdings zu beobachten, dass die Evakuierungszeit für gaußverteilte Geschwindigkeiten signifikant höher ist als für uniforme Geschwindigkeiten. Mit den gaußverteilten Geschwindigkeiten gibt es langsame Fußgänger, die später ins Ziel kommen, obwohl sie den kürzesten Weg nehmen. Dies hat eine Erhöhung der Gesamtzeit zur Folge. Deshalb sind auch nach einigen Kopplungsrunden die Zeiten sowohl für Optimierung als auch für Simulation höher.

Die Lücke zwischen Optimierung und Simulation für gaußverteilte Geschwindigkeiten bleibt dennoch größer als für uniforme Geschwindigkeiten. Das ergibt sich aus der größeren Anzahl unterschiedlicher Geschwindigkeiten: Für jeden Zeitschritt gibt es Fußgänger mit höheren und niedrigeren Geschwindigkeiten als die gemittelte Geschwindigkeit. Die Reisezeiten werden anhand der Durchschnittsgeschwindigkeiten berechnet. Daher brauchen die langsameren Fußgänger in der Simulation deutlich länger, um eine Kante zu traversieren als die Reisezeit, die von der Optimierung berechnet wird.

Dieselbe Ursache hat noch eine andere Auswirkung: Die Kurven der Simulationszeit sind nicht so glatt wie in dem Fall mit konstanten Geschwindigkeiten. Die Optimierung gibt Verteilungsraten für die Zeitintervalle vor, in denen Fußgänger im Mittel an einem Knoten ankommen. Da es schnellere und langsamere Fußgänger gibt, kommen nicht alle innerhalb des berechneten Intervalls an dem Knoten an, und werden daher anhand anderer Verteilungsraten weitergeschickt. Dieses Phänomen tritt in der Optimierung nicht auf – dort werden ausschließlich Durchschnittswerte verwendet. Die Simulation kann den optimalen Fluss nie

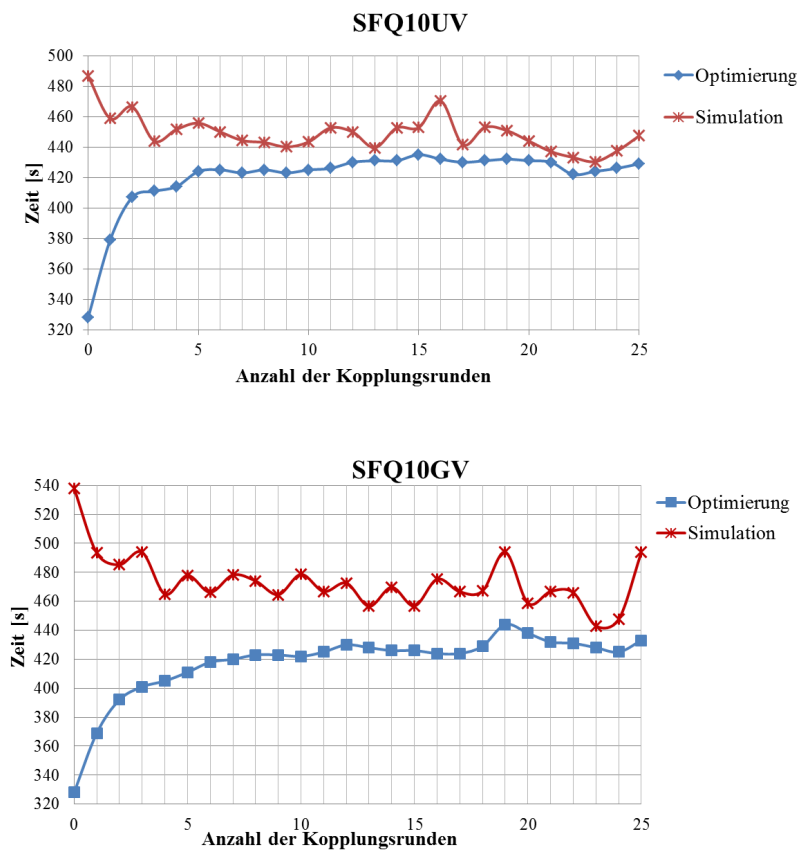


Abbildung 5.11: Ergebnisse der Kopplung des in Abbildung 5.7 dargestellten Szenarios: Vergleich von uniformer mit gaußverteilter Geschwindigkeit

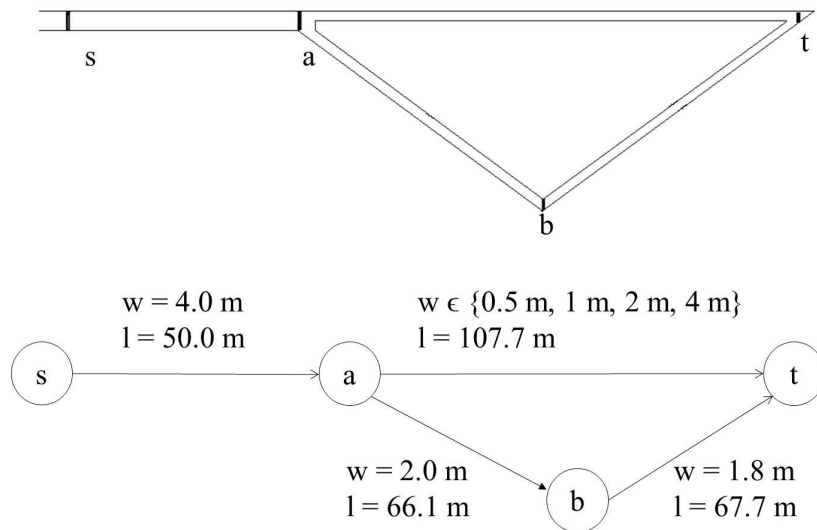


Abbildung 5.12: Szenario und abgeleitetes Netzwerk: Der Korridor *a-t* wird mit unterschiedlicher Breite modelliert

reproduzieren, da aufgrund der unterschiedlichen Verteilungsraten Staus an Zwischenzielen auftreten können.

Zusammenfassend lässt sich sagen, dass die Kopplung mit gaußverteilten Geschwindigkeiten zwar die Lücke zwischen Simulation und Optimierung verringert, die Ergebniskurven allerdings nicht so glatt sind wie im Fall mit konstanten Geschwindigkeiten. Trotzdem sind individuelle Geschwindigkeiten realistischer als uniforme Geschwindigkeiten; die Oszillationen sowie die größere Lücke zwischen den Zeiten sind daher akzeptierbar.

Für alle weiteren Tests wird die Konfiguration *SQF10GV* gewählt, um realistische Ausgangswerte zu erhalten.

### Testfall 2: Einfluss von unterschiedlichen Wegbreiten

Um die Auswirkungen unterschiedlicher Wegbreiten zu untersuchen, wird das obige Testzenario leicht verändert. Statt der Engstelle bei *b* wird die Engstelle bereits bei Knoten *a* variiert. Der kürzeste Weg wird mit unterschiedlichen Wegbreiten getestet, um den Einfluss der Optimierung auf die Simulation zu testen. Das Szenario mit dem erstellten Netzwerk ist in Abbildung 5.12 dargestellt. Die in Abbildung 5.13 dargestellten Ergebnisse zeigen, dass für alle vier unterschiedlichen Breiten des kürzesten Wegs der Abstand der jeweiligen Evakuierungszeiten geringer wird. Die Differenz beträgt 90 Sekunden für den Fall mit Korridorbreite von 0.5 *m* und 30 Sekunden für den Fall mit einer Korridorbreite von 4 *m*. Dies ist dadurch erklärt, dass für den zweiten Fall die Kapazität für den kürzeren Weg so groß ist, dass kein Stau mehr an Knoten *a* auftritt und die Evakuierungszeit deshalb nicht wesentlich beeinflusst wird. Im ersten Fall hingegen entsteht ein Stau vor *a*, der dazu führt, dass Fußgänger den längeren Weg zugewiesen bekommen und damit auch die Gesamtevakuierungszeit erhöhen. Wenn die Werte der

nullten Runde betrachtet werden, so ist erkennbar, dass die Simulationszeiten umso mehr verringert werden, desto schmaler der kürzeste Weg ist. Dies ist zum Einen die Folge davon, dass ein Stau vor der Engstelle auftritt, und zum Anderen bremsen langsamere Fußgänger die voraus gehen, schnellere ab, da schnellere Fußgänger in dem  $0.5\text{ m}$  breiten Gang nicht überholen können.

Insgesamt ist zu beobachten, dass die Simulationszeit in den Fällen  $w = 0.5\text{ m}$ ,  $1\text{ m}$  und  $2\text{ m}$  verbessert wird, nicht jedoch für  $w = 4\text{ m}$ . Dort gibt es keine Engstelle und es entsteht kein Stau. Folglich sind nur die mikroskopischen Effekte die Ursache dafür, dass die Zeit der Simulation über der Zeit der Optimierung liegt. Zudem trägt die Optimierung in diesem Fall nicht dazu bei, die Evakuierungszeit zu verringern. Dieses Ergebnis ist nicht überraschend, da ohne Engstellen jeder Fußgänger den kürzesten und damit schnellsten Weg gehen kann, und keine Optimierung der Routenwahl vorgenommen werden muss.

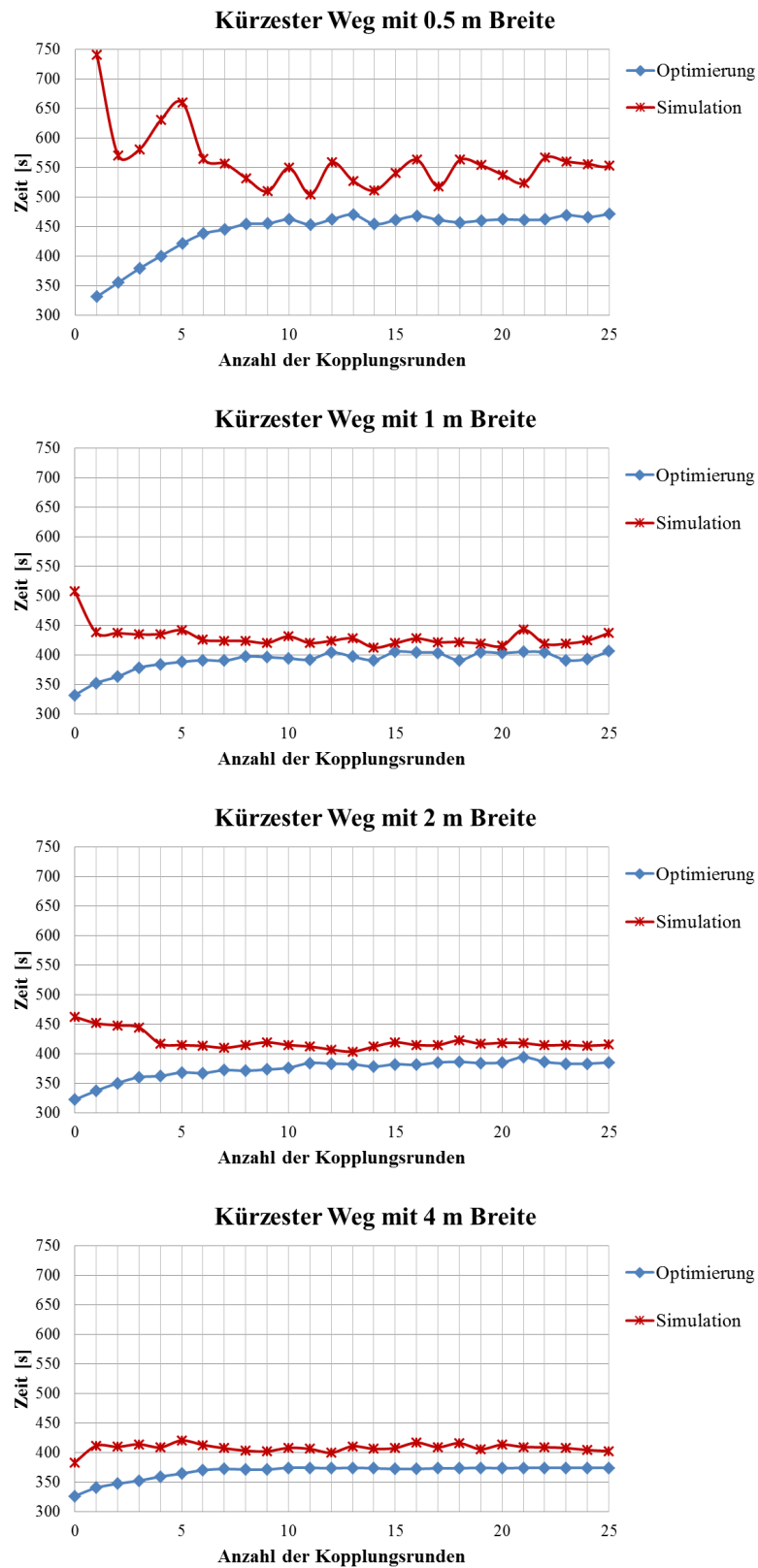


Abbildung 5.13: Ergebnisse der Kopplung des in Abbildung 5.12 dargestellten Szenarios

### Testfall 3: n-Wege Kreuzungen

Der nächste Testfall dient dazu, den Einfluss von  $n$ -Wege Kreuzungen zu testen. Das Szenario besteht hier aus drei unterschiedlichen Wegen von der Quelle zum Ziel. Der kürzeste Weg hat die geringste Kapazität; die beiden anderen Wege sind symmetrisch und eineinhalb mal so lang wie der kürzeste Weg. Die Summe der Wegbreiten aller Wege entspricht der Breite des Korridors vor der Kreuzung. Eine Abbildung des Szenarios sowie des daraus abgeleiteten Netzwerks ist in Abbildung 5.14 dargestellt.

In Abbildung 5.15 sind die Ergebnisse zu sehen. Die von der Simulation und Optimierung ermittelten Zeiten nähern sich in den ersten acht Kopplungsrunden an und umschließen einen konstanten Abstand von ca. 20 Sekunden für die übrigen Runden. Die Oszillationen können durch die auftretenden Interaktionen vor der Kreuzung bei  $a$  erklärt werden. Da einige Fußgänger am oberen Rand des Korridors auf den unteren Korridor verteilt werden und umgekehrt, kann es zu Stauungen kommen, die dadurch entstehen, dass sich die Fußgänger gegenseitig kreuzen.

Dennoch zeigen die Ergebnisse einen typischen Regelkreis mit kleinen Oszillationen. Einmal mehr ist zu beobachten, dass die Optimierung Verteilungsraten liefert, die die Evakuierungszeiten der Simulation reduzieren und die Optimierung auf der anderen Seite realistischere untere Schranken liefert.

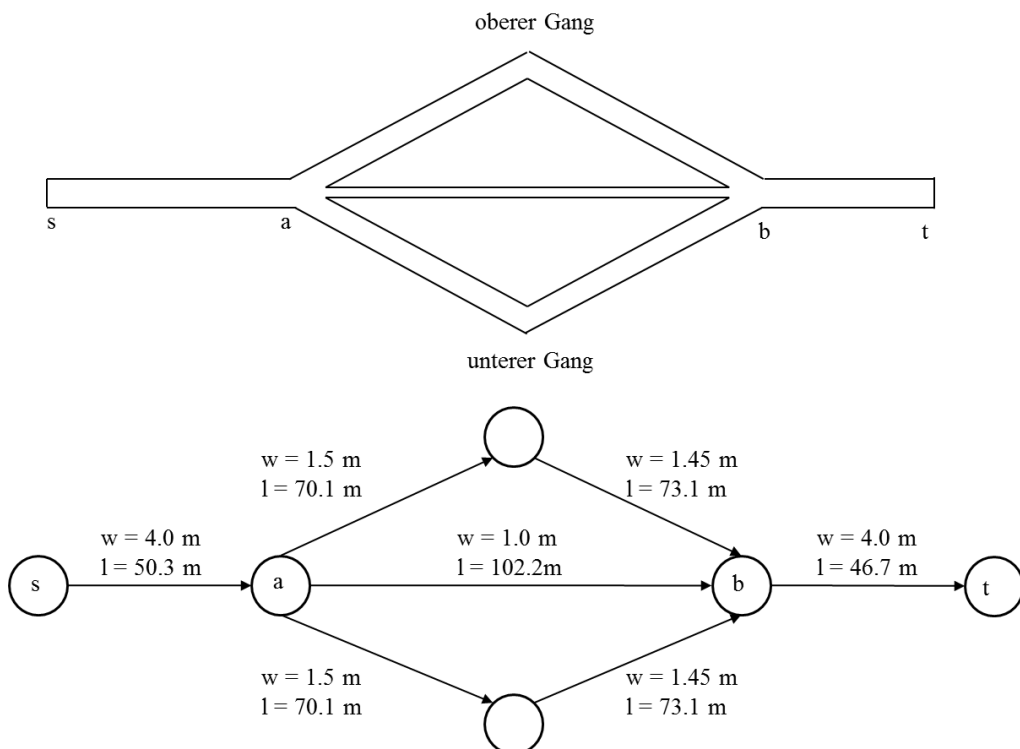


Abbildung 5.14: 3-Wege Szenario mit zugehörigem Netzwerk



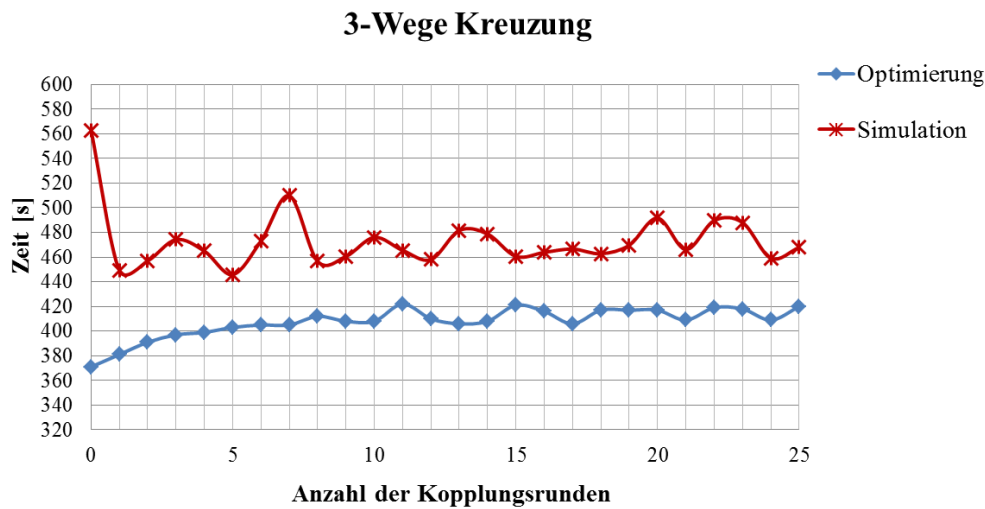


Abbildung 5.15: Berechnete Evakuierungszeiten mit gekoppelter Simulation und Optimierung

### Bewertung

Zusammenfassend lässt sich sagen, dass die Kopplung ein vielversprechendes Verfahren ist, das zumindest bei kleinen Szenarien gut anwendbar ist. Sicherlich verstärken sich die Oszillationen bei komplexeren Szenarien, bedingt durch die Sensitivität der Durchschnittswerte an Kreuzungen, die zwischen den beiden Komponenten ausgetauscht werden. Dennoch ist deutlich zu sehen, dass die Optimierung die Simulation „positiv“ beeinflusst: Kürzere Wege sind weniger verstopft, was eine insgesamt schnellere Evakuierungszeit zur Folge hat. Dieses makroskopische Allwissen besitzt in der Realität z.B. Sicherheitspersonal bei Großveranstaltungen. Sicherheitspersonal kann an Wegkreuzungen positioniert werden, um den Fußgängerstrom zu leiten.

Auch für die Optimierungskomponente zeigt sich, dass die Untergrenzen realistischer werden, indem mikroskopische Aspekte mit einfließen und unterschiedliche Geschwindigkeitswerte beachtet werden.

Wie aus den Ergebnissen der bisherigen Tests festzustellen ist, ist gerade in den ersten Runden eine erhebliche Annäherung der beiden Verfahren ersichtlich. Von dieser Erkenntnis wird Gebrauch gemacht, um einen etwas abgeänderten Kopplungsaufbau zu verwenden: Statt mehreren Iterationen werden die Verteilungen der Optimierung nur einmalig in der Simulation verwendet. Dieser Ansatz wird im Folgenden mit *minimaler Kopplung* bezeichnet. Er bietet sich vor allem für komplexere Szenarien an, in denen das Netzwerk sehr groß wird. Durch die vielen Iterationen im ursprünglichen Verfahren erhöht sich die Laufzeit der Optimierungskomponente enorm. Zudem können sich Oszillationen an einzelnen Kreuzungen potenzieren, so dass ein stationärer Zustand nicht erreicht werden kann. Im Folgenden wird die minimale Kopplung anhand eines Realbeispiels getestet.

### 5.5.2 Anwendungsbeispiel: Minimale Kopplung für komplexe Szenarien<sup>2</sup>

Um auch für komplexe Szenarien diesen Ansatz zu verwenden, ohne das Potenzieren von Oszillationen zu erzeugen, wird hier die minimale Kopplung eingeführt: Statt mehrere Kopplungsrunden durchzuführen, besteht der Kopplungszyklus nur aus einer Runde. Der Aufbau der Kopplung ist exakt gleich, nur die Iterationen werden weggelassen. Der Ablauf der minimalen Kopplung ist in Abbildung 5.16 dargestellt. Aus einem Szenario wird ein Graph abgeleitet, der zu einem dynamischen Netzwerk erweitert wird. Die Optimierung liefert zeitabhängige Verteilungsraten  $DRs$  für jeden Knoten des Netzwerks zurück, die als Eingabeparameter an die Simulation übergeben werden. Die Simulation wird mit gegebenem Graph und Szenario durchgeführt und erzeugt als Ergebnis eine verbesserte Evakuierungszeit gegenüber der reinen Simulation ohne Input aus der Optimierung.

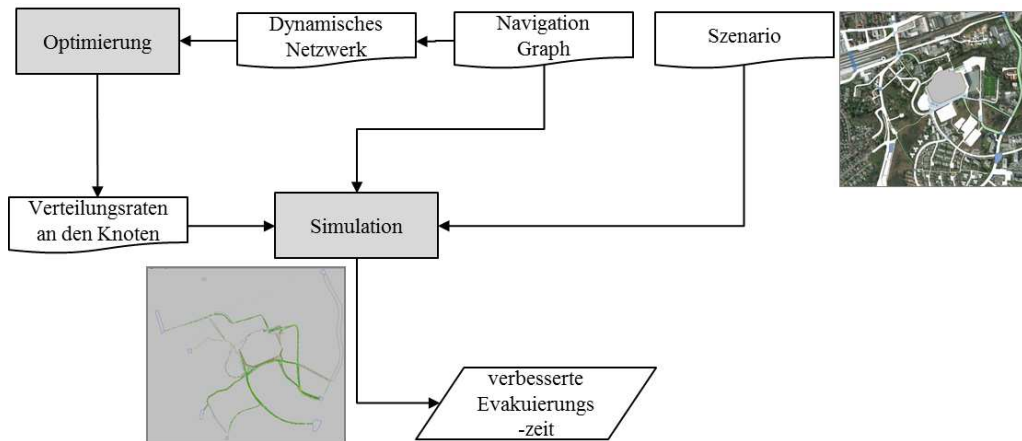


Abbildung 5.16: Ablauf der minimalen Kopplung

Als Anwendungsbeispiel dient ein Fußballstadion in Deutschland. Hier sollen 40.000 Fußballfans nach einem Spiel von den Toren des Stadions möglichst schnell zu unterschiedlichen Zielen evakuiert werden. In Abbildung 5.17a ist ein Luftbild<sup>3</sup> der Umgebung dargestellt.

Das Szenario wird automatisiert aus dem Luftbild abgeleitet: Alle Wege werden weiß eingefärbt; damit kann jede Zelle als betretbar modelliert werden, deren Mittelpunkt weiß ist. Alle anderen Farben werden als nicht betretbar – also Hindernisse – modelliert.

Die Quellen sind um das Stadion platziert. Die Ziele entsprechen den in der Umgebung liegenden Parkplätzen sowie dem Bahnhof, der sich in der nordöstlichen Ecke des Ausschnitts befindet (Abbildung 5.17b). Aus diesem Szenario wird der Graph für die Optimierung bzw. die Simulation abgeleitet (Abbildung 5.17c). In

<sup>2</sup>Dieser Ansatz wurde in Kneidl *et al.* (2011) bereits vorgestellt.

<sup>3</sup>maps.google.com

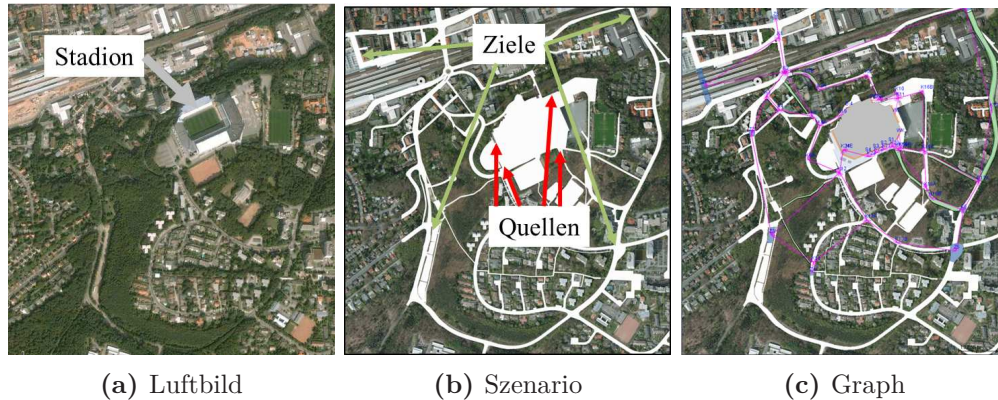


Abbildung 5.17: Szenario für die Kopplung: in a) ist ein Luftbild aus *google maps* zu sehen; in b) ist das abgeleitete Szenario dargestellt; c) zeigt den gemeinsamen Graph für Optimierung und Simulation

Tabelle 5.5 sind die verwendeten Parameter aufgelistet. Da der Kopplungszyklus nur aus einer Runde besteht, wird kein *SFQ* Feedback benötigt. Begonnen wird

Parameter	Wert
Anzahl der Fußgänger	40.000
$\alpha$	0.3
Mittlere Geschwindigkeit zu Beginn der Kopplung	1.34 m/s
SFR	Im Vorverarbeitungsschritt berechnet

Tabelle 5.5: Parameter für das Anwendungsbeispiel Fussballstadion

mit einem Optimierungslauf, der die Verteilungsraten  $DR$  ermittelt. Jedem Ziel sind von jeder Quelle aus eine bestimmte Anzahl von Fußgängern zugewiesen. Diese Verteilungsraten dienen der Simulation als Eingangsparameter, die Fußgänger werden anhand dieser Raten auf die einzelnen Wege verteilt.

Die Ergebnisse der Simulation ohne Kopplung und der Simulation mit Kopplung sind in Tabelle 5.6 dargestellt: In der stand-alone Version ist zu erkennen, dass bereits nach 10 Minuten große Staus an der südwestlichen sowie an der nordöstlichen Ecke des Stadions entstehen. Da keine expliziten Verteilungsraten vorgegeben werden, laufen alle Fußgänger entlang der schnellsten Wege. Blickt man hingegen auf die Ergebnisse mit der Kopplung, so ist zu erkennen, dass keine Staus auftreten. Zudem wird die Evakuierungszeit gegenüber der Evakuierungszeit ohne Kopplung mit dem Netzwerkflussmodell um 20 % verbessert.

Sicherheitskräfte können von diesen Ergebnissen profitieren und diese zur Planung und Durchführung von Großveranstaltungen verwenden.









Zeit	Simulation ohne Optimierungsinput	Simulation mit Optimierungsinput
Nach 10 Minuten		
Nach 15 Minuten		
Nach 20 Minuten		
Nach 25 Minuten		

Tabelle 5.6: Ergebnisse der Simulation, einmal mit und einmal ohne Optimierungsinput

## 5.6 Zusammenfassung

In diesem Kapitel wurde ein Kopplungsverfahren vorgestellt, das ein mikroskopisches mit einem makroskopischen Simulationsmodell (Netzwerkflussmodell) kombiniert. Dieses Verfahren kann dazu eingesetzt werden, eine realistischere untere Schranke von Evakuierungszeiten, die in der Netzwerkoptimierung berechnet werden, unter Miteinbeziehung von mikroskopischen Effekten zu bestimmen. Da in makroskopischen Simulationsmodellen einzelne Individuen nicht betrachtet werden und damit auch keine Interaktionen, sind die daraus resultierenden Evakuierungszeiten zu optimistisch. Mit dem Kopplungsverfahren werden diese unteren Schranken angehoben und damit die Aussagekraft der Optimierung verbessert.

In mikroskopische Modelle hingegen fließt mit diesem Verfahren globales Wissen mit ein. Damit ist es möglich, z.B. das Wissen von Sicherheitspersonal zu modellieren, das die Personen entlang optimaler Routen schickt, optimal in Bezug auf die Gesamtevakuierungszeit. Auch hier verbessert das Kopplungsverfahren die Simulationsergebnisse, indem die Gesamtevakuierungszeit abgesenkt wird.

In den schematischen Testfällen wurde das Funktionieren des Verfahrens nachgewiesen. Dabei wurde festgestellt, dass die Wahl der Zeitschrittlänge die Konvergenz wesentlich beeinflusst; werden die Zeitschritte zu klein gewählt, kann es passieren, dass die Werte nicht genügend geglättet werden und so größere Oszillationen in den Ergebnissen verursachen. In Kombination mit dem Quellflussmengen-Feedback kann eine ungeschickte Wahl sogar dazu führen, dass die von der Optimierung ermittelten Evakuierungszeiten oberhalb der von der Simulation berechneten Zeiten liegen. Daher ist bei der Anwendung des Verfahrens darauf zu achten, diese Parameter entsprechend zu justieren.

Komplexere Szenarien können ebenfalls größere Oszillationen versuchen. Dies liegt darin begründet, dass sich durch die Komplexität des zugrundeliegenden Netzwerks bei hintereinander geschalteten Kreuzungen Oszillationen potenzieren können, da bei jeder Kreuzung Unruhe in die Durchschnittswerte kommt. Da in den schematischen Testfällen gerade in der ersten Kopplungsrunde eine erhebliche Annäherung der Ergebnisse von Optimierung und Simulation zu beobachten ist, wird dem durch ein leicht abgeändertes Verfahren entgegengewirkt, der minimalen Kopplung. Diese wurde anhand eines Anwendungsbeispiel vorgestellt und getestet. Die Ergebnisse zeigen, dass auch die minimale Kopplung bereits zu einer Annäherung beider Komponenten führt.

Daraus kann geschlossen werden, dass eine solche Kopplung Einsatz in der Planung und Durchführung von Großveranstaltungen finden kann.



## Kapitel 6

# Wegefindung: Routingalgorithmen zur Abbildung menschlichen Navigationsverhaltens

Wie in Kapitel 3 bereits erwähnt, ist der globale Wegefindungsprozess von Fußgängern ein wichtiger Aspekt, der bei der Modellierung von Fußgängern auf der taktischen Ebene abgebildet wird (Hoogendoorn & Bovy, 2004, Reynolds, 1999). Dieses Kapitel befasst sich mit der Modellierung unterschiedlicher Typen von Fußgängern zur Abbildung des Wegefindungsverhaltens. Dabei werden neue Algorithmen eingeführt, die kognitive Aspekte bei der Wegefindung in Algorithmik überführen.

### 6.1 Kognitive Aspekte bei der Wegefindung von Menschen

Der Prozess der Wegefindung wird von Wissenschaftlern aus unterschiedlichen Disziplinen untersucht. Hier wird vor allem auf Arbeiten von Soziologen, Psychologen, Anthropologen, Informatikern sowie Geographen näher eingegangen, um einen umfassenden Überblick über den aktuellen Stand der Wissenschaft zu geben.

#### **Definition und Aspekte der Wegefindung**

Um den Begriff Wegefindung klar abzugrenzen, wird zunächst eine Definition nach Golledge (1999a, S. 6) angegeben:

**Definition 6.1** *Wegefindung (engl. wayfinding) ist der Prozess, einen Weg oder eine Route zwischen einem Start und einem Ziel zu finden und zu verfolgen. Es ist eine beabsichtigte, gerichtete und motivierte Aktivität.*

Historisch betrachtet stellt der Begriff Wegefindung eine Erweiterung des Begriffs Navigation dar, der die Steuerung von Schiffen und Flugzeugen beschrieb. Daher

wird im Folgenden der Begriff *Wegefindung* verwendet. Zu beachten ist, dass das Ziel beim *Wegefindungsprozess* nicht im direkten Wahrnehmungsbereich des Menschen liegt, sondern in sogenannten „large-scale spaces“ (Downs & Stea, 1977, Kuipers, 1983). Diese Räume können nicht von einem einzigen Blickwinkel aus wahrgenommen werden, sondern müssen durch *Traversieren* erkundet werden. Als „large-scale spaces“ werden Landschaften, Städte und Gebäude bezeichnet.

*Wegefindung* besteht aus psychologischen und kognitiven Aspekten, die in jedem Menschen unterschiedliche Ausprägungen annehmen. Psychologische Aspekte basieren auf Emotionen wie Angst, Wohlfühlen, Kommunikation mit anderen, Drang nach Neuem, Erfahrung usw. (Gluck, 1991).

Kognitive Prozesse werden im Kontext von *Wegefindung* als

- Lesen und Navigieren in einer Karte,
- zu einem bekannten Ort navigieren,
- Richtungsanweisungen zu geben oder zu empfangen und umzusetzen sowie
- einen Pfad wiederholt zu verfolgen

beschrieben.

Allen (1999) hat eine Taxonomie zu *Wegefindungsaufgaben* aufgestellt, die bis heute so verwendet wird. Er unterscheidet drei unterschiedliche Aufgaben: (1) Zu einem bekannten Ziel steuern (2) Exploratives Wandern mit dem Ziel, wieder zum Ausgangspunkt zurückzukehren, und (3) Sich in Richtung eines unbekanntes Ziels zu bewegen.

In dieser Arbeit werden die Kategorien (1) und (3) näher betrachtet, da im Fokus dieser Arbeit das Abbilden des Navigationsverhaltens von Personen mit klaren Zielen liegt.

### **Aspekte des menschlichen *Wegefindungsprozesses***

Es gilt als wissenschaftlich erwiesen, dass Menschen zur *Wegefindung* eine kognitive Repräsentation des zu erkundenden Gebiets erstellen und darauf zurückgreifen. Man spricht dabei von „kognitiven Karten“. Der Begriff wurde erstmals von Tolman (1948) eingeführt. Er fand bei einem Experiment mit Ratten heraus, dass sich diese bei der *Wegefindung* durch ein Labyrinth Wissen über räumliche Relationen zwischen Start und Ziel erworben hatten.

Ein weiterer Versuch wurde 1960 von Lynch durchgeführt (Lynch, 1960), der Menschen von drei verschiedenen Städten in den USA eine Karte der Innenstädte ihrer Stadt zeichnen ließ. Ziel war es, etwas über die Struktur der Städte zu lernen. Als „Nebenprodukt“ kam heraus, dass Menschen dazu tendierten, Orte, an denen sie sich gerne aufhalten, genauer zu zeichnen als Orte, die unbeliebter sind. Die Karten hatten zudem Löcher oder Verzerrungen. Diese kognitiven Karten enthielten metrische Fehler (Entfernungen), jedoch keine topologischen (relationalen) Fehler.

Wie bei Lynch erstmals herausgefunden und später von Golledge zusammengefasst wurde, besteht eine kognitive Karte aus Punkten, Linien, Regionen und Oberflächen (Golledge, 1999a). Punkte können hier beispielsweise Landmarks repräsentieren, Linien stehen für Routen oder Wege, in Regionen werden u.a.



Stadtviertel abgebildet und Oberflächen stellen Steigungen oder generelle Formen dar. Eine komplette Liste von Bestandteilen der kognitiven Karten ist in Golledge (1999b) gegeben.

Landmarks sind ein wichtiges Werkzeug, das beim Prozess der Wegfindung eingesetzt wird. Untersuchungen zeigten, dass Menschen Landmarks, also prägnante und bekannte Orte verwenden, um sich zu orientieren. Landmarks können in urbanen Umgebungen bekannte Plätze oder persönliche Orte wie das Zuhause darstellen. Innerhalb von Gebäuden werden Schilder und Wegweiser als Landmarks bezeichnet.

In einem Versuch von Golledge und Spector 1974 wurden 400 Teilnehmer gebeten, Landmarks in Columbus, Ohio (USA) zu benennen. Circa die Hälfte aller Landmarks waren dabei personenbezogene Landmarks. Die andere Hälfte der Landmarks waren allgemein bekannte Orte, die auch von anderen Personen als solche bezeichnet wurden.

### **Forschungsansätze zur Beschreibung von Wegfindungsprozessen**

Die wissenschaftlichen Arbeiten zur Wegfindung können laut Gluck (1991, S. 118) in zwei überlappende Kategorien unterteilt werden: Kompetenz (engl. *competence*) und Performanz (engl. *performance*).

Die Literatur der Kategorie Kompetenz wendet kognitive Modelle bei der Beschreibung von Wegfindung an. Das oben beschriebene Experiment von Lynch kann dieser Kategorie zugeordnet werden.

In der Literatur zur Performanz wird die Fähigkeit der Menschen betrachtet, ihren Weg zu finden. Es wird – ähnlich zu psycholinguistischen Aspekten beim Erlernen einer Sprache – die Genauigkeit bei der Erstellung einer kognitiven Karte und der Vergleich mit der Realität, wie die richtige Einschätzung von Entfernung und Winkeln, untersucht. Dazu wurden unterschiedliche Experimente durchgeführt. Die Ergebnisse zeigten, dass Menschen metrische Informationen weniger präzise einschätzen können als topologische. Als Beispiel seien die Ergebnisse eines von Golledge durchgeführten Versuchs genannt, die belegen, dass die zugrunde liegende euklidische Metrik sehr verzerrt wahrgenommen werden kann. Es konnten sehr unterschiedliche Arten von Verzerrungen derselben Karte, wie Fischaugeneffekt oder Spaghettidarstellung, nachgewiesen werden (Golledge, 1999a). Daher ist auch die Wegfindung für jedes Individuum unterschiedlich und resultiert in verschiedenen Ergebnissen.

Gluck (1991, S. 129) kritisiert, dass der Fokus der wissenschaftlichen Arbeiten auf dem Verstehen des Erzeugens einer kognitiven Karte liegt und fordert, sich mehr auf den Bedarf der Informationsbeschaffung zu konzentrieren. Er argumentiert damit, dass das Erzeugen einer kognitiven Karte oftmals mehr Zeit in Anspruch nimmt, als der Person in diesem Moment zur Verfügung steht, in dem sie in einer fremden Umgebung nach einem Weg sucht. Die Person konzentriert sich also mehr auf den „sense-making“-Prozess. Diesen Prozess beschreibt er als kreativen menschlichen Prozess, in dem der Mensch die Welt zu einem bestimmten Zeitpunkt an einem bestimmten Ort zu verstehen versucht. Dieser Prozess des Verstehens ist durch die menschlichen physiologische Kapazitäten, die Gegenwart, Vergangenheit und Zukunft begrenzt. Gluck legt seinen Forschungsschwerpunkt

daher mehr auf den Wegefindungsprozess an sich als auf die Darstellung des Umfelds, in der die Wegefindung stattfindet.

Gluck (1991, S. 118) beschreibt den Wegefindungsprozess als einen inkrementellen Prozess. Zunächst werden wichtige Plätze bzw. Landmarks wahrgenommen, diese werden dann durch Routen verknüpft, daraus entstehen kognitive Karten, die schließlich auch Routen zwischen beliebigen Punkten enthalten. Dabei nehmen die Menschen ihre Umgebung individuell wahr und konstruieren ein individuelles Bild ihrer Umgebung.

Ein weiterer interessanter Aspekt ist der Unterschied des Wegefindungsprozesses bezüglich Geschlecht, nach kulturellen Unterschieden, Kinder oder ältere Leute usw. Im Rahmen dieser Arbeit wird von Europäern ausgegangen.

In der Studie von Lawton & Kallai (2002) wurde herausgefunden, dass es Unterschiede in der Wegefindung zwischen Frauen und Männern gibt: Frauen haben weniger räumliches Selbstbewusstsein als Männer. Räumliches Selbstbewusstsein bezieht sich darauf, dass Männer sich darauf verlassen, auch die nicht-sichtbare Umgebung korrekt einzuschätzen und somit globale Konfigurationsstrategien verwenden. Frauen dagegen orientieren sich eher an lokalen Landmarks, um den Weg zu finden.

### **Charakteristika von Wegen**

Es wird die These vertreten, dass sich Menschen in urbanen Umgebungen so geradlinig wie möglich in Richtung ihres Ziels bewegen. In Dalton (2001) wird ein Experiment beschrieben, das dazu dient, diese These zu stützen, die teilweise auch durch Ergebnisse von Golledge (1995) und Conroy (2001) untermauert wird. Das Experiment wurde mit 30 Probanden durchgeführt, die virtuell durch ein städtisches Gebiet navigieren sollten. Dabei gab es an jeder Kreuzung des Testgebiets zwischen drei und zehn ausgehende Wege. Die Winkel bei Richtungswechseln variierten zwischen 150 Grad und 180 Grad (geradeaus). Die Ergebnisse zeigen, dass Personen dazu neigen, so geradlinig wie möglich in Richtung ihres Ziels zu gehen.

Dies geht Hand in Hand mit den Untersuchungen zur Linearisierung der Wege. Damit ist gemeint, dass Personen Routen mit weniger Richtungswechseln grundsätzlich als kürzer einschätzen, als Routen mit vielen Richtungswechseln (Tolman, 1948, Sadalla & Montello, 1989, Montello, 1991). Hillier stellt ebenfalls die Hypothese auf, dass Personen den Weg mit der längsten Sichtlinie wählen, die ihre Richtung annähert.

Golledge (1995) führte Experimente durch, um das Navigationsverhalten von Fußgängern in urbanen Umgebungen zu untersuchen. Die naheliegendste Annahme zur Charakterisierung von Wegen ist, dass Menschen den kürzesten oder schnellsten Weg wählen. Dies überprüfte Golledge in seinem Experiment. Dazu legte er 32 Testpersonen Karten vor, in denen unterschiedliche Routen innerhalb eines stadtähnlichen Gebiets eingezeichnet waren. Die Routen waren alle exakt gleich lang, hatten aber unterschiedliche geometrische Eigenschaften (möglichst nahe an der Luftlinie zum Ziel, kurze Strecken zuerst, lange Strecken zuerst). Die Testpersonen sollten angeben, welche Route sie bevorzugen würden, falls sie diese Route täglich gehen würden. In einer zweiten und dritten Versuchsreihe wurde

die einfache Geometrie komplexer gemacht. Es wurden Diagonalen eingefügt und schließlich wurde die Geometrie um gekrümmte Straßen sowie attraktive Plätze (Parks) und unattraktive Plätze (z.B. Müllhalden) erweitert. Zusätzlich zum Einzeichnen der Routen wurden die Personen danach befragt, anhand welcher Kriterien sie dachten, die Wegewahl im Experiment durchgeführt zu haben. Die Ergebnisse der Befragung stimmten nicht mit den gewählten Routen überein. Des Weiteren wurde in dieser Versuchsreihe festgestellt, dass menschliche Wegfindung nicht einem bestimmten Muster entspricht, sondern sehr vielschichtig ist und nach unterschiedlichen Kriterien erfolgt.

Golledge geht sogar so weit, dass er Wegfindung als einen zweckgebundenen Prozess beschreibt. Damit meint er, dass täglich gelaufene Wege, wie z.B. der tägliche Weg zur Arbeit nach anderen Kriterien gewählt werden als Wege, die in der Freizeit gelaufen werden. Während täglich gelaufene Wege möglichst effizient in Bezug auf Strecke und Zeit sind, können Wege in der Freizeit besonders schöne Routen beinhalten. Eine Tabelle mit unterschiedlichen Kriterien ist in Golledge (1999a, S. 31) angegeben.

Inwiefern Landmarks oder Straßenbezeichnungen bei Wegbeschreibungen unterschiedlichen Einfluss haben, wurde in einem Experiment von Tom & Denis (2003) untersucht. Dabei wurde festgestellt, dass der Einsatz von Landmarks bei der Wegbeschreibung die Wegfindung signifikant besser unterstützt als Straßennamen.

Dieser Einblick in die wissenschaftlichen Arbeiten unterschiedlicher Disziplinen soll die hier entwickelten Algorithmen zur Wegfindung motivieren und dafür sensibilisieren, dass der Prozess der Wegfindung ein sehr komplexer ist. Entsprechend können die Algorithmen nur approximieren, wie Menschen ihren Weg finden.

## 6.2 Algorithmen zur Abbildung unterschiedlicher Strategien bei der Wegfindung

Ausgehend von den in Abschnitt 6.1 beschriebenen Aspekten werden verschiedene Algorithmen vorgestellt, die den Wegfindungsprozess und die daraus resultierenden Wege möglichst realistisch abbilden sollen. Hierbei ist zu erwähnen, dass die Algorithmik die Kognition der Menschen nur annähern kann, da menschliches Verhalten durch einen eigenen Willen und individuelles Handeln geprägt ist und nicht vollständig mit computergestützten Algorithmen abgebildet werden kann. Bei der Modellierung wird zwischen Normalsituation und Evakuierungsfall unterschieden. Als Normalsituation werden Situationen bezeichnet, in denen keine Gefahr besteht. Personen wenden in diesen Situationen Wegfindungsprozesse zum Navigieren im Alltag an.

Als Evakuierung wird in diesem Zusammenhang eine geordnete Evakuierung bezeichnet, in der keine unmittelbare Gefahr besteht. Stattdessen wird davon ausgegangen, dass keine lebensbedrohliche Situation für die Personen besteht, sondern eine mittelbare Gefahr, wie beispielsweise eine Bombenentschärfung in der näheren Umgebung oder Austritt von Gasen. Es wird hier auf die Modellierung

von Nofalleevakuierung (also Evakuierung aufgrund einer unmittelbaren Gefahr) verzichtet, da das Verhalten in Paniksituationen noch wenig erforscht ist und so eine Modellierung nur sehr ungenau sein kann.

In beiden grundlegenden Situationen wird jeweils zwischen ortskundigen und ortsfremden Personen unterschieden.

Die Algorithmen, die den Personenkreis abbilden, der sich ortskundig in einer Normalsituation bewegen, sollen zum Einen das Verhalten abbilden, dass Personen möglichst den kürzesten Weg zum Ziel verfolgen. Zum Anderen werden Personen modelliert, die sich möglichst nahe an der Luftlinie zum Ziel orientieren.

Ortsfremde Personen werden in mehrere Kategorien unterteilt. Personen mit einem guten Orientierungssinn werden ebenfalls mit der Strategie abgebildet, möglichst nahe entlang der Luftlinie zu laufen, allerdings ohne globales Wissen über die Umgebung. Der Personenkreis mit durchschnittlichem Orientierungssinn wird durch das Verhalten modelliert, dass der gewählte Weg möglichst wenige Richtungswechsel enthält. Personen ohne Orientierungssinn können mit einem probabilistischem Algorithmus abgebildet werden, der das Verhalten abbildet, anderen Personen zu folgen.

Für den Evakuierungsfall wird ebenfalls zwischen ortskundigen und ortsfremden Personen unterschieden. Ortskundige Personen orientieren sich an dem nächstgelegenen Ziel und versuchen dieses so schnell wie möglich zu erreichen. Ortsfremde Personen dagegen orientieren sich an anderen Personen und folgen diesen.

Ein Überblick über die modellierten Fußgängertypen und die zugehörigen Algorithmen ist in Tabelle 6.1 und Tabelle 6.2 gegeben.

Der Einsatz von Landmarks in der Wegfindung wird ebenfalls betrachtet. In Abschnitt 6.3 wird ein Konzept zur Integration von bekannten Orten vorgestellt. Dieses Konzept ist mit allen Fußgängertypen kombinierbar.

Zur realitätsnahen Abbildung von menschlichem Wegfindungsverhalten müssen diese unterschiedlichen Algorithmen je nach Demographie und Grundsachverhalts eines zu simulierenden Szenarios miteinander kombiniert werden.

Alle in den nächsten Abschnitten beschriebenen Algorithmen verwenden als Grundlage den in Kapitel 4 vorgestellten Graph. Durch unterschiedliche Metriken bei der Definition von Kanten- und Knotengewichten werden die beschriebenen Verhaltensmuster abgebildet. Im Folgenden werden die einzelnen Algorithmen detailliert erklärt und die Ergebnisse anhand eines einfachen schematischen Testfalls veranschaulicht, der in Abbildung 6.1 dargestellt ist.

Statt, wie in Abschnitt 3.1 beschrieben, der Zuweisung eines Ziels, auf das die Fußgänger zusteuern, wird den Fußgängern als Resultat der Algorithmen eine Liste von Zwischenzielen mitgegeben, die sie auf ihrem Weg zu ihrem eigentlichen Ziel passieren müssen.

### 6.2.1 Strategie „Schnellster Weg“: Fastest Path Algorithmus

Die Strategie des schnellsten Wegs soll den Personenkreis abbilden, der mit einer Umgebung sehr gut vertraut ist und Alternativrouten kennt, falls Staus auf dem kürzesten Weg auftreten. Dies ist der einfachste Modellierungsfall, da hier

Normalsituation			
ortskundig		ortsfremd	
Verhalten	Algorithmus	Verhalten	Algorithmus
schnellster Weg	Fastest Path	wenig Abweichung von der Zielrichtung	Greedy Heuristic Path
wenig Abweichung von der Zielrichtung	Heuristic Path	Lange Strecken geradeaus	SALL
		anderen Personen folgen	Probabilistic Choice

Tabelle 6.1: Überblick über die implementierten Algorithmen und Verhaltensmuster in Normalsituationen

auf den Dijkstra-Algorithmus (Dijkstra, 1959) zurückgegriffen werden kann, der in einem Graph den optimalen Weg von einem Startknoten zu einem Zielknoten findet, optimal in Bezug auf die Metrik des Kantengewichts. Die Metrik wird hier als Reisezeit definiert. Der Algorithmus ist in die Klasse der Greedy-Algorithmen einzuordnen (Cormen, 2010). Greedy-Algorithmen sind dadurch definiert, dass sie ausgehend von einer Ausgangssituation den nächsten Zustand wählen, der in Bezug auf den vorliegenden Zustand den größten Gewinn bringt. Die Entscheidung fällt dabei lokal, um global ein optimales Ergebnis zu erzielen.

Der Dijkstra-Algorithmus sucht – ausgehend vom Startknoten – immer die Kante mit dem geringsten Kantengewicht aus. Falls es einen Weg mit geringeren Gewicht zwischen dem Startknoten und dem aktuellen Knoten gibt, wird das aktuell gespeicherte Gewicht durch das geringere Gewicht ersetzt und der Vorgänger wird auf den Anfangsknoten der eingehenden Kante gesetzt. Sofern ein Weg zwischen Startknoten und Endknoten existiert, wird der Weg mit dem geringsten Gewicht gefunden. Eine ausführliche formale Definition ist in Dijkstra (1959) zu finden.

Die Kantengewichte für den hier vorliegenden Fall werden als Reisezeiten angegeben. Eine Möglichkeit ist es, Reisezeiten zu verwenden, die als Durchschnittswerte aus der Geschwindigkeit auf der Kante sowie dem euklidischen Abstand (es herrscht Sichtverbindung) berechnet werden. Da jede Person ihre aktuelle Geschwindigkeit kennt, kann diese direkt verwendet werden. Als Geschwindigkeit kann der Kante aber auch die minimale, maximale oder die Durchschnittsgeschwindigkeit der Personen, die sich auf der Kante befinden, zugewiesen werden. Angenommen, die minimale Geschwindigkeit würde gewählt, und es wäre weiter die Situation gegeben, die in Abbildung 6.2 dargestellt ist: Dann besäßen die Kanten  $e_2$  und  $e_3$  eine hohe Reisezeit, da die Personen kurz vor der Engstelle sehr langsam sind. Wenn man aber die Länge der Kanten betrachtet und die Zeit, wie

<b>Evakuierung</b>			
ortskundig		ortsfremd	
Verhalten	Algorithmus	Verhalten	Algorithmus
zum nächstgelegenen Ziel laufen	Fastest Path zum nächsten Ziel	anderen Personen folgen	Probabilistic Choice

Tabelle 6.2: Überblick über die implementierten Algorithmen und Verhaltensmuster im Evakuierungsfall

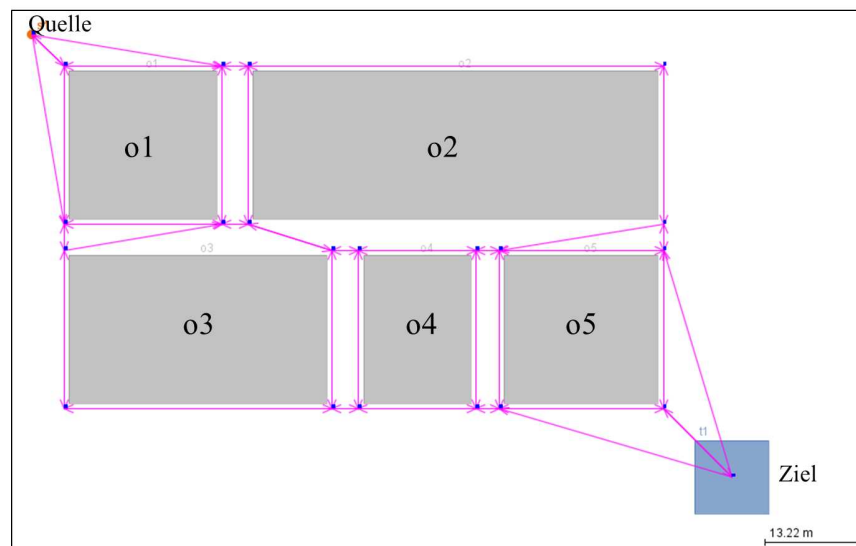


Abbildung 6.1: Beispielszenario zur Veranschaulichung der unterschiedlichen Algorithmen, bestehend aus einer Quelle und einem Ziel sowie verschiedenen Hindernissen ( $o1-o5$ ), die die Sichtverbindung zum Ziel blockieren.

lange der Fußgänger braucht, um an der Engstelle anzukommen, hätte sich der Stau schon aufgelöst und die Kante  $e_2$  wäre die schnellste. Verwendet man umgekehrt die Maximalgeschwindigkeit auf einer Kante, so würde immer die Kante gewählt, auf der sich Personen noch frei bewegen könnten, auch wenn am Ende der Kante ein Stau auftritt. Dies ist besonders bei kurzen Kanten ungünstig, da die Reisezeit hier sehr gering ist, und zu dem Zeitpunkt, an dem der Fußgänger am Stau angekommen ist, dieser noch nicht aufgelöst ist. Bei Durchschnittswerten bekommt man zwar eine bessere Annäherung, bezieht aber die Ankunftszeit ebenfalls nicht mit ein.

Diese Einschränkungen lassen sich umgehen, wenn die Reisezeiten aus den Navigationsfeldern ermittelt werden, die in Unterabschnitt 3.2.2 vorgestellt wurden. Diese Navigationsfelder berechnen die Ankunftszeiten einer propagierten Welle unter Miteinbeziehung aller Fußgänger und dienen als Grundlage für die Naviga-

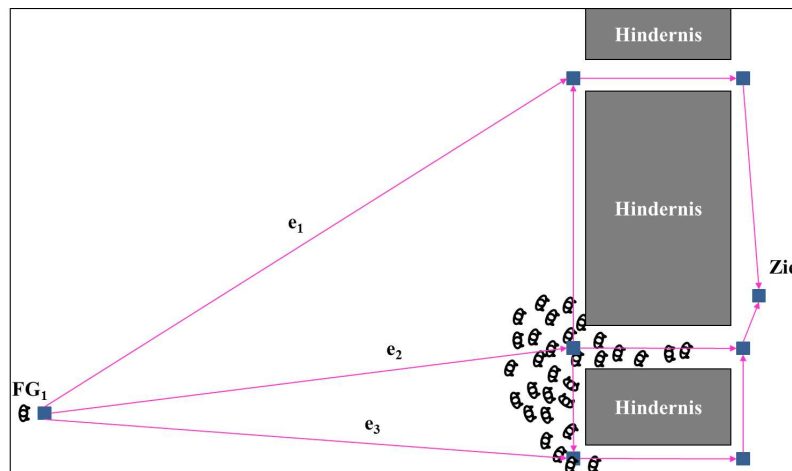


Abbildung 6.2: Beispiel für Reisezeiten, ermittelt aus minimalen Geschwindigkeiten: Die Kante  $e_2$  würde nicht gewählt aufgrund der geringen Geschwindigkeit vor der Engstelle – jedoch wäre der Stau längst aufgelöst, wenn der Fußgänger  $FG_1$  an der Engstelle ankommt. Die minimale Geschwindigkeit ist somit kein gutes Maß zur Abschätzung der Reisezeit.

tion auf dem zellulären Automaten. Statt ein globales Navigationsfeld zu generieren, werden viele lokale Navigationsfelder berechnet: Für jede Kante des Graphen wird ein solches Feld generiert, ausgehend vom Zielknoten in Richtung des Startknotens. Daher kann der Wert, den das Feld am Startknoten annimmt, direkt als Reisezeit auf der Kante und somit als Kantengewicht interpretiert werden. Diese Reisezeit liefert eine Abschätzung unter Miteinbeziehung der Entfernung und der Personen, die sich auf der Kanten befinden. Dies nähert die menschliche Wahrnehmung besser an als direkt aus Geschwindigkeiten abgeleitete Reisezeiten. Ein Beispiel für den schematischen Aufbau der kantenbezogenen Navigationsfelder ist in Abbildung 6.3 gegeben. Die Unterteilung des globalen Navigationsfeldes in

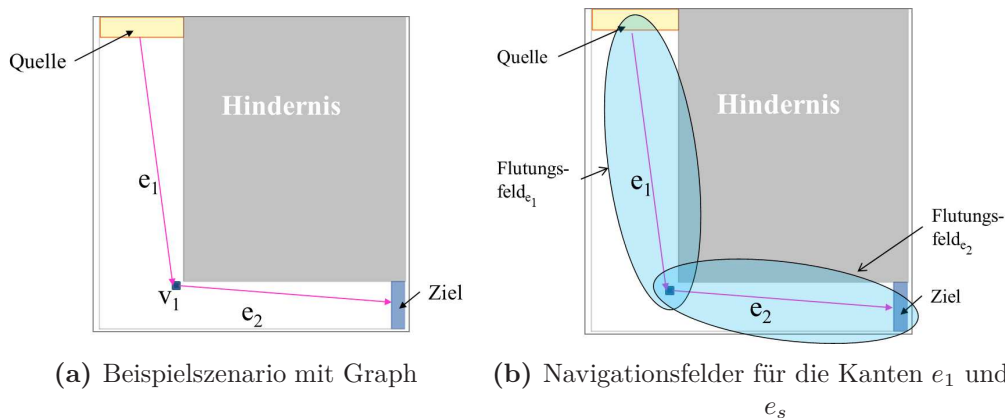


Abbildung 6.3: a) Beispielszenario, bestehend aus einer Quelle und einem Ziel sowie zwei Kanten  $e_1$  und  $e_2$ ; b) Schematische Darstellung der Navigationsfelder für die beiden Kanten  $e_1$  und  $e_2$

viele kleine kantenbezogene Navigationsfelder bringt gleich mehrere Vorteile mit sich: Es müssen nur kleine Teile des Szenarios aktualisiert werden, nämlich nur diejenigen Felder, auf denen sich Personen bewegen. Zudem kann eine gerichtete Flutung entlang der Kante erfolgen. Dies ist anders als im Ansatz ohne Navigationsgraph: Dort muss sich die Welle in alle Richtungen ausbreiten, da das Ziel nicht vom Startpunkt aus sichtbar ist und damit die Richtung des Ziels nicht bekannt ist.

Da hier Sichtverbindung zwischen Quelle und Ziel herrscht, ist die Richtung des Ziels bekannt und kann direkt verwendet werden. Mit der gerichteten Flutung werden Zellen, die geflutet werden, nicht nach ihrem Flutungswert (der Ankunftszeit der Welle) sondern nach ihrem Abstand zum Ziel sortiert. Die Idee ist folgende: Sei  $V^O$  und  $V^D$  der Startknoten und Endknoten einer Kante; dann können, der Idee von Petres *et al.* (2007) folgend, die Knoten des Gitters  $\vec{x}_i$  (entspricht den Zellmittelpunkten des zellulären Automaten) im FMM-Algorithmus<sup>1</sup> (Sethian, 1999) nicht basierend auf dem Schlüssel  $\kappa(\vec{x}_i) = T(\vec{x}_i)$ , sondern als eine Kombination aus kürzester Zeit  $T(\vec{x}_i)$  und Abstand zum Startknoten  $d(\vec{x}_i, V^O)$  sortiert werden:

$$\kappa(\vec{x}_i) = \alpha T(\vec{x}_i) + (1 - \alpha)\beta d(\vec{x}_i, V^O),$$

wobei  $\alpha$  und  $\beta$  geeignete Konstanten darstellen für die gilt:  $0 \leq \alpha \leq 1$  und  $\beta > 0$ .

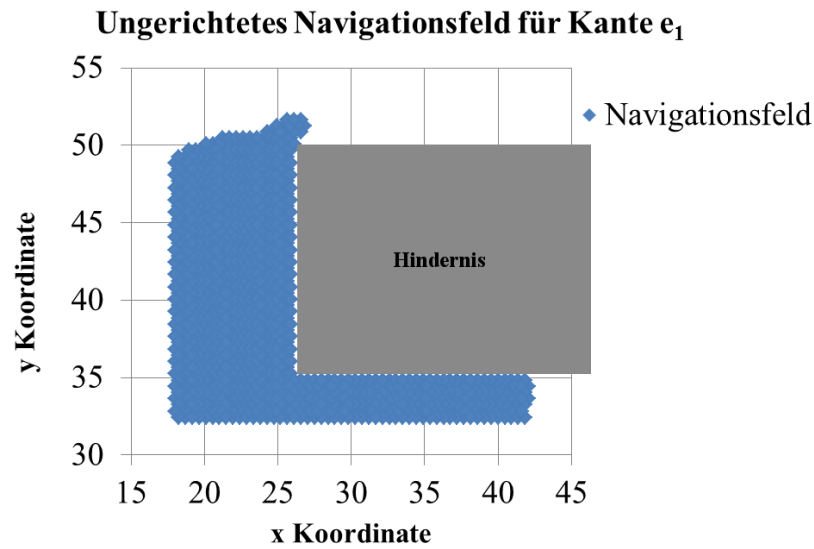
Dies erzwingt ein bevorzugtes Aktualisieren der Zellen, die in Richtung des Startknotens liegen, also eine mehr oder weniger strikt gerichtete Berechnung (abhängig von der Wahl der Konstanten  $\alpha$  und  $\beta$ ) des Feldes der zugehörigen Kante  $e_{OD}$ . Ein Vergleich zwischen einem ungerichteten und einem gerichteten Navigationsfeld mit unterschiedlichen Werten für  $\alpha$  ist für die Kante  $e_1$  aus dem Beispiel in Abbildung 6.3a in Abbildung 6.4 gegeben.

Das ungerichtete Navigationsfeld (Abbildung 6.4a) breitet sich in alle Richtungen aus, was in einem sehr großen Feld resultiert, bei dem viele Zellen geometrisch weit entfernt von der eigentlichen Kante  $e_1$  liegen. Die Form des gerichteten Navigationsfeldes (Abbildung 6.4b) ist sehr stark abhängig von der Wahl für  $\alpha$ .  $\alpha$  muss daher geeignet gewählt werden, um eine angemessene Zellabdeckung auf dem Kantengebiet zu gewährleisten.

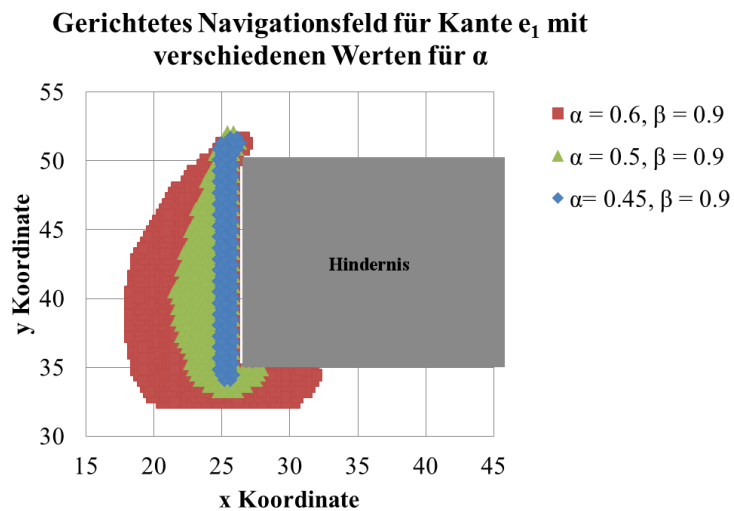
---

<sup>1</sup>FMM (Fast Marching Method): Die Methode ist in Unterabschnitt 3.2.2 beschrieben





(a) ungerichtetes Navigationsfeld



(b) gerichtetes Navigationsfeld

*Abbildung 6.4:* a) Darstellung eines ungerichteten Navigationsfeldes für die Kante  $e_1$  aus dem Beispiel in Abbildung 6.3a, verglichen mit gerichteten Navigationsfeldern b) für unterschiedliche Werte für  $\alpha$ . Die Welle breitet sich von Knoten  $v_1$  in Richtung Quelle aus. Das ungerichtete Feld wird dabei sehr groß und enthält viele Zellen, die geometrisch weit entfernt zur Kante  $e_1$  liegen.

Die Ergebnisse einer Simulation unter Anwendung des Algorithmus sind in Abbildung 6.5 dargestellt. Es werden insgesamt 1000 Personen simuliert, in der Quelle werden vier Personen pro Sekunde erzeugt. Die Simulationszeit beträgt 300 Sekunden. Man erkennt, dass es aufgrund der Symmetrie des Szenarios zwei kürzeste Wege gibt. Der eine verläuft nördlich des Hindernisses  $o1$ , der zweite verläuft westlich des Hindernisses  $o1$ . Die Fußgänger wählen beide Wege und verteilen sich gleichmäßig auf beide Varianten, so dass kein Stau auftritt (Abbildung 6.5a). Da der weitere Verlauf des kürzesten Wegs nach der Umgehung des Hindernisses  $o1$  wieder zusammenläuft, ergibt sich ein höhere Dichte im Durchgang zwischen  $o3$  und  $o4$ , so dass der Weg östlich des Hindernisses  $o4$  schneller wird (Abbildung 6.5b). Die Pfade aller Fußgänger sind in (Abbildung 6.5c) dargestellt. Die meisten Fußgänger laufen entlang des kürzesten Wegs, der zum Großteil der Zeit auch dem schnellsten Weg entspricht. Aufgrund höherer Dichten zwischen den Hindernissen, die sich nach einiger Zeit bilden, weisen jedoch die Wege um die Hindernisse herum kürzere Reisezeiten auf.

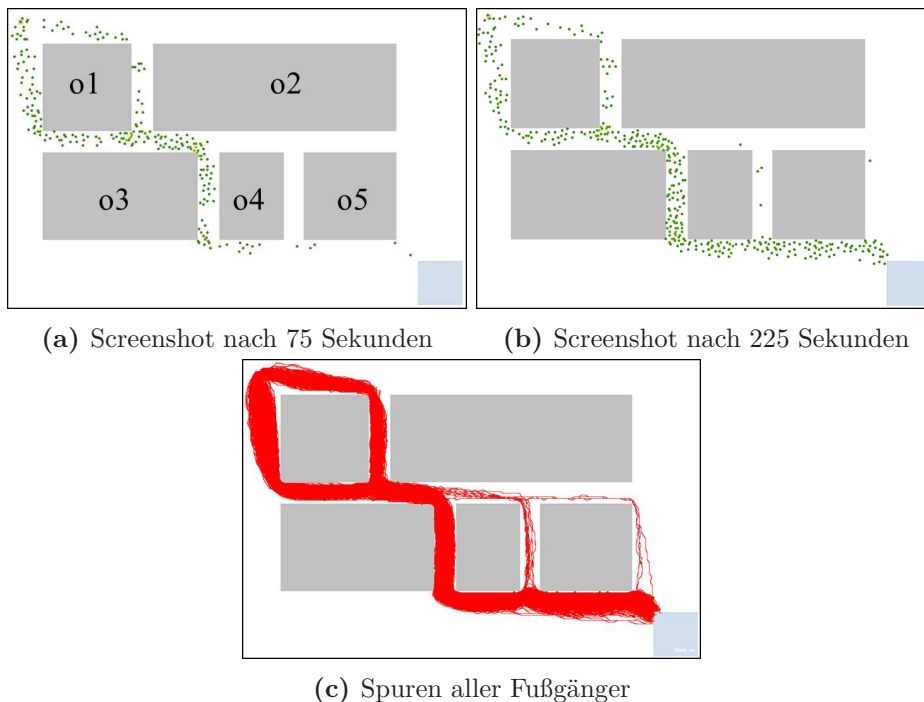


Abbildung 6.5: Simulationsergebnisse für den Algorithmus *Fastest Path*: Es ist zu erkennen, dass aufgrund der Symmetrie des Szenarios zwei kürzeste Wege existieren. Der eine verläuft nördlich des Hindernisses  $o1$ , der zweite verläuft westlich des Hindernisses  $o1$ . Die Fußgänger wählen beide Wege und verteilen sich gleichmäßig auf beide Varianten, so dass kein Stau auftritt.

### 6.2.1.1 Reduzierung des Wissens

Mit dem so beschriebenen Ansatz würde ein Stau vor einer Engstelle von Fußgängern schon miteinbezogen, bevor er überhaupt sichtbar ist. Dies ist in der Realität nicht der Fall. Ein Fußgänger kann nicht durch Hindernisse hindurchsehen und damit auch keinen Stau sehen, der hinter einer Obstruktion liegt. Ein Beispiel für eine solche Situation ist in Abbildung 6.6 angegeben. Hier hat sich ein Stau vor einer Engstelle geformt, der von dem Ausgangspunkt des Fußgängers *FG* nicht sichtbar ist. Wenn nun die dynamischen Navigationsfelder zur Berechnung des schnellsten Wegs verwendet werden, würde der Fußgänger direkt südlich des Hindernisses entlang laufen, da dies der schnellste Weg ist.

Um dieses unnatürliche Verhalten zu vermeiden, müssen Kanten, die nicht sichtbar sind, anders gewichtet werden. Für jeden Fußgänger wird daher ein Set von sichtbaren Kanten vorab gespeichert. Dies kann in der Vorverarbeitungsphase geschehen, indem für jeden Knoten ein Set von sichtbaren Kanten ermittelt wird. Für alle Kanten außerhalb des Sichtbereichs wird der euklidische Abstand als Kantengewicht verwendet. Dadurch kann die Allwissenheit reduziert und realistischeres Laufverhalten erzeugt werden. Um die Unterschiede in den Simulati-

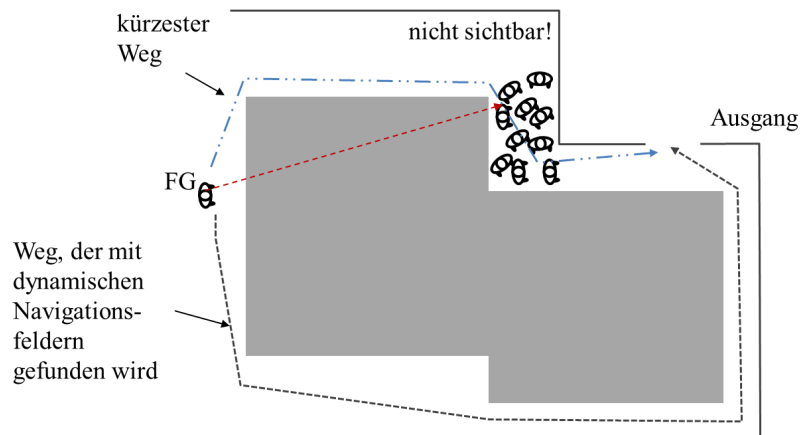


Abbildung 6.6: Aufgrund der dynamischen Navigationsfelder würde der Stau erkannt, der sich im nicht sichtbaren Bereich des Fußgängers befindet. Dies kann verhindert werden, indem nicht-sichtbare Kanten euklidische Abstände als Gewicht zugewiesen bekommen.

onsergebnissen zu verdeutlichen, wird folgendes Beispiel herangezogen (vgl. Abbildung 6.7a): Es besteht aus mehreren Hindernissen, die den Blick von der Quelle auf das Ziel versperren. Hinter dem Hindernis *o1* befindet sich eine Engstelle in Richtung Ziel. Im Simulationslauf werden sechs Fußgänger pro Sekunde erzeugt, die Simulationszeit beträgt 600 Sekunden.

Unterscheidet man nicht zwischen sichtbaren und nicht-sichtbaren Kanten, so laufen zunächst alle Fußgänger nördlich um das Hindernis *o1*, und nach einiger Zeit bildet sich an der Engstelle ein Stau. Wenn dieser Stau groß genug ist, wird die globale Reisezeit auf dem Weg über die südliche Seite von Hindernis *o1* geringer. Deshalb beginnen die Fußgänger, südlich des Hindernis entlang zu laufen,

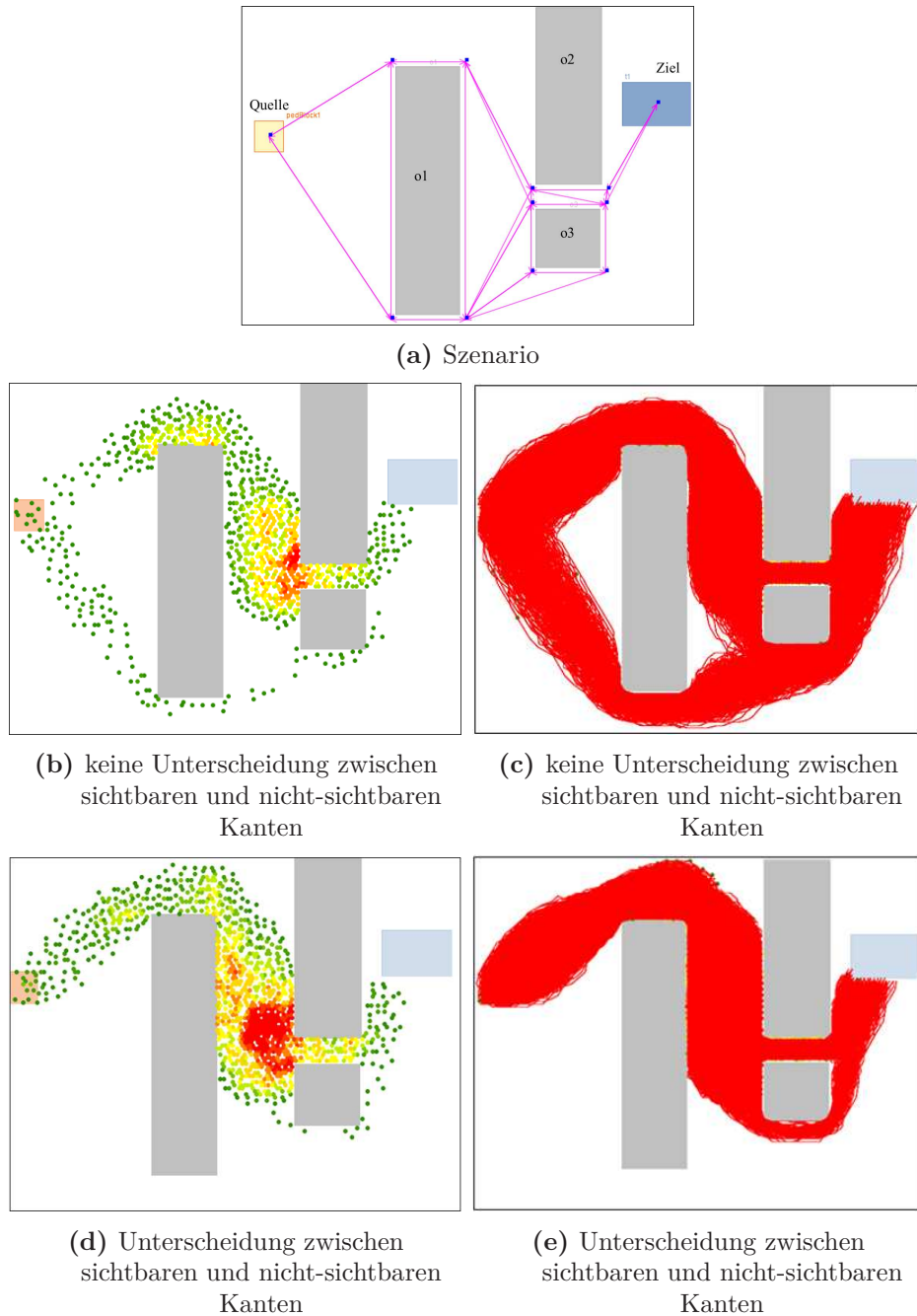


Abbildung 6.7: Beispielszenario zur Demonstration des unterschiedlichen Navigationsverhaltens, abhängig von der Unterscheidung zwischen sichtbaren und nicht-sichtbaren Kanten

obwohl sie den Stau an der Engstelle nicht wahrnehmen können.

Wenn nun zwischen sichtbaren und nicht-sichtbaren Kanten unterschieden wird, so passiert Folgendes: Zunächst bildet sich wieder ein Stau an der Engstelle. Da aber im sichtbaren Bereich der Weg nördlich des Hindernisses immer noch schneller ist, als um die südliche Ecke zu laufen, laufen Fußgänger weiterhin nördlich

um das Hindernis herum. Erst nachdem sie um die Ecke gelaufen sind, nehmen sie den Stau wahr und versuchen, über das südliche Ende von  $o\beta$  auszuweichen. Dies entspricht eher der Realität, da Personen in der Regel auf nicht sichtbare Staus nicht reagieren können.

Ein Screenshot nach 400 Sekunden sowie die Pfade aller Fußgänger für die Simulation, in der alle Kantengewichte aus dynamischen Navigationsfeldern ermittelt werden, sind in Abbildung 6.7b und Abbildung 6.7c dargestellt. Die Ergebnisse der Simulation mit Unterscheidung zwischen sichtbaren und nicht-sichtbaren Kanten, sind in Abbildung 6.7d und Abbildung 6.7e veranschaulicht.

### 6.2.1.2 Abbildung der Fehleinschätzung euklidischer Metrik

Da Menschen die euklidische Metrik nicht korrekt einschätzen können, wird im Folgenden eine entsprechende Anpassung des Algorithmus diskutiert. Statt des schnellsten Wegs werden stattdessen die  $k$  schnellsten Wege berechnet und aus den gefundenen Wegen zufällig ein Weg gewählt.  $k$  kann dabei beliebig groß gewählt werden. Zu beachten ist dabei, dass eine obere Grenze für die Abweichung in der Gesamtlänge angegeben wird. Sonst kann es passieren, dass in einem sehr kleinen Szenario die  $k$  besten Wege sehr große Unterschiede in der Länge aufweisen und damit nicht alle Wege der Charakteristik schnellster Weg entsprechen.

Die Ergebnisse der Simulation mit den 10 schnellsten Wegen, die eine geringere Abweichung als 10 % vom schnellsten Weg haben, sind in Abbildung 6.8 dargestellt. Die Pfade der Fußgänger verteilen sich von Anfang an auf drei unterschiedliche Wege. Nach der Umgehung des nördlichsten Hindernisses prägen sich drei Routen um die südlichen Hindernisse aus. Alle drei Wege haben eine sehr ähnliche Länge und werden daher auch vom Algorithmus gefunden. Dies bildet die Realität besser ab, da das menschliche Auge kaum unterscheiden kann, welcher der drei Wege der kürzeste ist.

### 6.2.2 Strategie „Möglichst nahe entlang der Luftlinie zum Ziel“: Heuristic Path Algorithmus

Der zweite Typ Fußgänger, der implementiert wurde, stützt sich auf die Thesen von Dalton (2001) und Conroy (2001), die in Experimenten ermittelt haben, dass sich Menschen möglichst geradlinig ihrem Ziel annähern. Um dies algorithmisch abzubilden, wird ein heuristisches Verfahren eingesetzt, nämlich der A\*-Algorithmus.

Dieser Algorithmus wurde erstmals von Hart *et al.* (1968) eingeführt, und dient der Wegsuche in Graphen. Er wird häufig in der Robotik oder in Computerspielen eingesetzt, um kürzeste Pfade zu berechnen. Der Unterschied zum Dijkstra-Algorithmus liegt darin, dass zusätzlich zum bekannten Kantengewicht eine Heuristik verwendet wird, um schneller eine Lösung zu finden. Die Voraussetzung für den Einsatz der Heuristik ist, dass Informationen über den zu erreichenden Zielknoten bekannt sind. Ist zum Beispiel bekannt, in welcher Richtung das Ziel liegt, kann die Luftlinie zum Ziel als Heuristik verwendet werden. Ein Beispiel dafür, inwiefern diese Heuristik die Suche effizienter gestalten kann, ist in Abbildung 6.9

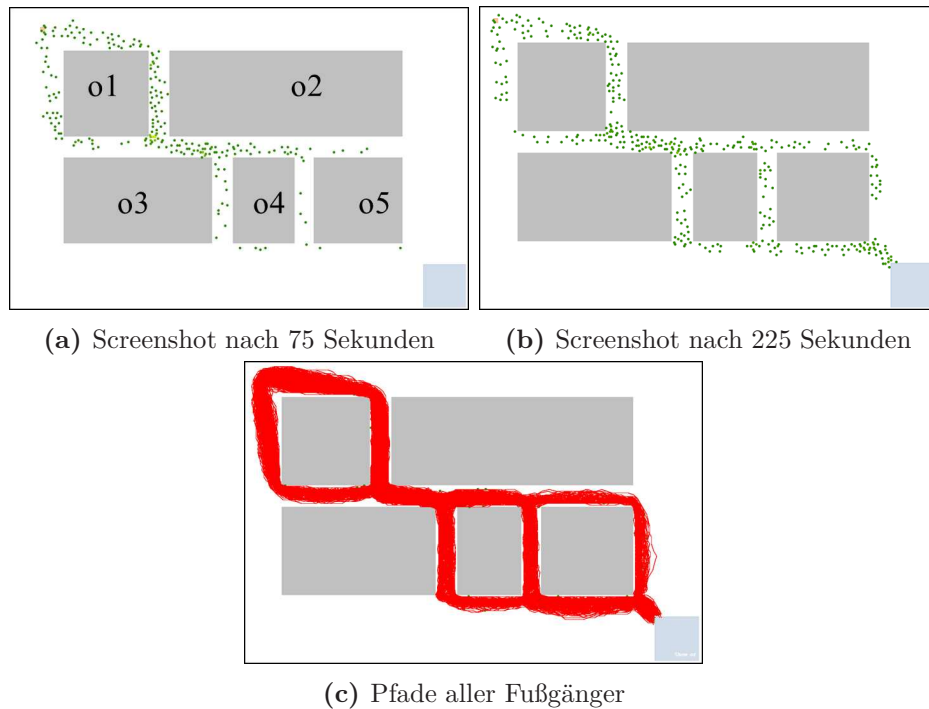


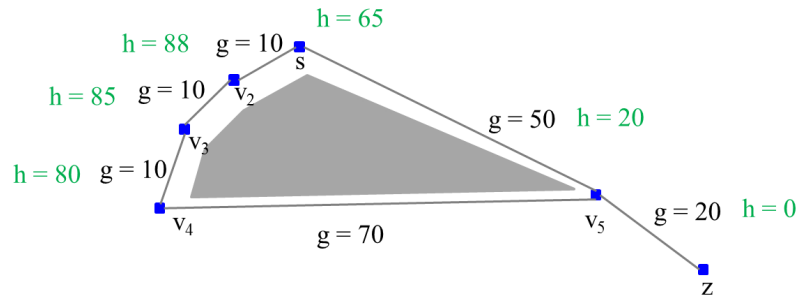
Abbildung 6.8: Simulationsergebnisse für den Algorithmus *Fastest Path* unter Miteinbeziehung der  $k$  kürzesten Wege: Die Pfade der Fußgänger verteilen sich von Anfang an auf drei unterschiedliche Wege. Nach der Umgehung des nördlichsten Hindernisses prägen sich drei Routen um die südlichen Hindernisse aus. Alle drei Wege haben eine sehr ähnliche Länge und werden daher auch vom Algorithmus gefunden.

gegeben. Hier soll ein Weg von Knoten  $s$  zu Knoten  $z$  gefunden werden. Unter Zuhilfenahme der Luftlinie als Heuristik kann der A\*-Algorithmus den schnellsten Weg sehr viel schneller finden als der Dijkstra-Algorithmus. Dies ist aus den Suchbäumen der beiden Algorithmen ersichtlich. Beim A\*-Algorithmus können ganze Teilbäume aus dem Suchbaum herausgeschnitten werden, was die Suche beschleunigt<sup>2</sup>. Voraussetzung für eine Heuristik ist, dass sie mindestens zulässig ist. Zulässig bedeutet, dass die realen Kosten nie überschätzt werden. Eine noch strengere Bedingung ist, dass eine Heuristik monoton ist. Monoton bedeutet, dass sichergestellt sein muss, dass die geschätzten Kosten eines Knotens kleiner oder gleich der realen Kosten plus die geschätzten Kosten des Nachfolgeknotens sind. Formal ausgedrückt bedeutet das, dass für den Knoten  $k$  und Nachfolgeknoten  $k'$  gelten muss:

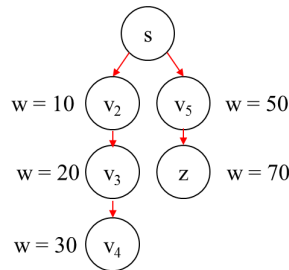
$$h(k) \leq g(k, k') + h(k'),$$

wobei  $g(k, k')$  die realen Kosten der Kante zu  $k'$  und  $h(k')$  die geschätzten Kosten des Nachfolgeknotens  $k'$  bezeichnen. Wird eine monotone Heuristik gewählt, so

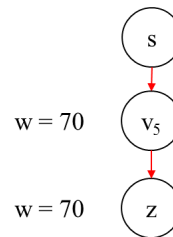
<sup>2</sup>In der Umsetzung des *Fastest Path* Algorithmus wurde der A\* Algorithmus eingesetzt. Zur deutlichen Unterscheidung des *Fastest Path* Algorithmus und des *Heuristic Path* Algorithmus wurde der *Fastest Path* als Dijkstra Algorithmus eingeführt.



(a) Graph, in dem ein Weg von  $s$  nach  $t$  gefunden werden soll,  $h$  ist die heuristische Komponente,  $g$  entspricht dem Kantengewicht



(b) Suchbaum des Dijkstra-Algorithmus



(c) Suchbaum des A\*-Algorithmus

Abbildung 6.9: Beispiel zur Demonstration des Suchverhaltens des Dijkstra-Algorithmus und des A\*-Algorithmus. Mithilfe einer Heuristik können ganze Teilbäume aus der Suche ausgeschlossen und somit schneller eine Lösung gefunden werden.

findet der A\* Algorithmus immer eine optimale Lösung. Der Beweis hierzu ist in Hart *et al.* (1968) zu finden.

Hier wird der A\*-Algorithmus in einer leicht abgewandelten Form eingesetzt, angelehnt an Höcker *et al.* (2010) und Höcker (2010): Die Heuristik soll das nicht-vollständige Wissen von Fußgängern abbilden. Ziel ist es nicht, einen optimalen Weg bezüglich der Gesamtlänge zu finden, sondern einen optimalen Weg bezüglich der Heuristik zu berechnen. Wie in Abschnitt 6.1 beschrieben, können die Menschen nur einen Teil der so genannten „large-scale spaces“ erfassen. Die nicht sichtbaren und nicht wahrgenommenen Regionen müssen abgeschätzt werden. Als Heuristik wird deshalb die Luftlinie zum Ziel gewählt. Der Algorithmus funktioniert wie folgt (Hart *et al.*, 1968):

Zusätzlich zu den Kantengewichten werden Knotengewichte eingeführt. Für jede traversierte Kante wird als Kantengewicht der euklidische Abstand gewählt. Das Knotengewicht setzt sich aus dem bisher zurückgelegten Weg (Summe aller traversierten Kantengewichte) und einem Schätzwert für den noch zurückzulegenden Weg, also der Luftlinie zum Ziel, zusammen. Formal ausgedrückt:

$$g_v = \sum_{e \in E_{trav}} g_e + \alpha \cdot h_v$$

wobei  $g_v$  dem Gewicht in Knoten  $v$  entspricht,  $g_e$  der Summe aller Gewichte der bisher traversierten Kanten und  $h_v$  dem euklidischen Abstand des Knotens zum Ziel (entspricht der Luftlinie).  $0 \leq \alpha \leq 1$  bezeichnet eine Gewichtungskonstante und gibt an, wie stark der heuristische Teil einfließt. Es ist zu beachten, dass hier keine optimalen Lösungen gefunden werden, sobald der heuristische Teil größer als 1 gewichtet wird. Denn dann wird die Monotoniebedingung verletzt. Dies ist aber gerade gewollt, da Fußgänger eben nicht global gesehen nach der besten Lösung suchen, sondern auf die Heuristik bezogen einen optimalen Pfad verfolgen sollen. Der Weg wird komplett zu Beginn berechnet und ist somit von Anfang bekannt ist. Ein Beispiel zur Verdeutlichung ist in Abbildung

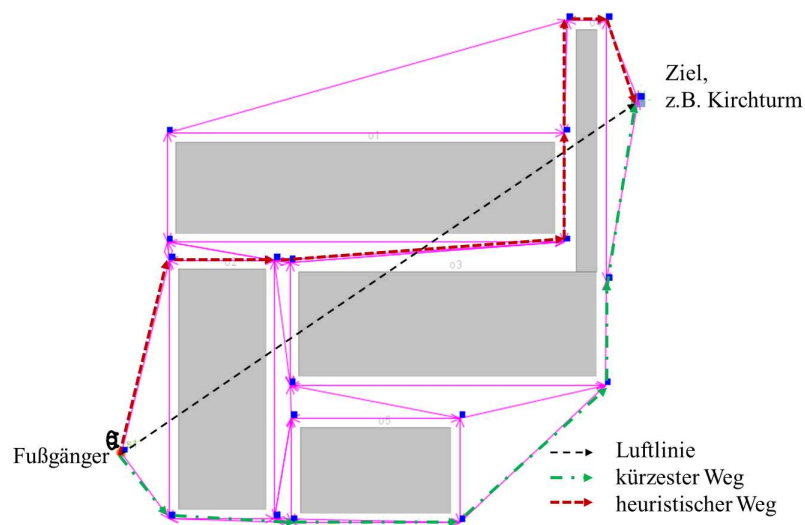


Abbildung 6.10: Beispielszenario zur Veranschaulichung des *Heuristic Path* Algorithmus: Während der kürzeste Weg südlich der Gebäude entlang führt, findet der *Heuristic Path* Algorithmus den Weg, der mit möglichst wenig Abweichung von der Luftlinie verläuft.

6.10 gegeben. Der schematische Aufbau könnte zum Beispiel für eine Situation stehen, in der ein Fußgänger sich in einer Stadt befindet und in Richtung eines weithin sichtbaren Ortes (z.B. Kirche mit Kirchturm) laufen möchte. Der Kirchturm ist dabei sichtbar, die Gebäude verdecken jedoch den Weg, der zu dieser Kirche führt. Bei Kenntnis des kürzesten Wegs würde man südlich der Gebäude entlang laufen, um schließlich nach Norden in Richtung Kirche abzubiegen. Orientiert man sich jedoch an der Luftlinie, führt der Weg zwischen den Gebäuden hindurch, bis die Kirche schließlich sichtbar wird. Um den Einfluss von  $\alpha$  näher zu untersuchen, wurde die Simulation mit Werten für  $\alpha = 0$  (entspricht dem Dijkstra-Algorithmus, da keine Heuristik mit einfließt und nur die Kantengewichte betrachtet werden) bis  $\alpha = 2.0$  durchgeführt.

Die Ergebnisse sind in Abbildung 6.11 dargestellt. Es ist zu erkennen, dass in diesem Szenario der Algorithmus den kürzesten Weg bis zu einem Wert von  $\alpha = 1.5$  findet. Generell gilt: Je höher der Wert für  $\alpha$ , desto näher führt der Weg entlang der Luftlinie.



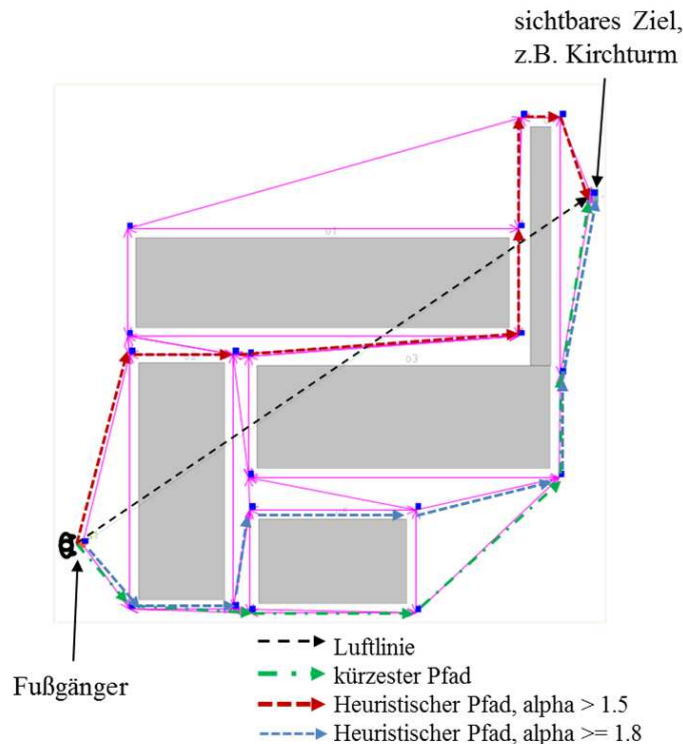


Abbildung 6.11: Ergebnisse der gefundenen Wege für unterschiedliche Werte für  $\alpha$ .

Angewandt auf das Beispielszenario aus Abbildung 6.1 sind die Ergebnisse der Simulation in Abbildung 6.12 dargestellt. Es ist zu beobachten, dass es genau einen Weg gibt, dem alle Fußgänger folgen. Dieser Weg verläuft erkennbar nahe an der Luftlinie. Da als Kantengewicht keine Reisezeiten herangezogen werden, tritt nach einer gewissen Zeit ein Stau zwischen den Hindernissen  $o1$  und  $o2$  auf.

### 6.2.2.1 Integration von Fehlern bei der Abschätzung der Richtung

Die bisherige Annahme für den Algorithmus ist, dass Menschen fehlerfrei die Richtung abschätzen können, in der das Ziel liegt. Untersuchungen haben ergeben, dass metrische Abschätzungen fehlerbehaftet sind. Es kann hier zwischen unterschiedlichen Fehlerquellen unterschieden werden (Hochmair & Frank, 2003): Es gibt (a) den Fehler bei der Abschätzung von Winkeln (engl. *rotation error*), (b) den Fehler in der Abschätzung von Strecken sowie (c) den Fehler beim Zeigen in die Richtung des Ziels (engl. *pointing error*). Zu allen Fehlern sind in der Literatur verschiedene Experimente zu finden. Zu (a) gibt es Studien von Loomis *et al.* (1993) und Sadalla & Montello (1989), die belegen, dass Menschen Winkel nahe 90 Grad sehr genau schätzen, allerdings Winkel unter 90 Grad über- und Winkel über 90 Grad unterschätzen. Zu (b) gibt es in der Literatur folgende Erkenntnisse: Routen mit mehr Richtungswechseln werden tendenziell als länger eingeschätzt (Downs & Stea, 1977). Außerdem erscheinen Abstände weiter, wenn ein Gebiet voller Menschen ist (Thorndyke & Hayes-Roth, 1982). Zu (c) soll hier das Experiment von Philbeck *et al.* (1997) erwähnt werden, in dem herausgefunden

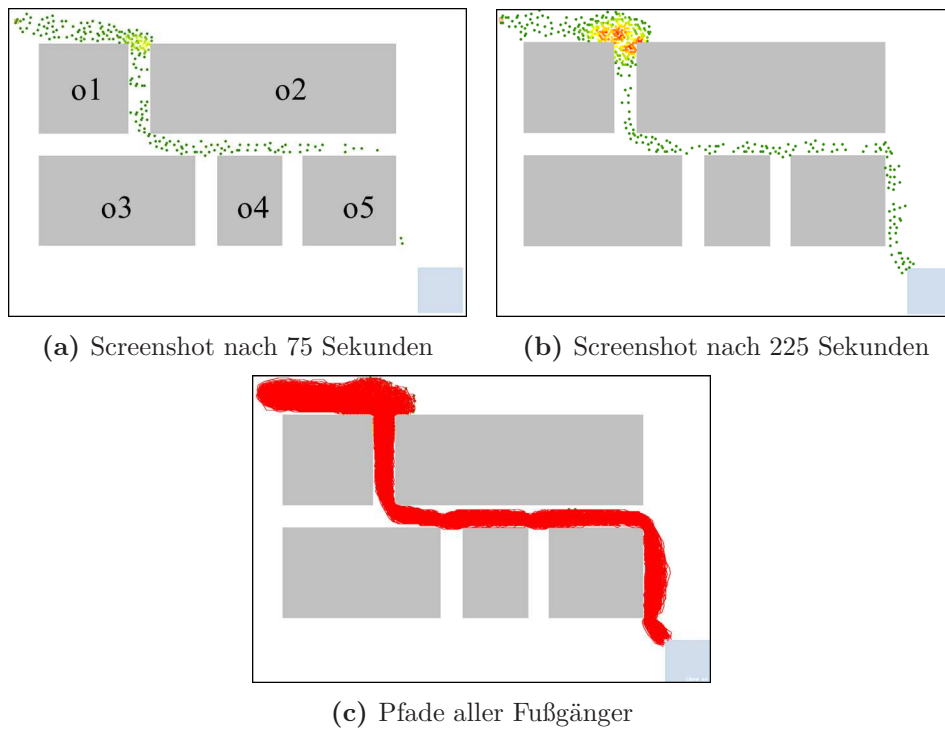


Abbildung 6.12: Simulationsergebnisse für den *Heuristic Path* Algorithmus: Es existiert genau ein Weg, dem alle Fußgänger folgen. Dieser Weg verläuft erkennbar nahe an der Luftlinie. Da als Kantengewicht keine Reisezeiten herangezogen werden, tritt nach einer gewissen Zeit ein Stau zwischen den Hindernissen *o1* und *o2* auf.

den wurde, dass Menschen Fehler in der Richtungsschätzung des Ziels meist im Uhrzeigersinn machen, also Ziele weiter östlich vermuten, als sie tatsächlich sind. Insgesamt ist dieser Fehler jedoch zu vernachlässigen, da er im Allgemeinen sehr klein ausfällt.

Aufgrund dieser Erkenntnisse wird bei der Abschätzung der Entfernung (also der heuristische Teil) ein Fehler eingebaut. Dies kann so erfolgen, indem ein Faktor aufmultipliziert wird, um die Fehleinschätzung der Menschen abzubilden.

Die Ergebnisse eines Simulationslaufs, bei dem ein Faktor von 10 % mit einfließt, ist in Abbildung 6.13 zu sehen. Der heuristische Teil wird mit dieser Einstellung um einen Faktor zwischen 0.9 und 1.1 unter- bzw. überschätzt. Die Wahl des Faktors erfolgt zufällig. Die Fußgänger verteilen sich durch diese Einstellung auf mehrere Wege. Durch die Fehleinschätzung ist der Weg nicht deterministisch festgelegt. Die Mehrheit der Wege führt immer noch entlang des deterministisch ermittelten Weges, es gibt aber auch Wege, die östlich und westlich des nördlichen Hindernisses, sowie bei der Umrundung der drei südlichen Hindernisse unterschiedliche Kanten wählen. Bei dem hier vorgestellten Algorithmus wird vorausgesetzt, dass die Fußgänger auch nicht-sichtbare Regionen kennen. Sie orientieren sich trotzdem an der Luftlinie, da sie diesen Weg als kürzesten einschätzen. Eine einfache Abwandlung des Algorithmus kann auch ortsfremde

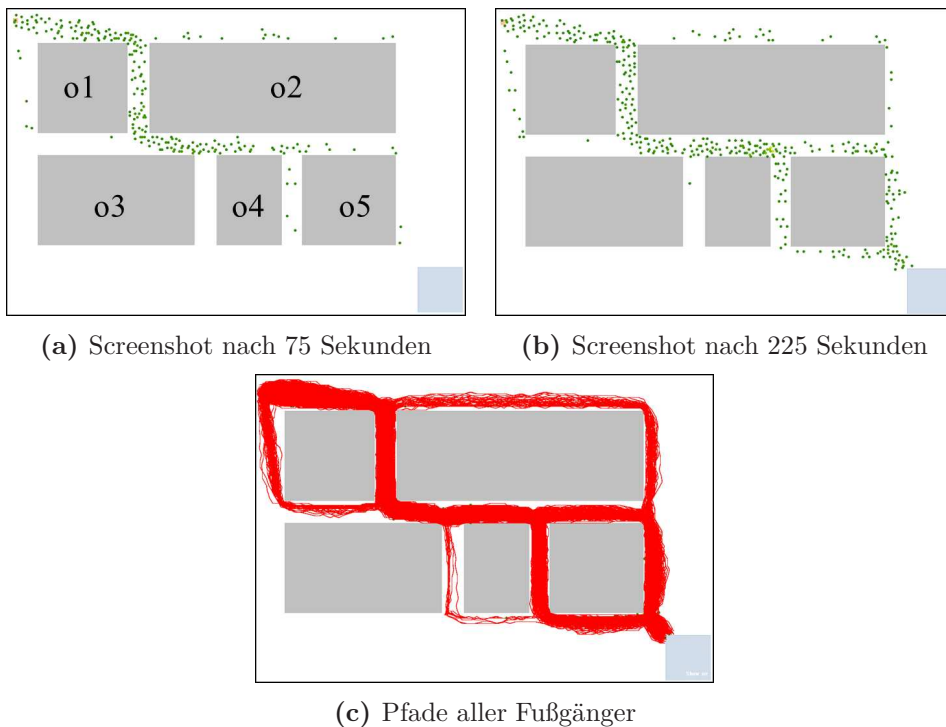


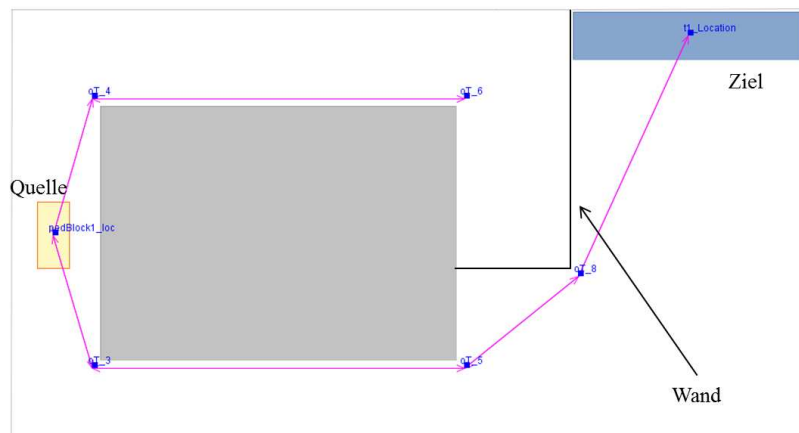
Abbildung 6.13: Simulationsergebnisse für den *Heuristic Path* Algorithmus: Die Fußgänger verteilen sich auf mehrere Wege. Die Mehrheit der Wege führt immer noch entlang des deterministisch ermittelten Weges, es gibt aber auch Wege, die östlich und westlich des nördlichen Hindernisses, sowie bei der Umrundung der drei südlichen Hindernisse unterschiedliche Kanten wählen.

Fußgänger abbilden, die sich an der Luftlinie zum Ziel orientieren. Er wird im nächsten Abschnitt vorgestellt.

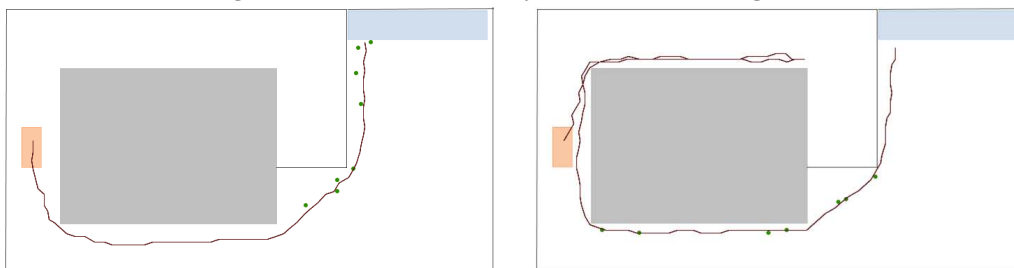
### 6.2.3 Strategie „Orientierung anhand der Luftlinie, aber ortsfremd“: Greedy Heuristic Path Algorithmus

Um den Aspekt der fremden Umgebung miteinzubeziehen, wird der oben beschriebene Algorithmus wie folgt verändert: Statt den gesamten Weg vorab zu berechnen, wird nun „gierig“ (engl. *greedy*) immer die Kante gewählt, dessen Endknoten das bisher geringste Knotengewicht aufweist. Deshalb ist dieser Algorithmus ebenfalls der Klasse der Greedy-Algorithmen (vgl. Unterabschnitt 6.2.1) zuzuordnen. Die Idee sei an folgendem Beispiel veranschaulicht (vgl. auch Abbildung 6.14a):

Angenommen der Weg, der am nächsten der Luftlinie folgt, führt in eine Sackgasse. Mit dem in Unterabschnitt 6.2.2 beschriebenen Algorithmus würde dies erkannt und dieser Weg somit ausgeschlossen (siehe Abbildung 6.14b) werden. Bei der Greedy-Variante führt der Weg zunächst in Richtung Sackgasse und erst dann wird der eigentliche Weg in Richtung Ziel gefunden (siehe Abbildung 6.14c). Umgesetzt wird das so, dass nur die Nachbarknoten untersucht werden und unter



(a) Beispielszenario zur Unterscheidung des *Heuristic Path* Algorithmus und des *Greedy Heuristic Path* Algorithmus.



(b) Ergebnisspur für einen Fußgänger, simuliert mit dem *Heuristic Path* Algorithmus.

(c) Ergebnisspur für einen Fußgänger, simuliert mit dem *Greedy Heuristic Path* Algorithmus

Abbildung 6.14: Ergebnisspur für einen Fußgänger, simuliert mit dem *Heuristic Path* Algorithmus und dem *Greedy Heuristic Path* Algorithmus

ihnen der beste Knoten ausgesucht wird, ohne den Rest des Wegs zu beachten. Der komplette Pfad wird also nicht im Voraus berechnet, sondern stückweise zusammengesetzt. Dazu wird den Fußgängern ein Gedächtnis gegeben. Dieses Gedächtnis wird in Form einer Liste mit besuchten Knoten und einer Liste von komplett untersuchten Knoten umgesetzt. Komplett untersucht bedeutet hier, dass alle Nachbarn des Knotens bereits untersucht worden sind. Dies verhindert, dass Fußgänger im Kreis gehen. Gibt es einen Knoten, der keine unbesuchten Nachbarknoten mehr hat, so wird soweit zurückgegangen, bis ein noch nicht besuchter Knoten gefunden wird. Der Pseudocode zum Algorithmus ist in Listing 3 dargestellt.

**Algorithmus 3** A\* Greedy

---

```

1: Gegeben: Gedächtnis  $M_F$  des Fußgängers;
2:  $M_F.openNodes$ ; // Alle besuchten Knoten
3:  $M_F.closedNodes$ ; // Alle komplett untersuchten Knoten
4: Knotengewichte  $g$ ;
5: Kantengewichte  $w$ ;
6: Input: Graph  $G(V, E)$ ; Startknoten  $s \in V$ ; Zielknoten  $z \in V$ ;
7: Ergebnis: nächster Knoten  $n$  auf dem Weg nach  $z$ ;
8: Initialisiere alle Kantengewichte  $w$  mit euklidischer Distanz;
9: Füge  $s$  zu  $M_F.closedNodes$ ;
10:  $g_s = 0$ ;
11: if  $s = z$  then
12:     return  $z$ ;
13: end if
14: Initialisiere  $minGewicht$  mit  $MAXFLOAT$ ;
15: for jede ausgehende Kante  $e_i \in E$  do
16:      $next = \text{Endknoten von } e_i$ ;
17:      $g_{next}^{neu} = g_s + w_{e_i}$ ;
18:     if  $next \in M_F.closedNodes$  then
19:         continue;
20:     end if
21:     if  $next \in M_F.openNodes$  then
22:         if  $g_{next} < g_{next}^{neu}$  then
23:             if  $minGewicht > g_{next} + d_{Luftlinie \text{ zum Ziel}}$  then
24:                  $minGewicht = g_{next} + d_{Luftlinie \text{ zum Ziel}}$ ;
25:                  $n = next$ ;
26:             end if
27:         end if
28:     else
29:          $g_{next} = g_{next}^{neu}$ ;
30:          $M_F.openNodes.add(next)$ ;
31:         Setze Vorgänger von  $next$ :  $pred(next) = s$ ;
32:     end if
33:     if  $minGewicht > g_{next} + d_{Luftlinie \text{ zum Ziel}}$  then
34:          $minGewicht = g_{next} + d_{Luftlinie \text{ zum Ziel}}$ ;
35:          $n = next$ ;
36:     end if
37: end for
38: if  $n$  not  $NULL$  then
39:     return  $n$ ;
40: end if
41: if  $pred(n)$  not  $NULL$  then
42:      $n = pred(n)$ ;
43:     Füge Knoten  $n$  zu  $M_F.closedNodes$  hinzu;
44:     return  $n$ ;
45: end if
46: return error;

```

---

Bezogen auf das Beispielszenario liefert dieser Algorithmus dasselbe Ergebnis wie der Heuristic Path Algorithmus, da der Weg nahe der Luftlinie ohne Hindernis verläuft.

#### 6.2.4 Strategie „Möglichst wenige Richtungswechsel“: Straight and Long Legs (SALL) Algorithmus

Ein weiterer Algorithmus soll abbilden, dass Fußgänger dazu neigen, möglichst wenig die Richtung zu wechseln, um so die Orientierung nicht zu verlieren. Dies wurde in einer Untersuchung von Golledge (1995) belegt sowie durch das Experiment von Kneidl & Borrmann (2011) bestätigt. Der Algorithmus ist dabei folgendermaßen aufgebaut. Wie auch bei den bereits beschriebenen Algorithmen werden die Kanten entsprechend bestimmter Kriterien gewichtet, um die Eigenschaften der Wegefindung abzubilden. Hier sollen Kanten präferiert werden, die möglichst lange in eine Richtung verlaufen, sowie abhängig von der Richtung, aus der man kommt, einen möglichst geringen Abweichungswinkel aufweisen. Gleichzeitig soll der Weg in Richtung Ziel führen. Diese drei Kriterien bilden das Kantengewicht:

- Relativer Wert zum maximalen Entfernungspunkt:

$$\mathit{maxDirection} = \frac{\mathit{dist}_{\mathit{maxDirectionPointOfEdge}}}{\mathit{dist}_{\mathit{furthestPoint}}}$$

- Abweichung von der Bewegungsrichtung ( $\omega$  möglichst nahe  $180^\circ$ ),  $e_1$ : eingehende Kante,  $e_2$ : ausgehende Kante

$$\omega_{e_1e_2} = \arccos \frac{e_1 \cdot e_2}{|e_1| \cdot |e_2|}$$

- Relative Verbesserung zum Ziel für Kante  $e = (i, j)$

$$\mathit{relativeEnhancement} = \frac{\mathit{dist}_{i,dest}}{\mathit{dist}_{j,dest}}$$

Der relative Wert zum maximalen Entfernungspunkt ist bestimmt durch die Relation des maximalen Punktes in Richtung der Kante bezogen auf den maximalen Entfernungspunkt aller ausgehenden Kanten. Der maximale Entfernungspunkt in Verlängerung der Kante ist der entfernteste Knoten, der weniger als  $10^\circ$  von der Kantenrichtung einer der ausgehenden Kanten abweicht.

Die Abweichung von der Bewegungsrichtung ist durch den Abweichungswinkel  $\omega$  bestimmt. Der Abweichungswinkel bezieht sich auf die bisherige Bewegungsrichtung, die durch die Kante, auf der sich der Fußgänger befindet, vorgegeben ist. Die relative Verbesserung zum Ziel bildet die Bewegungsrichtung in Richtung Ziel ab und wird durch die Entfernung des Anfangsknotens der Kante zum Ziel in Relation zum Endknoten der Kante zum Ziel gebildet. Dabei werden euklidische Abstände verwendet.

Die drei Bestandteile werden auf einen Wert zwischen 0 und 1 skaliert. Je näher der Wert an 1 liegt, desto höher ist das Gewicht und desto eher wird die Kante

gewählt. Die Kantengewichte werden also nach dem Maximum ausgewählt.  $maxDirection$  nimmt definitionsgemäß bereits ein Wert zwischen 0 und 1 an.  $\omega$  wird so skaliert, dass  $\pi$  maximal ist, wenn die Richtung der Kante der bisherigen Bewegungsrichtung entspricht. Zudem wird die Richtung der Kante in Relation zum Ziel geprüft. Ein Beispiel ist in Abbildung 6.15 gegeben. Führt sie weg vom Ziel, so wird der kleinere Winkel genommen (im Beispiel  $edge_2$ ), führt sie in Richtung Ziel, so wird der größere Winkel herangezogen (im Beispiel  $edge_1$ ). Bei allen entgegengesetzten Kanten wird der kleinere Winkel gewählt (im Beispiel  $edge_3$ ). Der Parameter  $relativeEnhancement$  wird ebenfalls skaliert: Dazu wird

■ Ziel

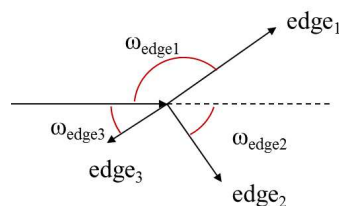


Abbildung 6.15: Beispiel zur Winkelbestimmung von  $\omega$

der Wert der Kante mit dem größten Verbesserungswert gewählt und als Relationswert gesetzt. Alle anderen ausgehenden Kanten bekommen so einen Wert zwischen 0 und 1 zugewiesen. Die skalierten Werte werden mit unterschiedlichen Faktoren potenziert und miteinander multipliziert. Das Gesamtgewicht bestimmt sich somit aus:

$$w_{edge} = maxDirection^\alpha \cdot \omega^\beta \cdot relativeEnhancement^\gamma$$

Zur Veranschaulichung dient das in Abbildung 6.16 dargestellte Beispielszenario.

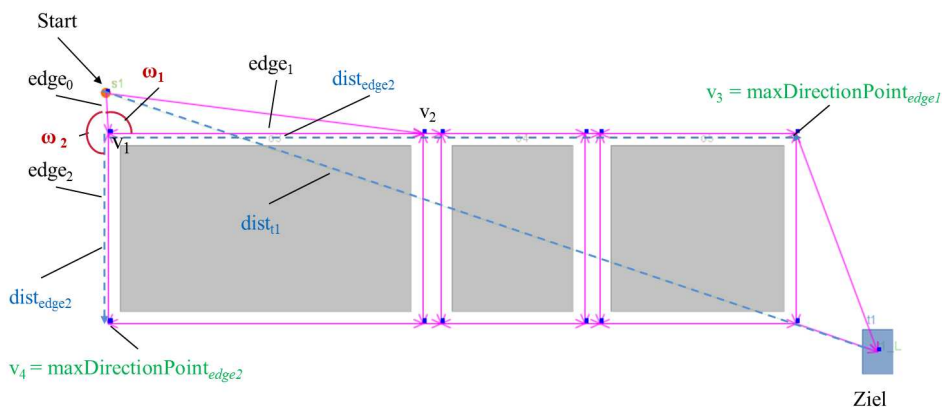


Abbildung 6.16: Beispielszenario zur Veranschaulichung der Kantengewichte des SALL Algorithmus

Ausgehend von der Kante  $edge_0$  wird am Knoten  $v_1$  die nächste Kante ausgewählt. Der maximale Entfernungspunkt ist für alle ausgehenden Kanten  $v_3$ . Der Wert

$dist_{maxDirectionPoint_{edge_1}}$  für die Kante  $edge_1$  entspricht deshalb dem Wert des Abstands zu Knoten  $v_3$  und  $maxDirection$  nimmt den Wert 1 an. Zusammengefasst ergibt sich folgendes Gewicht für die Kante:

$$\begin{aligned} angle_{\omega_1} &= 90^\circ / 180^\circ \\ relativeEnhancement &= dist_{v_1-t_1} / dist_{v_2-t_1} \\ maxDirection &= dist_{maxDirectionPoint_{edge_1}} / dist_{v_3} \end{aligned}$$

Für die Kante  $edge_2$  ergibt sich entsprechend das Gewicht:

$$\begin{aligned} angle_{\omega_2} &= 180^\circ / 180^\circ \\ relativeEnhancement &= dist_{v_1-t_1} / dist_{v_4-t_1} \\ maxDirection &= dist_{maxDirectionPoint_{edge_2}} / dist_{v_3} \end{aligned}$$

In diesem Beispiel hat also die Kante  $edge_1$  das größte Gewicht, wenn alle drei Bestandteile mit dem selben Gewicht einfließen ( $\alpha = \beta = \gamma$ ).

Der Einfluss der Gewichtungsfaktoren wurde in einer Parameterstudie untersucht. Dabei hat sich herausgestellt, dass die Ergebnisse den gewünschten Effekt am besten abbilden, wenn die Faktoren ähnlich gewichtet werden, mit einem geringeren Wert für  $\beta$ . Dies liegt darin begründet, dass in dem Parameter  $maxDirection$  die Richtung zum Ziel nicht enthalten ist, sondern nur die Länge der Richtung, in die maximal gegangen werden kann. Als geeignete Werte haben sich  $\alpha = 2.0$ ,  $\beta = 1.0$  und  $\gamma = 2.0$  herausgestellt.

Da bei Generierung des Fußgängers noch keine Richtung existiert, aus der er kommt, wird aus allen Kanten, die in Richtung Ziel führen, eine Kante zufällig zugewiesen, die in Richtung Ziel liegt. Falls keine Kante existiert, deren Endpunkt näher in Richtung Ziel liegt, wird aus allen ausgehenden Kanten zufällig eine Kante gewählt. Erst ab der zweiten zu traversierenden Kante werden Kantengewichte berechnet und der weitere Weg wird anhand der Gewichte ermittelt. Angewandt auf das Beispielszenario aus Abbildung 6.1 werden die in Abbildung 6.17 dargestellten Wege gefunden. Es ist zu erkennen, dass sich zwei Wege ausbilden. Durch die Symmetrie des Szenarios und der zufälligen Zuweisung der ersten Kante kann dies erklärt werden.

#### 6.2.4.1 Abbildung der Fehleinschätzung von Strecken

Um auch bei diesem Algorithmus die Imperfektion menschlicher Kognitionsprozesse abzubilden, werden jeweils die zwei ausgehenden Kanten (sofern vorhanden) mit dem größten Gewicht miteinander verglichen. Unterscheiden sie sich um weniger als einen bestimmten Prozentsatz (umgesetzt mit 10 %), so wird die nächste ausgehende Kante zufällig gewählt.

In dem hier gewählten Beispielszenario liefert diese Variation keine unterschiedlichen Ergebnisse, in dem Realbeispiel aus Kapitel 7 werden die Auswirkungen jedoch deutlich.



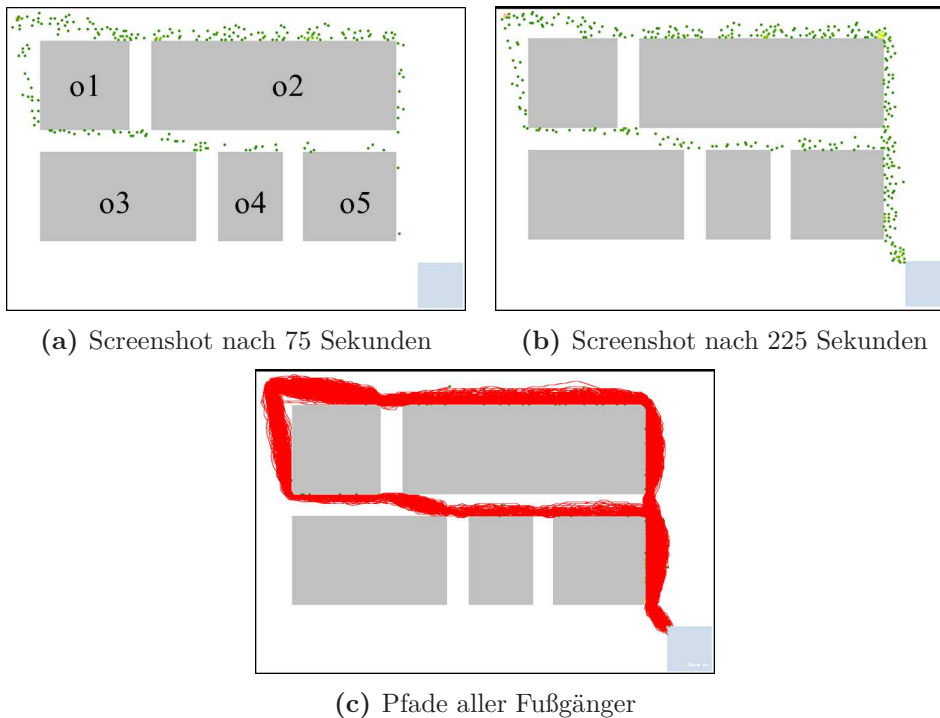


Abbildung 6.17: Simulationsergebnisse für den *SALL* Algorithmus: Es ist zu erkennen, dass sich zwei Wege ausbilden. Durch die Symmetrie des Szenarios und der zufälligen Zuweisung der ersten Kante kann dies erklärt werden.

### 6.2.5 Strategie „Anderen Personen folgen“: Probabilistic Choice Algorithmus

Die bisher beschriebenen Algorithmen modellieren die Wegewahl in Normalsituationen. Der hier beschriebene Algorithmus modelliert das Verhalten sowohl in Normalsituationen als auch in Evakuierungssituationen. Er bildet das Verhalten ab, anderen Menschen zu folgen, wenn man selbst ortsfremd ist. Dies kann modelliert werden, indem virtuelles Pheromon auf häufig besuchten Kanten abgelagert wird. Diese Kanten strahlen damit eine höhere Attraktivität, so dass der „Herdentrieb“ widergespiegelt wird.

Der Algorithmus folgt dem Prinzip von Ameisenalgorithmen (engl. *Ant Colony Optimisation*). Diese bedienen sich des Vorbilds aus der Natur: Ameisen markieren zurückgelegte Strecken auf der Suche nach der Futterquelle mit Pheromonablagerungen. Nachfolgende Ameisen orientieren sich an den Pheromonablagerungen und wählen bevorzugt den Weg mit der höheren Menge an Pheromon. Dadurch sammeln sich mittelfristig auf kürzeren Strecken stärkere Pheromonspuren an. Metaheuristiken nutzen dieses Phänomen zur Lösung von kombinatorischen Optimierungsproblemen (z.B. das Traveling Salesman Problem).

Dorigo (1992) ist mit seiner Diplomarbeit über diese Art von Verfahren Vorreiter mit folgender Grundidee: Gegeben sei ein Graph, auf dem „Ameisen“ entlang laufen. Auf jeder traversierten Kante hinterlässt jede Ameise eine Einheit virtuelles Pheromon. An jedem Knoten wird probabilistisch über die nächste Kante

in Abhängigkeit des Pheromons entschieden. Zusätzlich besitzt jede Ameise ein Gedächtnis über schon besuchte Orte und sucht diese nicht erneut auf (um Kreise zu vermeiden). Dieses Prinzip wurde bereits von Angus (Angus) auf die kürzeste Wege Suche in unbekanntem Terrain angewandt.

Hier wird das Prinzip auf menschliches Verhalten bei der Wegsuche übertragen. Fußgänger generieren Trampelpfade, was in städtischen Umgebungen sehr gut zu beobachten ist. In Abbildung 6.18 ist der Königsplatz in München aus der Luft zu sehen; dort sind über die Zeit entstandene Trampelpfade gut zu erkennen. Diese können durch den Einsatz von Ameisenalgorithmen nachempfunden werden, in dem Wege, die zunächst von einigen wenigen Personen gegangen werden, nach und nach von immer mehr Personen gewählt werden aufgrund der Pheromonablagerungen.

Aber nicht nur Trampelpfade können über die Pheromonablagerungen abgebildet werden, sondern auch das Verhalten, anderen Menschen in der Überzeugung zu folgen, dass diese den Weg kennen. Dies trifft vor allem auf ortsfremde Fußgänger zu. Sie treffen Entscheidungen unter Miteinbeziehung von lokalen Gegebenheiten, da kein globales Wissen vorhanden ist. Zudem kann der Nicht-Determinismus bei Entscheidungen von Fußgängern sehr gut durch den Algorithmus abgebildet werden, da die Auswahl der nächsten zu traversierenden Kante probabilistisch erfolgt.



*Abbildung 6.18:* Ausschnitt aus einer Luftbildkarte (Quelle: maps.google.com). Der Ausschnitt zeigt den Königsplatz in München, auf dem deutlich zwei Trampelpfad zu erkennen sind.

Diese Annahmen fließen in den Algorithmus in Form von unterschiedlichen Komponenten in die Berechnung der Kantengewichte mit ein. Das Kantengewicht

setzt sich aus drei unterschiedlichen Teilen zusammen. Zwei Teile sind statisch, also unveränderlich über die Zeit. Dazu kommt ein dynamischer Teil, dessen Wert sich über die Zeit verändern kann. Die statischen Teile bestehen aus der Winkelabweichung zwischen Luftlinie und ausgehender Kante in Richtung Ziel sowie der relativen Verbesserung des euklidischen Abstands zwischen dem aktuellen Ort und dem Endknoten der Kante. Ein Beispiel ist in Abbildung 6.19 gegeben.

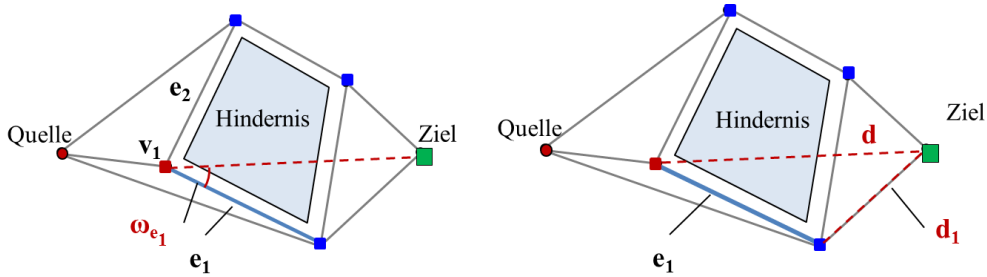


Abbildung 6.19: Links: Bestimmen des Winkels  $\omega_{e_1}$  für die Kante  $e_1$ . Rechts: Relative Verbesserung zum Ziel für Kante  $e_1$

Formal wird der Winkel berechnet aus:

$$\omega_{e_1} = \arccos \frac{e_1 \cdot d}{|e_1| \cdot |d|}$$

Der Unterschied zur Winkelberechnung des *SALL*-Algorithmus besteht darin, dass nicht der Winkel zwischen bisheriger Richtung und ausgehender Kante berechnet wird, sondern der Winkel zwischen ausgehender Kante und Abweichung zum Ziel.

Die relative Verbesserung zum Ziel (*dest*) für die Kante  $e = (i, j)$  wird durch

$$relativeEnhancement = \frac{dist_{i,dest}}{dist_{j,dest}}$$

angegeben.

Der dritte und dynamische Teil des Gewichts ist definiert durch die Pheromonablagerungen auf der Kante. Dabei gilt folgendes:

- $\sum Pheromone_{Graph} = \text{Anzahl Kanten im Graph}$
- Jede Kante wird mit Pheromonmenge = 1 initialisiert
- pheromoneUpdate (0.2222<sup>3</sup>): Pheromon-Einheit, die von jedem Fußgänger auf jeder Kante, die er besucht, abgelegt wird
- pheromoneDecay (0.0005<sup>3</sup>): Pheromon-Menge pro Kante, die nach jeder Iteration „verpufft“

Folgende Gleichung muss zu jeder Zeit erfüllt sein:

$$totalSystemPheromone = \frac{noOfAnts}{pheromoneDecay}$$

<sup>3</sup>entnommen aus Angus (Angus)

mit

$$noOfAnts = \sum_{\text{alleKanten}} (\text{pheromoneUpdate} \cdot noOfPeds)$$

Alle Kanten werden in jeder Iteration mit Pheromon wie folgt belegt:

$$\text{pheromone}_{edge} = \text{pheromone}_{edge} + (\text{pheromoneUpdate} \cdot noOfPeds)$$

Damit Kanten nicht bevorzugt werden, die länger sind und auf denen sich dadurch mehr Fußgänger bewegen können, wird die Anzahl der Fußgänger, die eine Kante traversieren, in Abhängigkeit zur Länge gesetzt. Dieser Wert entspricht dem *noOfPeds*, der angibt, wieviel Pheromon deponiert wird.

Die drei Teile des Gewichts werden mit Gewichtungsfaktoren potenziert und miteinander multipliziert, so dass sich folgendes Gesamtgewicht pro Kante ergibt:

$$\text{weight}_{edge} = \omega^\alpha \cdot \text{relativeEnhancement}^\beta \cdot \text{pheromone}^\gamma$$

Die Gewichte werden maximiert, d.h. dass je größer das Gewicht, desto höher die Wahrscheinlichkeit, dass die Kante gewählt wird. Wie beim *SALL* Algorithmus (vgl. Unterabschnitt 6.2.4) wird der Winkel  $\omega$  auf einen Wert zwischen 0 und 1 skaliert. Dabei wird für alle ausgehenden Kanten, die weg vom Ziel führen, der kleinere Wert genommen, und für die Kanten in Richtung Ziel der größere Wert. Auch die Werte der Komponente *relativeEnhancement* werden auf eine Zahl zwischen 0 und 1 skaliert. Dies geschieht wie bei dem *SALL* Algorithmus, indem der Wert der Kante mit der größten Verbesserung zum Ziel als Referenzwert genommen und die Werte der anderen Kanten in Relation dazu gesetzt werden.

Die Kante wird probabilistisch ausgewählt, es wird also nicht die Kante mit dem größten Gewicht gewählt, sondern das Gewicht erhöht die Wahrscheinlichkeit, dass die Kante gewählt wird. Eine verbreitete Methode, um diese Auswahl zu treffen, ist die Rouletterad Selektion (Bäck, 1996) bei heuristischen Verfahren. Hier wird die Gesamtsumme aller Gewichte auf eins skaliert, und die einzelnen Gewichte als Kreissegmente angetragen. Das Rad wird gedreht (Zufallszahl gezogen) und die Kante, in dessen Segment das Rad zu stehen kommt, wird ausgewählt. Zur Veranschaulichung ist ein Beispiel in Abbildung 6.20 dargestellt. Die Größe des Kreissegments entspricht also einer skalierten Wahrscheinlichkeit, die sich wie folgt berechnet:

$$Prob_{edge} = \frac{\text{weight}_{edge}}{\sum_{\text{outgoingEdges}} \text{weight}_{edge}}$$

Es wird eine Zufallszahl zwischen 0 und 1 generiert und entsprechend des Wertes wird diejenige Kante gewählt, in dessen Bereich die Zahl liegt.

Der Algorithmus findet zunächst eine Reihe von Wegen, wovon sich die besseren bezüglich der gewählten Metrik, also des Kantengewichts, nach einiger Zeit durchsetzen. Angewandt auf das Beispielszenario ist das Ergebnis in Abbildung 6.21 dargestellt. Hierbei werden die Gewichtungsfaktoren gleich gesetzt

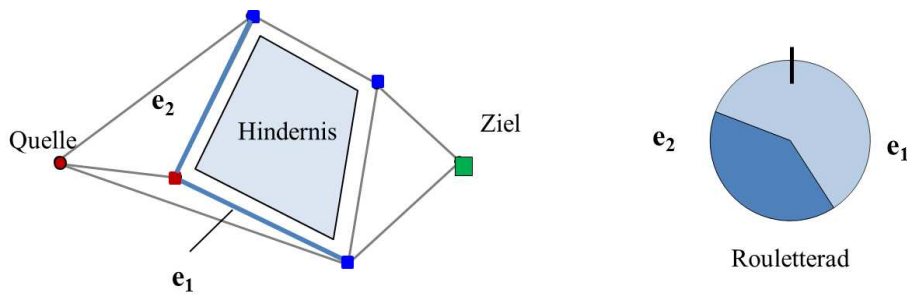


Abbildung 6.20: Beispiel zur Rouletterad Selektion: Rechts werden die Gewichte der Kanten  $e_1$  und  $e_2$  als Kreissegmente angetragen. In diesem Fall würde  $e_1$  gewählt, da der Zeiger in deren Segment zum Stehen gekommen ist.

mit  $\alpha = 1, \beta = 1, \gamma = 1$ . Um den Effekt besser darstellen zu können, wird die Simulationszeit von 300 auf 600 Sekunden hochgesetzt. In Abbildung 6.21c ist statt der Pfade aller Fußgänger ein Screenshot der Simulation zu einem späteren Zeitpunkt zu sehen. Die Pfade der Fußgänger sind hier nicht sehr aussagekräftig, da alle Kanten benutzt werden, und nach und nach die schlechteren Kanten aussortiert werden. Es ist zu erkennen, dass die Fußgänger sich zunächst auf alle Kanten aufteilen, mit einer größeren Mehrheit aber die Kanten gewählt werden, die bezüglich der relativen Verbesserung zum Ziel sowie Abweichung von der Zielrichtung gut sind. Nach und nach bilden sich dann zwei Wege aus (Abbildung 6.21b), bis schließlich ein einziger Weg übrigbleibt (Abbildung 6.21c). Dieser Algorithmus kann auch gut mit einem der anderen Algorithmen verknüpft werden, indem Personen, die bezüglich einer anderen Metrik laufen, ebenfalls Pheromon auf den Kanten hinterlassen. Dadurch kann das Verhalten, anderen Personen zu folgen, gut nachgeahmt werden. Im Beispiel aus Abbildung 6.22 laufen fünfzig Prozent der Personen nach dem *Fastest Path* Algorithmus und fünfzig Prozent nach dem *Probabilistic Choice* Algorithmus. In diesem Fall wird die Gewichtung des Pheromonteils mit  $\alpha = 2$  höher gewählt, da die Fußgänger den anderen ortskundigen Fußgängern folgen sollen. Die Ergebnisse sind in Abbildung 6.22 dargestellt. Zu Beginn werden mehrere Wege gefunden, aber bereits nach 225 Sekunden (Abbildung 6.22b) bilden sich klare Hauptwege aus. Dies ist auch an den Pfaden zu erkennen. Die Mehrzahl der Fußgänger folgt nach einer gewissen Zeit dem schnellsten Weg, da die Pheromonmenge auf diesen Kanten am höchsten ist.

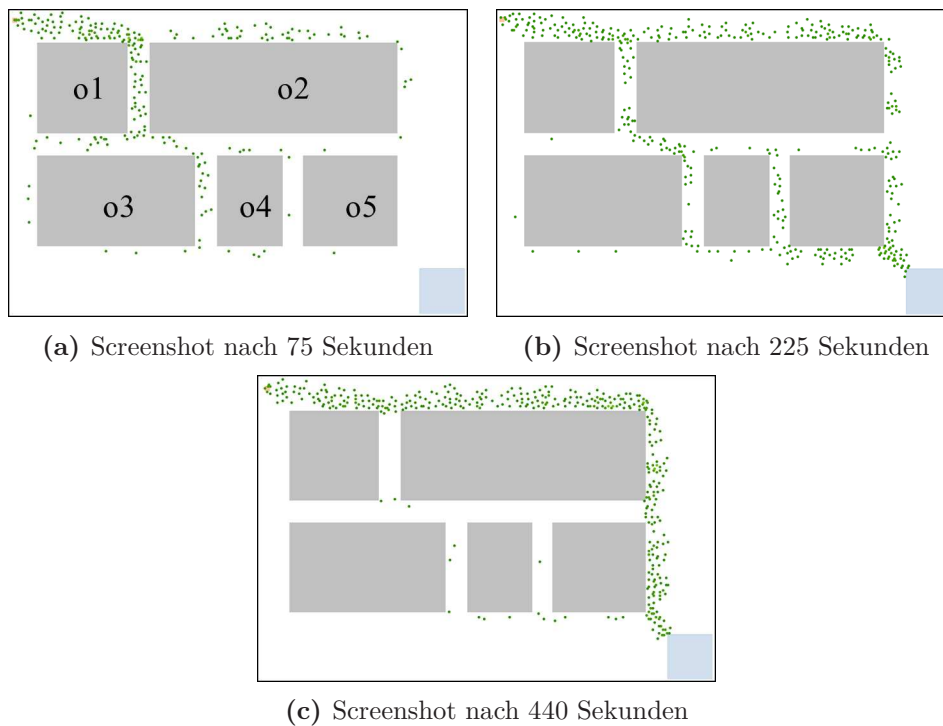


Abbildung 6.21: Simulationsergebnisse für den *Probabilistic Choice* Algorithmus: Es ist zu erkennen, dass die Fußgänger sich zunächst auf alle Kanten aufteilen, mit einer größeren Mehrheit aber die Kanten gewählt werden, die bezüglich der relativen Verbesserung zum Ziel sowie Abweichung von der Zielrichtung gut sind. Nach und nach bilden sich dann zwei Wege aus, bis schließlich ein einziger Weg übrigbleibt.

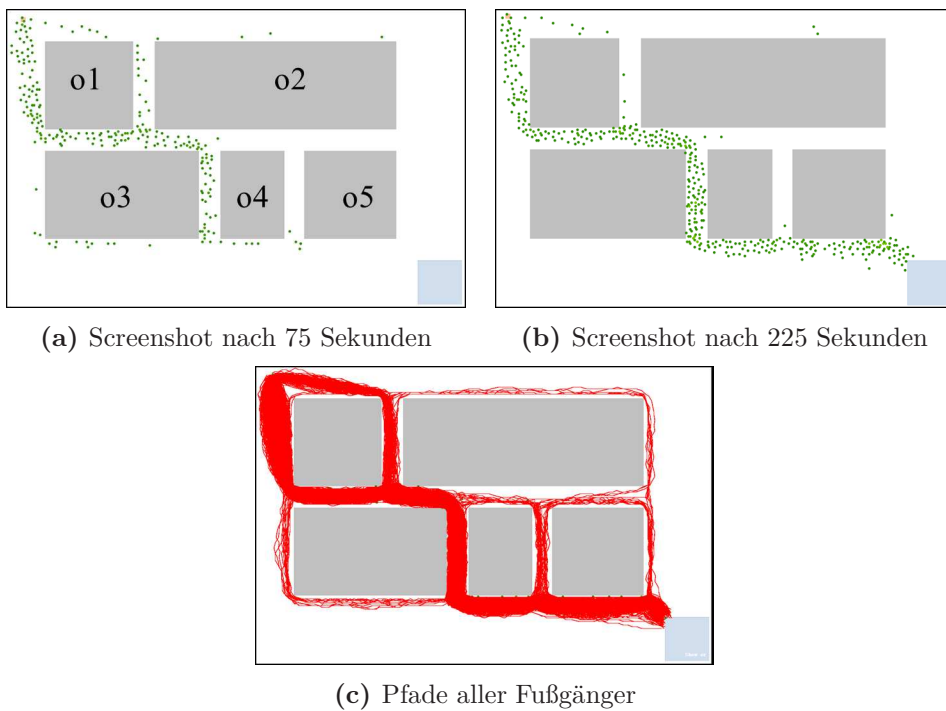


Abbildung 6.22: Simulationsergebnisse für die Kombination von Schnellstem Weg und *Probabilistic Choice* Algorithmus: Zu Beginn werden mehrere Wege gefunden, aber bereits nach 225 Sekunden bilden sich klare Hauptwege aus. Die Mehrzahl der Fußgänger folgt nach einer gewissen Zeit dem schnellsten Weg, da die Pheromonmenge auf diesen Kanten am höchsten ist.

### 6.2.6 Evakuierung – zum nächstgelegenen Ziel laufen

Bei Simulationsszenarien, die den Normalfall simulieren, bekommt jeder Fußgänger, sobald er von der Quelle erzeugt wurde, ein Ziel zugewiesen (vgl. Abschnitt 3.1). Zu diesem Ziel steuert er dann anhand des ihm zugewiesenen Algorithmus.

Dies muss bei Evakuierungen nicht der Fall sein. Statt des ihm zugewiesenen Ziels sollte der Fußgänger in der Lage sein, das ihm nächstgelegene Ziel zu wählen. Genau dies wird hier abgebildet. Statt eines fest zugewiesenen Ziels wird das nächstgelegene Ziel gesucht und ein Weg dorthin berechnet. Die Zuweisung des nächsten Ziels kann für alle bisher vorgestellten Algorithmen angewandt werden. Im Beispiel sei dies für den schnellsten Weg dargestellt. Als Szenario dient das in Kapitel Abschnitt 4.3 vorgestellte Szenario eines Bürogebäudes. In jedem Raum befinden sich einige Personen abhängig von der Raumgröße. Das Gebäude (Stockwerk) besitzt insgesamt drei Ausgänge, davon zwei Treppen, eine auf der östlichen und eine auf der westlichen Seite des Gebäudes, sowie einen Fahrstuhl in der Mitte des Gebäudes. Die Geometrie des Szenarios ist Abbildung 6.23, die Verteilung der Personen auf die Büros ist in Abbildung 6.24 dargestellt.



Abbildung 6.23: Bürogebäude als Beispielszenario: Die Etage hat insgesamt drei Ausgänge, davon zwei Treppen und einen Fahrstuhl.

In Abbildung 6.26 und 6.25 sind Screenshots der Simulation zu sehen, wenn sie im Evakuierungsmodus gestartet wird. Alle Fußgänger laufen nach dem schnellsten Weg. Es ist deutlich zu erkennen, dass die Fußgänger das nächstgelegene Ziel ansteuern, um das Gebäude zu verlassen. Was gleichzeitig auffällt ist, dass die Fußgänger, die über den Lift das Gebäude verlassen, am längsten brauchen, um aus dem Gebäude zu gelangen. Dies liegt daran, dass der Lift eine geringere Kapazität als die Treppen aufweist. Dadurch stauen sich die Personen vor dem Lift auf ihrem Weg aus dem Gebäude heraus. Dies ist – wenn auch nur schematisch



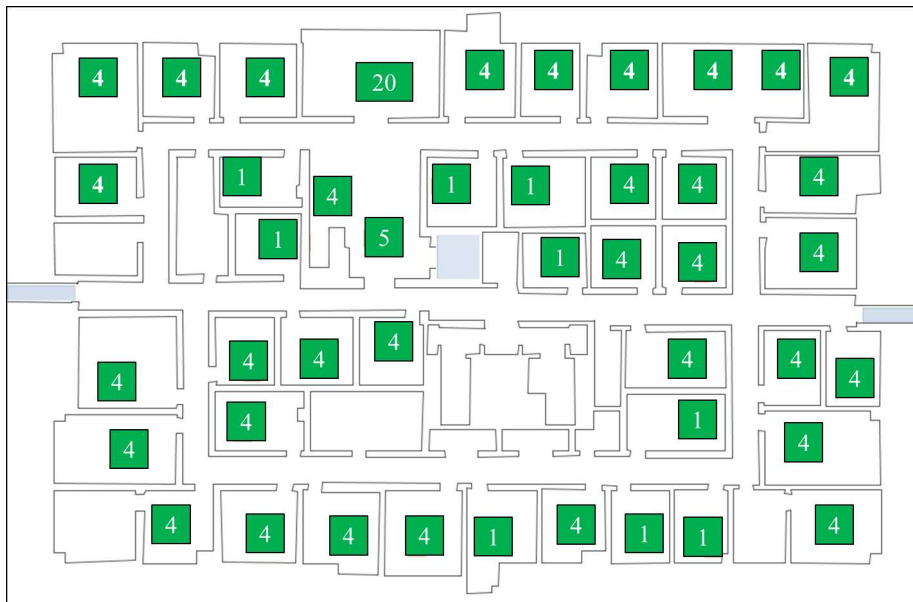


Abbildung 6.24: Belegung der Büros zu Beginn der Evakuierung.

– ein Beispiel dafür, dass Simulationen dazu dienen können, Schwachstellen bei der Planung von Gebäuden aufzudecken. Würde der Lift in der Mitte durch eine Treppe ersetzt, bestünde dieses Problem nicht, und die Evakuierungszeit würde gesenkt.

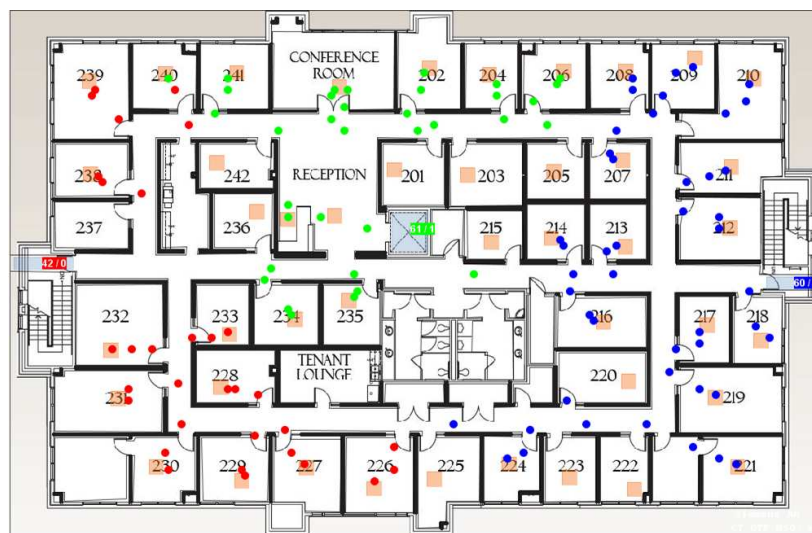


Abbildung 6.25: Screenshot zu Beginn der Simulation

### 6.3 Einbinden von Landmarks

Alle bisher vorgestellten Algorithmen finden einen Weg entsprechend der implementierten Metrik von einer Quelle zu einem Ziel. In der Literatur finden sich jedoch Belege dafür, dass sich Fußgänger in der Wegfindung an Landmarks ori-

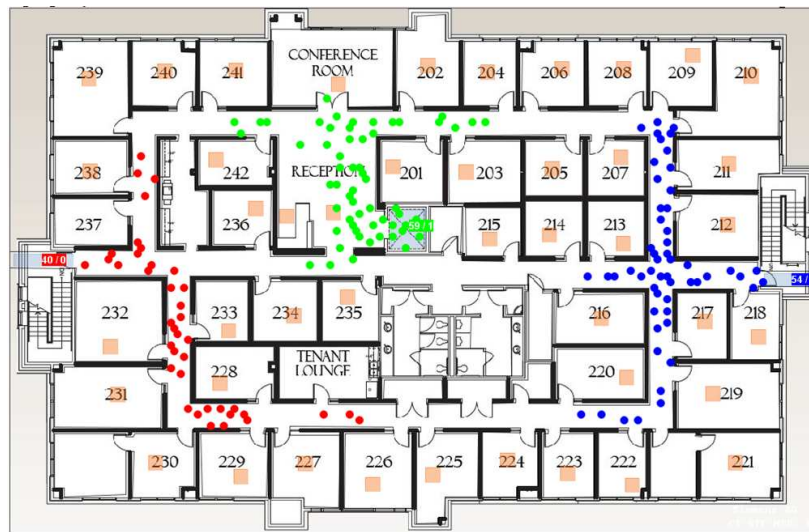


Abbildung 6.26: Screenshot der Simulation nach 10 Sekunden. Es ist klar erkennbar, dass die Personen sich in Richtung des ihnen am nächstgelegenen Ziels bewegen.

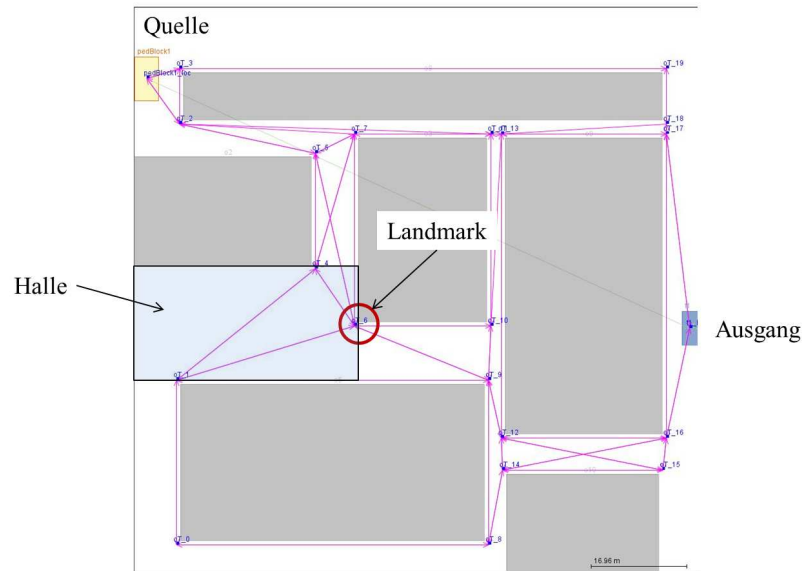
entieren, da Landmarks ein Teil der kognitiven Karten sind (Golledge, 1999a). Landmarks können hierbei personenbezogene prägnante Orte darstellen oder allgemein bekannte Orte.

Um dies auch in der Simulation abbilden zu können, wird das Konzept von Landmarks eingeführt (Krötzing, 2011). Dazu werden Knoten im Graph als Landmarks definiert. Die Definition der Landmarks muss manuell erfolgen, da Landmarks individuell verschieden sein können. Die Landmarks entsprechen Knoten, die in jedem Fall passiert werden müssen.

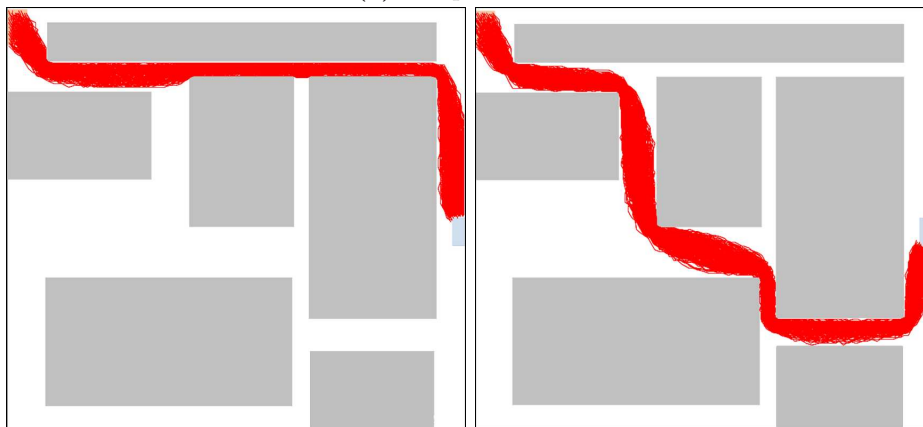
Statt nach dem gesamten Weg auf einmal zu suchen, werden Zwischenziele eingefügt, um Teilwege zu berechnen. Erst wenn ein Fußgänger an einem Zwischenziel angekommen ist, wird der nächste Teilweg berechnet. Dabei können unterschiedlich viele Landmarks bestimmt werden. Die Landmark-Zuordnung erfolgt über die Quelle-Ziel-Relationen; pro Quelle können unterschiedlich viele Landmarkpfade angegeben werden. Diese Pfade werden mit einem prozentualen Anteil gewichtet. So kann die Modellierung sehr flexibel gestaltet werden.

In Abbildung 6.27 ist ein einfaches Beispiel angegeben, das die Idee der Landmarks veranschaulicht. Das Szenario stellt schematisch ein Gebäude dar, mit einem Ausgang auf der östlichen Seite. Die Situation sei folgende: Personen verlassen einen Raum, der hinter der Quelle liegt und möchten in Richtung Ausgang gehen. Das Gebäude besteht neben den Gängen auch aus einer Haupthalle, die den Personen bekannt ist, die sich in dem Gebäude aufhalten. Wenn nun kein Landmark in der Haupthalle platziert wird, bewegen sich die Leute entlang des schnellsten Wegs Richtung Ausgang. Dieser führt durch einen schmalen Gang (Abbildung 6.27b). Dieser Gang ist jedoch ein Nebenweg und ist den Personen, die sich in diesem Gebäude befinden, nicht so bekannt. Vielmehr kennen die Personen diese Haupthalle, da sich in dieser Halle z.B. eine Kaffeetheke befindet. Wenn nun das Landmark mit in die Wegefindung einbezogen wird, laufen

die Fußgänger zunächst in Richtung des Landmarks und erst nach Erreichen des Landmarks wird der weitere Weg gesucht (Abbildung 6.27c). Dieses Beispiel stellt



(a) Beispielszenario



(b) Pfade der Fußgänger ohne  
Miteinbeziehung von Landmarks

(c) Pfade der Fußgänger unter  
Miteinbeziehung von Landmarks

Abbildung 6.27: Unterschied zwischen Simulation mit und ohne Landmarks. Personen suchen zunächst einen Weg in Richtung des Landmarks bevor sie in Richtung Ausgang laufen.

das Konzept der Landmarks schematisch dar. Um eine realistischen Einsatz von Landmarks zu modellieren, müssen genauere Verteilungen und unterschiedliche Wegfindungsalgorithmen gewählt werden. Dies wird in Kapitel 7 anhand eines Realbeispiels demonstriert.

## 6.4 Zusammenfassung

In diesem Kapitel wurden verschiedene Algorithmen zur Abbildung von menschlicher Wegefindung vorgestellt, aufbauend auf Erkenntnissen aus unterschiedlichen Richtungen der Wissenschaft. Dabei wurden kognitive Prozesse, die sich in der Wegefindung der Menschen abspielen, in Algorithmik überführt. Personen wurden dazu in ortskundig und ortsfremd unterteilt und deren Verhalten für Normalsituationen und für den Evakuierungsfall betrachtet. Daraus resultieren unterschiedliche Algorithmen, die Strategien, wie die Orientierung entlang der Luftlinie oder das Laufen möglichst langer Strecken ohne Richtungswechsel, mit einbeziehen. Die unterschiedlichen Algorithmen wurden so konfiguriert, dass sie nicht immer ausschließlich den besten Weg bezüglich einer Charakteristik ausgeben. Diese Auswahl wurde unterschiedlich umgesetzt; es werden die  $k$  besten Wege ausgegeben, oder alle Wege, deren Gesamtgewicht innerhalb eines bestimmten Bereichs liegen oder die Auswahl erfolgt probabilistisch.

Die Tatsache, dass sich Personen an Landmarks orientieren, wurde ebenfalls auf das Modell übertragen. Durch die Kombination dieser Algorithmen kann die taktische Ebene in Simulationsmodellen sehr gut abgebildet werden, da die Variation in der menschlichen Wegefindung beachtet wird und unterschiedliche Typen modelliert werden können. Die Ergebnisse der Algorithmen wurden anhand von schematischen Testfällen dargestellt. Die Übertragung der Algorithmen auf ein Realbeispiel und die Validierung werden in Kapitel 7 ausführlich behandelt.

## Kapitel 7

# Validierung

Die in Kapitel 6 entwickelten Algorithmen zur Abbildung realistischen Wegefindungsverhaltens werden in diesem Kapitel anhand von Daten eines Experiments validiert. Dazu wird eine neue Vergleichsmetrik eingeführt, die es erlaubt, aussagekräftige Vergleiche zwischen den Realdaten und den Simulationsdaten durchzuführen. Die Ergebnisse werden ausführlich diskutiert und bewertet.

### 7.1 Einführung

Um realistische Simulationsergebnisse zu erzeugen, muss ein Simulationsmodell zunächst gegenüber Realdaten kalibriert und anschließend validiert werden.

Kalibrierung bedeutet in diesem Zusammenhang das Adjustieren von Simulationsparametern, um den Simulator an einen zu simulierenden Sachverhalt anzupassen. Geschwindigkeiten, maximale Dichten und Gruppengröße der zu simulierenden Personen spielen dabei eine wichtige Rolle. Diese Parameter können sehr unterschiedliche Werte annehmen, abhängig von Zeit, Ort und Anlass des zu simulierenden Szenarios. Gehen Fußgänger beispielsweise auf ihrem täglichen Arbeitsweg ein eher hohes Tempo (Davidich & Köster, 2012b), schlendern sie beim Besuch eines Freizeitparks, da der Weg sie hier nicht möglichst schnell zu einem Ziel führen soll, sondern der Weg an sich schon die Unternehmung ist.

#### **Kalibrierung**

Für die Kalibrierung der Lokomotionsebene spielt die Demographie der zu simulierenden Personen eine große Rolle. Die Erhebung dieser Daten kann durch Experimente, durch die Auswertung von Fragebögen, Videodaten, Messungen oder statistische Auswertungen erfolgen. Aufgrund dieser Eingangsdaten wird eine Geschwindigkeits-Dichte-Relation ermittelt, wie auch in Abschnitt 3.1 beschrieben. Da auf diese Geschwindigkeits-Dichte-Relation eine Vielzahl von Parametern einen Einfluss haben und die Wechselwirkungen der einzelnen Parameter analytisch nicht bestimmbar sind, erfolgt das Anpassen der einzelnen Parameter durch genetische Algorithmen (z.B. Klein *et al.*, 2010, Davidich & Köster, 2012a, Höcker, 2010).

Für die Kalibrierung der taktischen Ebene spielt die Ortskenntnis der Personen sowie die Lage von bekannten Orten (Landmarks) eine wichtige Rolle. Diese Daten müssen aus vorliegenden Daten abgeleitet werden, um die Simulationsparameter entsprechend einzustellen. Eine systematische Kalibrierung wird hier nicht weiter betrachtet, da derzeit zu wenig experimentelle Daten vorliegen.

### Validierung

Die Validierung dient zur Überprüfung eines kalibrierten Modells, also zum Vergleich der Simulationsergebnisse mit der Realität. Ziel ist es nachzuweisen, dass die vom Simulator berechneten Ergebnisse mit der Realität übereinstimmen. Erst dann kann ein Modell als zuverlässig eingestuft werden.

Das Vorgehen bei der Validierung ist dreistufig: Zunächst müssen Daten zum Vergleich erhoben werden. Dies kann unterschiedlich geschehen; es können Messungen aller Art wie beispielsweise Inertialsensorik, Zeitmessungen, Lokalisierungsmethoden (z.B. W-Lan, RFID), empirische Daten oder Videosequenzen herangezogen werden. Die Daten müssen so aufbereitet werden, dass daraus Metriken abzuleiten sind, die einen aussagekräftigen Vergleich ermöglichen. Beispiele für Metriken basieren auf Fundamentaldiagrammen, Geschwindigkeiten, durchschnittlichen Gruppengrößen, maximalen Dichten oder Zeiten. Im letzten Schritt werden die abgeleiteten Metriken verglichen und die Güte der Simulation bestimmt.

Der schematische Ablauf von Kalibrierung und Validierung ist in Abbildung 7.1 dargestellt. Eine Herausforderung beim Kalibrierungs- und Validierungsprozess

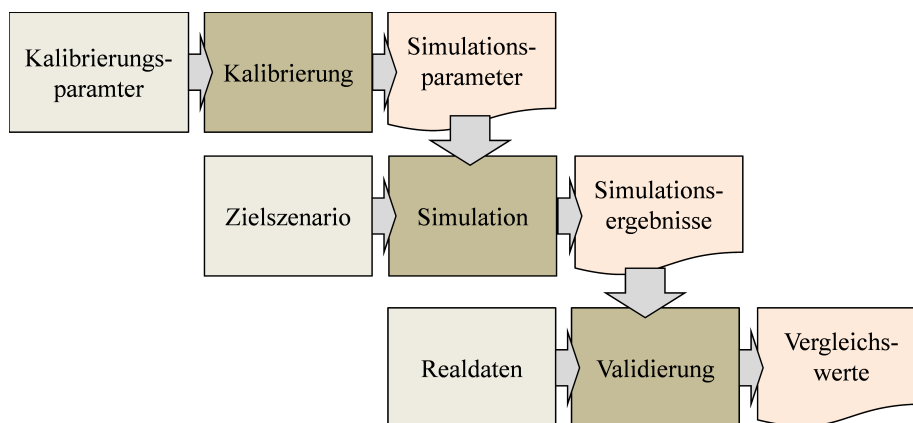


Abbildung 7.1: Schematischer Überblick über den Kalibrierungs- und Validierungsprozess

von Fußgängersimulationen ist die Beschaffung von Vergleichsdaten, also Realdaten. Vereinzelt stehen Daten aus Experimenten zur Verfügung, eine breite Datenbasis ist jedoch nicht vorhanden. Dies gilt insbesondere für Daten, die das menschliche Wegefindungsverhalten beschreiben, das im Fokus dieser Arbeit steht. Daraus entstand die Motivation, die hier zu validierenden Wegefindungsalgorithmen mithilfe eines eigens durchgeführten Experiments zu validieren. Da sich die Arbeit vor allem mit der taktischen Ebene auseinandersetzt, sei auf Arbeiten von

Davidich & Köster (2011, 2012a,b) verwiesen, die sich mit der Kalibrierung und Validierung der Lokomotionsebene des hier vorgestellten Simulationsmodells beschäftigen.

## 7.2 State of the Art: Validierung

In der Literatur findet sich bereits eine Reihe von Ansätzen zur Validierung von Fußgängersimulationen. Im Folgenden wird daher ein Überblick über die wichtigsten Arbeiten in diesem Bereich gegeben.

### Arbeiten zur Validierung der Lokomotionsebene

Der in Deutschland aktive Verein RiMEA (Richtlinie für mikroskopische Entfluchtungsanalysen) entwickelt eine Richtlinie zur Validierung von mikroskopischen Simulationsmodellen. Hauptziel dieser Richtlinie ist es, die Vergleichbarkeit der Ergebnisse unterschiedlicher Modelle zu erhöhen und eine Qualitätsprüfung einzuführen. Die Richtlinie gibt Simulationsparameter wie Geschwindigkeitsverteilung, Platzbedarf der Fußgänger etc. vor, die für die Testfälle einzustellen sind. Die Testfälle sind sehr modular, um gezielt bestimmte mikroskopische Aspekte wie Bahnenbildung oder Staus vor Engstellen zu überprüfen. Für jeden Testfall sind Kriterien definiert, nach denen der Test bestanden oder nicht bestanden ist. Die Tests und deren Vorgaben wurden teilweise durch eine Reihe von Laborexperimenten ermittelt. Seyfried *et al.* (2009) führte Experimente durch, um den Fluss durch Engstellen zu untersuchen. Auch der bidirektionale Fluss und die entsprechenden Bewegungsmuster wurden von Zhang *et al.* (2012) experimentell untersucht. Die Richtlinie beinhaltet bisher keine Vorgaben zur Validierung der taktischen Ebene.

Die Kalibrierung und Validierung von kräftebasierten Modellen wurde von Chraïbi (2012) untersucht.

Für die Validierung von agentenbasierten Modellen wurde in Torrens *et al.* (2012) ein erweiterbares Simulationsframework verwendet, um Metriken für mikroskopische und makroskopische Bewegungsmuster zu definieren. Aguirre *et al.* (2011) beschäftigt sich ebenfalls mit der Validierung von agentenbasierten Modellen.

Zusätzlich zu diesen allgemeineren Validierungsframeworks gibt es eine Vielzahl von Experimenten, die zur Gewinnung von Validierungsdaten für komplexere Szenarien durchgeführt wurden (Helbing *et al.*, 2007, Moussaïd *et al.*, 2009, Seyfried *et al.*, 2009, Butenuth *et al.*, 2011, Davidich & Köster, 2012b, Klüpfel, 2006). Das Vorgehen bei diesen Experimenten ist meist so, dass aus den ermittelten Daten Trajektorien extrahiert werden, die als Basis für die Definition von Validierungsmetriken wie Dichten, Flüsse, Geschwindigkeiten oder Platzbedarf dienen.

### Arbeiten zur Validierung der taktischen Ebene

Ein Schritt in Richtung Validierung der taktischen Ebene gehen Brogan & Johnson (2003). Hier werden die zurückgelegten Pfade von Fußgängern näher untersucht. Allerdings beziehen sich ihre Untersuchungen auf die Pfade der Fußgänger innerhalb von Gängen, die einsehbar sind – also auf kleinskaliger Ebene. Bei diesen Versuchen wurde bestätigt, dass Fußgänger langsamer um Ecken gehen und einen gewissen Abstand zu Wänden einhalten. Dazu wurde eine Metrik zum

Vergleich von Fußgängerrouen vorgeschlagen und verwendet, um die Pfade der Fußgänger quantifizierbar zu machen.

Graessle & Kretz (2011) führten ein Experiment zur Wegefindung von Fußgängern durch. Es wurden die Pfade der Fußgänger auf dem Hin- und Rückweg zu einem Fußballstadion aufgezeichnet und analysiert. Hier ist zu Tage getreten, dass die Fußgänger fast alle den längeren Weg gewählt haben. Die Daten sind nicht verfügbar, so dass eine Validierung anhand dieser Ergebnisse nicht durchgeführt werden kann.

Zusammenfassend lässt sich sagen, dass vereinzelt experimentell Daten zur Validierung erhoben wurden, diese sich aber hauptsächlich auf die Lokomotionsebene beschränken. Experimente, die sich mit der taktischen Ebene auseinandersetzen, sind in der Literatur kaum vorhanden. Um diese Lücke zu schließen, wurde ein Experiment durchgeführt, um eigene Daten für die Validierung zu erhalten.

### 7.3 Validierung der taktischen Ebene

Zur Validierung der taktischen Ebene werden Daten benötigt, die das menschliche Verhalten bei der Wegefindung abbilden. Dies bedeutet, dass ein experimentelles Setup Bereiche einschließen muss, die von der Position eines Fußgängers nicht einsehbar sind. Zudem sollte die Umgebung, in der sich die Probanden bewegen, so komplex sein, dass der kürzeste Weg zum Ziel nicht von Anfang an erkennbar ist. Ein weiteres Kriterium ist die Ortskenntnis. Wenn die Teilnehmer ausschließlich ortsfremd sind, so können keine Rückschlüsse auf Wegewahl an bekannten Orten gezogen werden. Auf Grundlage dieser Randbedingungen wurde das im Folgenden beschriebene Experiment aufgesetzt.

#### 7.3.1 Aufbau des Experiments<sup>1</sup>

Das Experiment wurde an der Technischen Universität München mit Studierenden des ersten Semesters durchgeführt. Die Aufgabe der Studierenden war folgende: In Kleingruppen von zwei bis vier Personen sollten die Teilnehmer von vier unterschiedlichen Straßenecken des Hauptgebäudes der Technischen Universität München zu zwei bekannten Zielen in der Münchener Innenstadt laufen und ihren zurückgelegten Weg dokumentieren. Die Ziele wurden so gewählt, dass die Straßenführung dorthin verwinkelt ist und beide Ziele ungefähr die gleiche Entfernung vom Startpunkt aus haben. Der Ausschnitt des Stadtplans mit den Zielen und Startpunkten ist in Abbildung 7.2 dargestellt. Die Distanz – also der kürzeste Weg – betrug für Ziel 1 (Hofbräuhaus) zwischen 1.8 und 2.2 Kilometer und für Ziel 2 (Lehel) zwischen 2.0 und 2.2 Kilometer. Eine Übersicht der Entfernungen von den einzelnen Startpunkten zu den Zielen ist in Tabelle 7.1 gegeben.

---

<sup>1</sup>Die Beschreibung des Experiments sowie erste Auswertungen sind auch in Kneidl & Borrmann (2011) zu finden



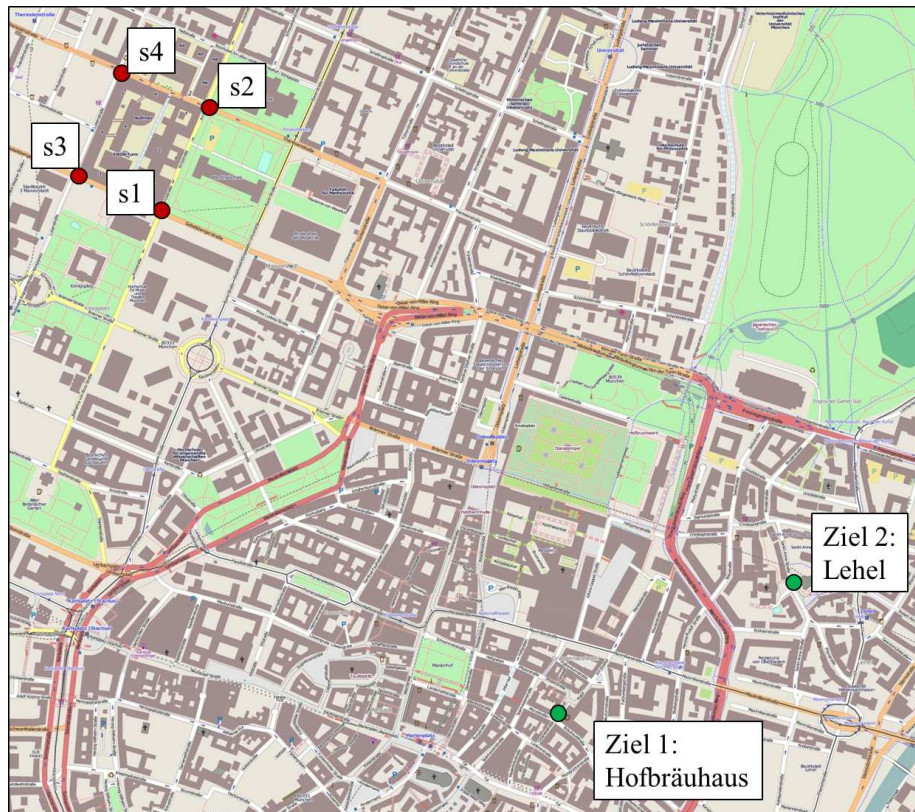


Abbildung 7.2: Übersicht über die vier Startpunkte und die zwei Ziele

Die Teilnehmer wurden in zwei Gruppen unterteilt. Die erste Gruppe (G1) sollte zu ihrem Ziel (Ziel 1) ohne weitere Hilfsmittel laufen, die zweite Gruppe (G2) bekam einen Stadtplan zur Orientierung ausgehändigt, um zu Ziel 2 zu gelangen.

Auf dem Weg zu ihrem jeweiligen Ziel dokumentierten die Teilnehmer jede Straßenkreuzung, die sie passiert haben.

Am Start wurde ein Stadtplan aufgehängt, auf den die Teilnehmer einen Blick werfen konnten, um so eine ungefähre Orientierung zu bekommen.

Nach ihrer Rückkehr zur Universität füllten alle Teilnehmer einen Fragebogen zur Wegefindung aus. Zudem sollten sie anhand der Aufzeichnungen die gelaufene Route in einem Stadtplan rekonstruieren.

Der Fragebogen beinhaltet Fragen zur räumlichen Orientierung und zur Routenwahl sowie zur Ortskenntnis der Teilnehmer.

### 7.3.2 Ergebnisse des Experiments

Insgesamt nahmen 154 Studierende an dem Experiment teil, davon sind 92 Teilnehmer der Gruppe G1, und 62 Teilnehmer der Gruppe G2 zugeordnet. 31.37 % der Teilnehmer sind weiblich, 66.76 % männlich und 1.96 % machten keine Anga-

Start	Entfernung Ziel 1 [km]	Entfernung Ziel 2 [km]
1	1.8	1.84
2	2.0	2.1
3	2.0	2.0
4	2.2	2.2

Tabelle 7.1: Entfernungen von den einzelnen Startpunkte zu den Zielen

Gruppengröße	G1	G2	Gesamt
1	2	1	3
2	9	4	13
3	9	7	16
4	10	8	18
5	1	0	1
<b>Gesamt</b>	31	20	51

Tabelle 7.2: Gruppengrößenverteilung

be. Die Probanden waren zwischen 19 und 21 Jahre alt. Die Gruppengrößenverteilung für G1 und G2 ist in Tabelle 7.2 aufgelistet. Insgesamt bildeten sich 51 Gruppen, 31 Gruppen waren Ziel 1 und 20 Gruppen Ziel 2 zugeordnet.

Die gelaufenen Wege sind in Abbildung 7.3 und Abbildung 7.4 dargestellt. Es ist zu beobachten, dass vor allem zu Ziel 1 eine große Bandbreite an Wegen gefunden wurde. Bei Ziel 2 klaffen die Wege nicht so weit auseinander, was zum Einen an der Lage des Ziels in Relation zu den Startpunkten, aber vor allem auch durch das zur Verfügung stehende Kartenmaterial zu begründen ist. Die Entfernungen der gelaufenen Wege sind in den Diagrammen 7.3 und 7.4 dargestellt.

Es ist zu beobachten, dass die Längen der Wege sehr unterschiedlich sind. Von fast jedem Startknoten wurde der kürzeste Weg von mindestens einer Gruppe gefunden. Auffällig ist, dass die Wege von Startpunkt 1, 3 und 4 in Richtung Ziel 2 bis zu 700 Meter länger sind. Dies ist aus dem Grund interessant, da die Probanden eine Karte zur Verfügung hatten. Daraus kann geschlossen werden, dass eine Karte zwar „Irrwege“, wie z.B. in die falsche Richtung laufen, ausschließt, die Probanden jedoch nicht den kürzesten Weg aus der Karte herauslesen konnten. Dies wurde im Fragebogen genauer untersucht. Die Probanden wurden gebeten, ihre gewählte Route hinsichtlich unterschiedlicher Kriterien zu bewerten. Diese

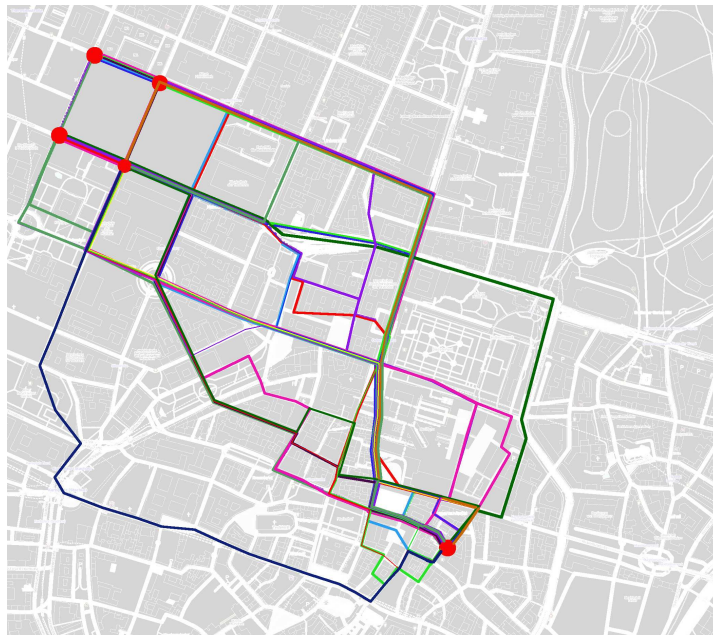


Abbildung 7.3: Die gelaufenen Wege zu Ziel 1: Jede Farbe steht für eine Route einer Gruppe

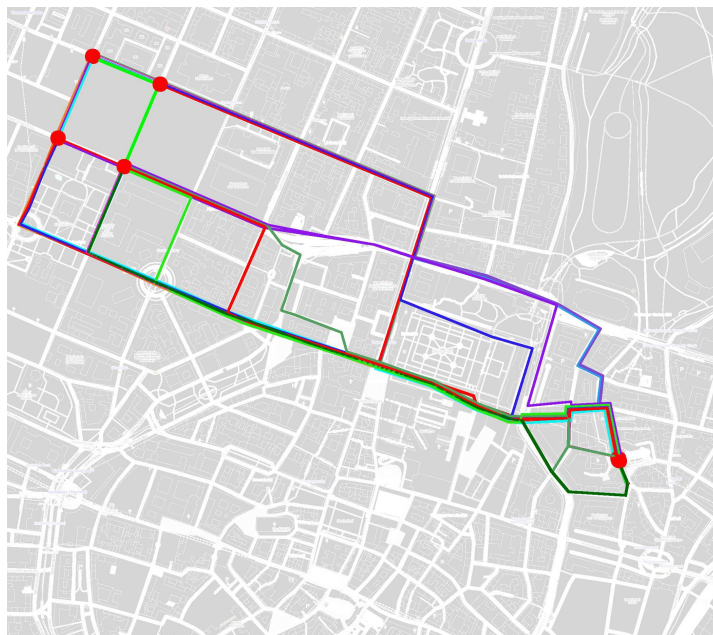


Abbildung 7.4: Die gelaufenen Wege zu Ziel 2: Jede Farbe steht für eine Route einer Gruppe

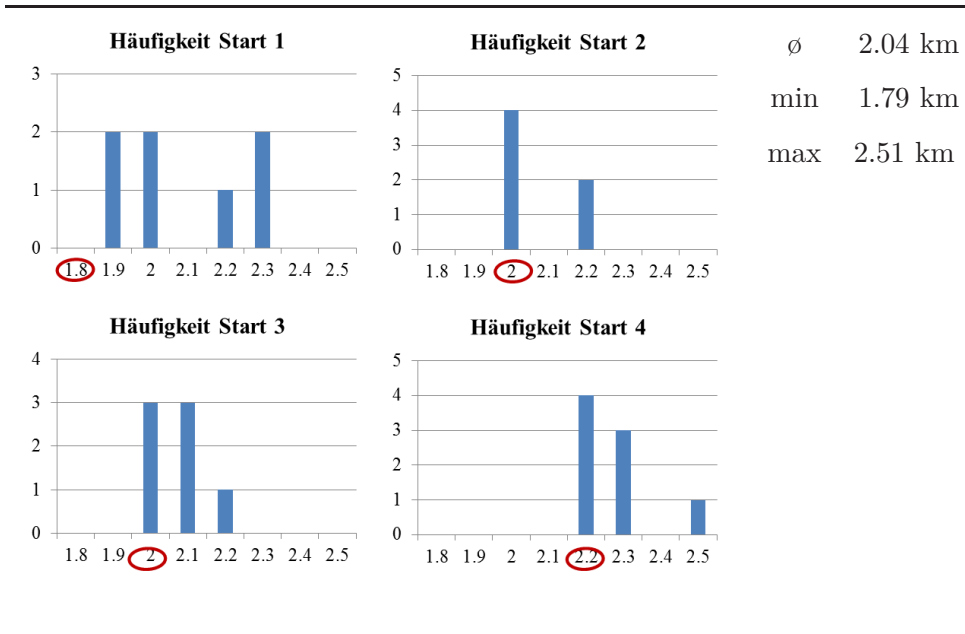


Tabelle 7.3: Weglängen zu Ziel 1. Die rot umkreisten Zahlen entsprechen dem kürzesten Weg.

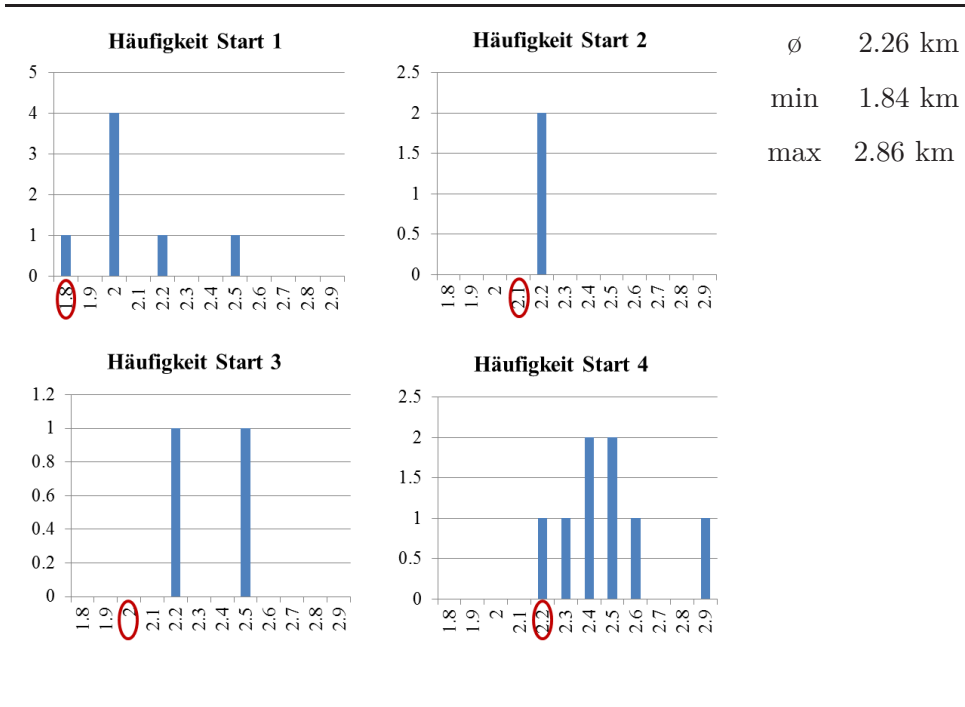


Tabelle 7.4: Weglängen zu Ziel 2. Die rot umkreisten Zahlen entsprechen dem kürzesten Weg.

Kriterien wurden in Anlehnung an das Experiment von Golledge (1995) gewählt. In Tabelle 7.5 sind die Antworten auf die Frage „Welche der folgenden Eigenschaften beschreibt am besten Ihre Routenwahl?“ zusammengefasst. Es ist interessant, dass fast 90 % aller Teilnehmer den von ihnen gewählten Weg als den kürzesten Weg bezeichnet haben. Ein Blick auf die Karte belegt das Gegenteil. Damit kann die von Golledge (1999a) aufgestellte These, dass Menschen Entfernungen nicht so gut einschätzen können wie topologische Zusammenhänge, untermauert werden.

Ein weiterer Aspekt ist die Übereinstimmung der Angaben im Fragebogen mit dem visuellen Bild der Wege: So gaben nur 26.80 % Prozent an, dass ihr Weg aus vielen Richtungswechseln besteht. Dies deckt sich mit dem visuellen Bild der Routen. Hingegen sagten 73 % genau das Gegenteil, dies ist ebenfalls aus den Ergebnissen ersichtlich. Eine interessante Erkenntnis ist die Tatsache, dass 77.12 % angaben, dass die Route eher aus Hauptstraßen besteht. Da im Allgemeinen Hauptstraßen lange Strecken geradeaus führen, deckt sich dies mit der Aussage, dass die Teilnehmer lange Strecken geradeaus bevorzugt gewählt haben.

### 7.3.3 Simulationsaufbau

Um die aus dem Experiment resultierenden Routen mit Routen aus den entwickelten Algorithmen vergleichen zu können, wird das Simulationsszenario wie folgt aufgebaut: Ein Ausschnitt aus OpenStreetMap-Daten<sup>2</sup> dient als Grundlage zur Generierung des Szenarios. Die Geometriedaten werden als xml-Datei von OpenstreetMap ausgelesen und mithilfe eines Parsers in eine Szenariodatei umgewandelt (Wildgruber, 2011). Diese Datei wird manuell verfeinert: Es werden Straßenbrücken gelöscht, Parks als Regionen mit verlangsamter Geschwindigkeit ausgezeichnet und um Häuserblocks Wände gezogen, sofern die Genauigkeit in OpenstreetMap nicht ausreichend ist. Das aus der OpenStreetMap abgeleitete Szenario ist in Abbildung 7.5 dargestellt. Zusätzlich werden die vier Startpunkte als Quellen definiert sowie die beiden Ziele hinzugefügt. Der Graph wird zur besseren Vergleichbarkeit manuell erstellt. An jeder Straßenkreuzung wird ein Knoten platziert und zwischen zwei benachbarten Straßenkreuzungen wird eine Kante eingefügt. Der daraus resultierende Graph ist in Abbildung 7.6 dargestellt.

#### 7.3.3.1 Konfiguration der Simulation

Die Simulation wird mit drei unterschiedlichen Algorithmen durchgeführt: Dem Fastest Path (siehe Unterabschnitt 6.2.1), dem Weg möglichst nahe entlang der Luftlinie, also Heuristic Path (siehe Unterabschnitt 6.2.2) sowie dem SALL-Algorithmus (siehe Unterabschnitt 6.2.4). Diese drei Algorithmen sollen die unterschiedliche Ortskenntnis der Teilnehmer abbilden. Da hier eine Normalsituation vorliegt, werden die anderen, für den Evakuierungsfall in Kapitel 6 vorgestellten Algorithmen nicht eingesetzt.

---

<sup>2</sup>[www.osm.org](http://www.osm.org)

Charakteristik des Wegs	trifft zu - trifft eher zu	trifft nicht zu - trifft über- haupt nicht zu	Gesamt
	[ % ]	[ % ]	[ % ]
Route ist der kürzeste Weg	89.54	10.46	100
Route besteht aus wenigen Richtungswechseln	73.20	25.49	98.69
Route besteht aus vielen Richtungswechseln	26.80	72.55	99.35
Route orientiert sich an der Luftlinie zum Ziel	71.24	28.11	99.35
Route beginnt mit einer kurzen Strecke bis zum ersten Richtungswechsel	33.33	65.66	98.69
Route beginnt mit einer langen Strecke bis zum ersten Richtungswechsel	62.09	36.03	98.69
Route besteht eher aus Hauptstraßen	77.12	22.23	99.35
Route enthält Umwege	11.76	88.24	100

Tabelle 7.5: Charakteristik der gewählten Routen

Die Algorithmen werden wie folgt konfiguriert: In diesem Szenario treten keine Staus auf, da es sich um ein sehr großes städtisches Gebiet handelt und nur eine geringe Anzahl von Fußgängern erzeugt wird. Daher entsprechen die Ergebnisse des Fastest Path Algorithmus dem kürzesten Weg. Der Algorithmus gibt die 10 schnellsten Wege aus, sofern die Gesamtlänge der gefundenen Wege sich nicht um mehr als 10 % unterscheidet. Dies ist eine realistische Annahme, da die gelaufenen Wege sich in der Gesamtlänge sehr wenig unterscheiden und eine korrekte Einschätzung der euklidischen Metrik von Personen nicht vorgenommen werden kann.

Die Simulation mit dem Heuristic Path Algorithmus wird wie in Unterunterabschnitt 6.2.2.1 konfiguriert. Alle Kantengewichte werden mit einem Fehlerfaktor zwischen 0.9 und 1.1 multipliziert.

Der SALL-Algorithmus wird wie folgt eingesetzt: Es werden an jedem Knoten jeweils die zwei ausgehenden Kanten mit den größten Kantengewichten betrachtet. Unterscheidet sich das Kantengewicht dieser Kanten um weniger als 5 %, so wird zufällig zwischen beiden Kanten gewählt.

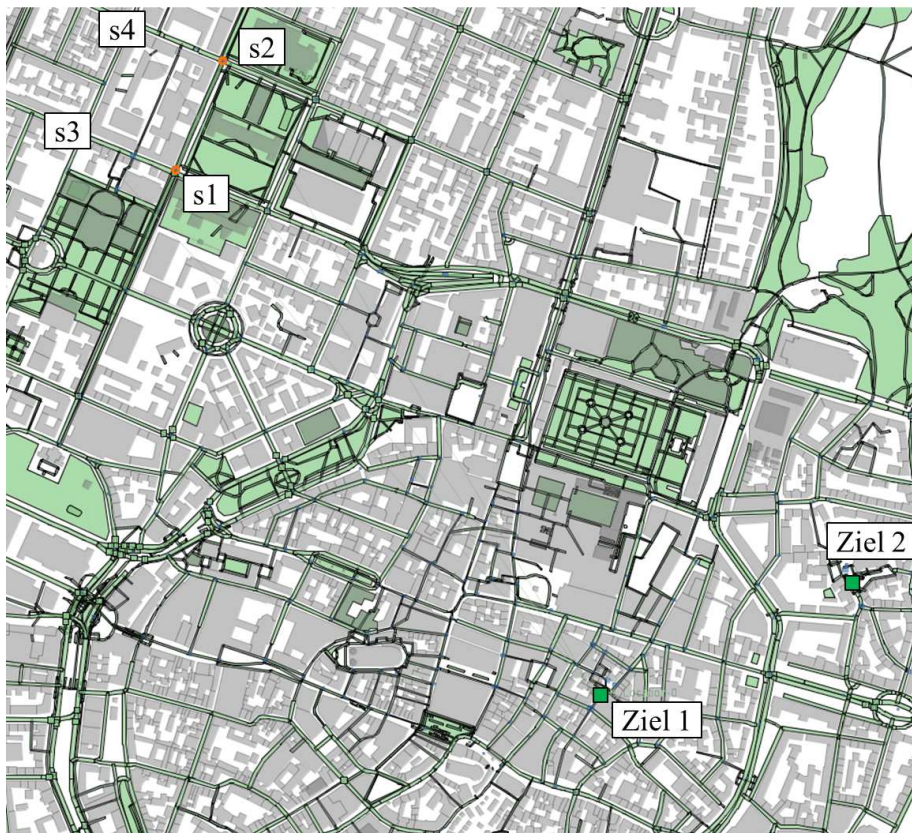


Abbildung 7.5: Simulationsszenario, erstellt aus dem Stadtplan von Abbildung 7.2

Pro Simulationslauf werden 100 Personen in jeder Quelle erzeugt, um eine größere Anzahl von Routen zu erhalten. Für den visuellen Vergleich werden alle drei Algorithmen zu gleichen Teilen in der Simulation gewichtet. Zur Ähnlichkeitsbetrachtung wird jeder Algorithmus getrennt simuliert. Für den Vergleich der relativen Häufigkeit fließen die Algorithmen entsprechend der Ergebnisse aus dem Vergleich der Ähnlichkeitsbetrachtung anteilig mit ein.

Die Simulation wird einmal ohne Landmarks und einmal mit dem manuellen Platzieren von Landmarks durchgeführt. Dabei werden die Landmarks aus der Charakteristik des Gebiets abgeleitet: Auf der Strecke zwischen den Startpunkten und dem Ziel liegen zwei bekannte Orte. Im westlichen Teil liegt ein für die Teilnehmer bekannter Platz, der Karolinenplatz, den diese auf dem Weg zur Universität überqueren. Dieser Platz dient als Landmark zur Orientierung. Fast die Hälfte aller Routen (15 von 31) zu Ziel 1 verlaufen über diesen Platz. Auch zu Ziel 2 führen 9 von 20 Routen über den Karolinenplatz.

Ein weiterer allgemein bekannter Platz ist der Odeonsplatz, der auf der östlichen Seite liegt. 24 der insgesamt 31 Routen zu Ziel 1 sowie 16 der 20 Routen in Richtung Ziel 2 verlaufen über den Odeonsplatz. Folglich wird auch dieser Platz als Landmark modelliert.

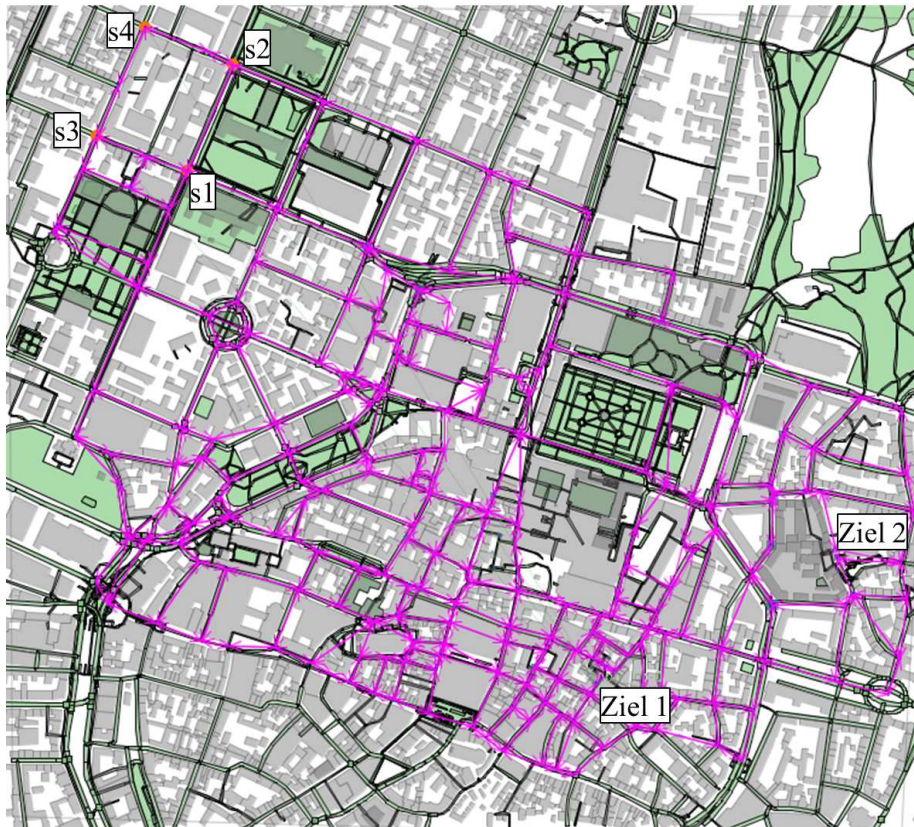


Abbildung 7.6: Navigationsgraph als Grundlage der Wegfindungsalgorithmen

### 7.3.4 Definition der Vergleichsmetriken

Die Routen werden auf drei unterschiedliche Arten mit den Simulationsergebnissen verglichen:

- Visueller Vergleich
- Ähnlichkeitsbetrachtung pro Route
- Relative Häufigkeit der traversierten Kanten

**Visueller Vergleich** – Mit dem visuellen Vergleich wird überprüft, ob das Gesamtbild der Routen aus dem Experiment von der Simulation reproduziert wird.

**Ähnlichkeitsbetrachtung pro Route** – Die gelaufenen Routen werden in eine Liste von Knoten übertragen, die im Graphen der Simulation enthalten sind. So liegen sie im gleichen Format wie die von der Simulation berechneten Routen vor und können miteinander verglichen werden. Für die Ähnlichkeitsbetrachtung werden Routenpaare gebildet. Ein Routenpaar besteht jeweils aus einer real gelaufenen Route und einer von der Simulation erstellten Route. Jede der aus der Simulation berechnete Route wird mit jeder real gelaufenen Route verglichen. Das Routenpaar mit der besten Übereinstimmung fließt in die Vergleiche in Unterunterabschnitt 7.3.5.2 mit ein.

Dieses Vorgehen kann auch mit Kalibrierung gleichgesetzt werden: Aus einer Viel-



zahl von Routenpaaren werden diejenigen Paare mit der besten Übereinstimmung gewählt. Abhängig von der Anzahl der simulierten Routen eines bestimmten Algorithmus aus den Routenpaaren mit den größten Werten können die Gewichtungen der einzelnen Algorithmen abgeleitet werden.

Sofern beide Routen des Routenpaares nicht komplett identisch sind, spannen sie sogenannte Fehlerpolygone auf, die aus genau den Kanten bestehen, in denen sich zwei Routen unterscheiden. Diese Polygone werden zwischen dem letzten gemeinsamen Knoten und dem ersten erneut gemeinsamen Knoten der zu vergleichenden Routen aufgespannt. Im ungünstigsten Fall, wenn zwei Routen komplett unterschiedlich verlaufen, entspricht der letzte gemeinsame Knoten dem Startknoten und der erste erneut gemeinsame Knoten dem Zielknoten. Ein Beispiel ist in Abbildung 7.7 dargestellt. Hier ist das Fehlerpolygon durch die Kanten  $e_1$  bis  $e_8$  definiert. Für jedes zu vergleichende Routenpaar werden folgende Kenngrößen

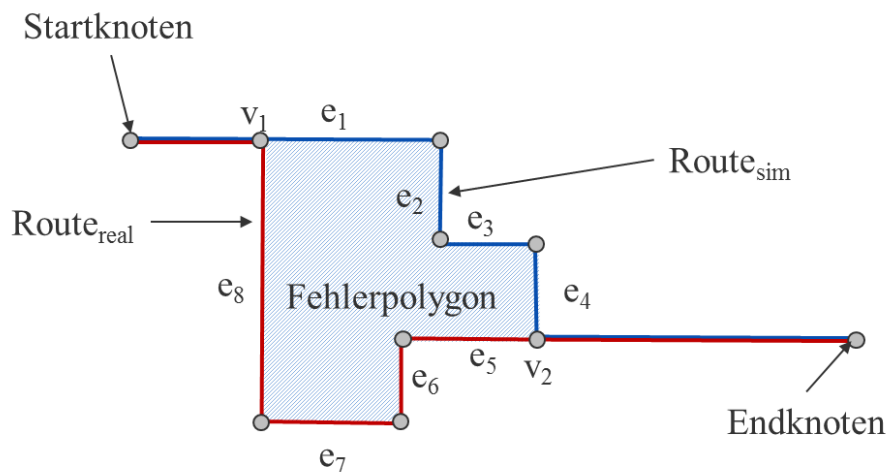


Abbildung 7.7: Beispiel für ein Fehlerpolygon, das von einem Routenpaar aufgespannt wird

berechnet:

- Relative Länge der Routen
- Ratio der übereinstimmenden Streckenlänge  $ratio_{length}$
- Turning Angle der Fehlerpolygone

*Relative Länge der Routen  $rel_{length}$* : Hier werden die Gesamtlängen eines Routenpaares in Relation zueinander gesetzt. Diese Größe gibt Auskunft darüber, wie ähnlich sich Routen bezogen auf die Gesamtlänge sind. Damit der Wert immer zwischen 0 und 1 liegt, wird die Route mit der kürzeren Länge in Relation zur längeren Route gesetzt. Bei Übereinstimmung ergibt sich der Wert 1.

*Ratio der übereinstimmenden Streckenlänge  $ratio_{length}$* : Die Ratio der übereinstimmenden Strecken zur Gesamtlänge ist das wichtigste Maß für die Ähnlichkeit der Routen. Sie wird berechnet, indem die Streckenlänge innerhalb der Fehlerpolygone von der Gesamtlänge der Strecke abgezogen werden. Dabei werden die

Kanten des Fehlerpolygons, die auf der einen Route liegen, durch die Gesamtlänge dieser Route geteilt. Um den Anteil gleicher Strecke zu ermitteln, wird dieser Wert von eins abgezogen. Mit der anderen Route wird analog verfahren. Aus den beiden daraus resultierenden Werten wird das Minimum gewählt. Formal wird der Wert wie folgt berechnet:

$$\begin{aligned} ratio_{real} &= 1 - \frac{dist_{errPol_{real}}}{dist_{completeRoute_{real}}} \\ ratio_{sim} &= 1 - \frac{dist_{errPol_{sim}}}{dist_{completeRoute_{sim}}} \\ ratio_{length} &= \min(ratio_{real}, ratio_{sim}) \end{aligned}$$

Im Beispiel aus Abbildung 7.7 wird die Länge für die Route<sub>real</sub> berechnet, indem die Längen der Kanten  $e_5$  bis  $e_8$  addiert und durch die Gesamtlänge der Route geteilt werden. Um den gemeinsamen Anteil an der Gesamtlänge zu ermitteln, wird dieser Teil von eins abgezogen. Gleiches wird für die simulierte Route berechnet. Das Minimum der beiden berechneten Größen bestimmt den Wert für die Ratio.

Auch hier ergibt sich ein Wert zwischen 0 und 1. Stimmen zwei Routen komplett überein, so nimmt der Wert 1 an. Falls sich die Routen nur im Startknoten und Endknoten treffen, resultiert daraus der Wert 0.

*Turning Angle (TA) der Fehlerpolygone:* Die Turning Angle Funktion (in der Literatur auch als  $\psi$  - s Kurve bezeichnet, (Ballard & Brown, 1982, S.237)) setzt die Strecken ins Verhältnis zu den Winkeln eines Polygons. Dazu wird die Länge der Kanten entlang der x-Achse und die Winkel zwischen zwei benachbarten Kanten entlang der y-Achse aufgetragen. Die daraus resultierende Funktion ist eine Stufenfunktion, aus der die Kantenlängen im Verhältnis zu den Winkeln des Polygons abzulesen sind (siehe Abbildung 7.8).

Arkin *et al.* (1991) verwendet diese Turning Angle Function als eine Metrik zum Vergleich von Polygonen im Bereich der Bilderkennung.

Hier werden die Kanten und Winkel untersucht, die innerhalb des Fehlerpolygons existieren. Verglichen wird nicht das komplette Polygon, sondern die Winkel und Längen der Kanten aus der real gelaufenen Route mit den Winkeln und Längen der Kanten aus der simulierten Route. Das Fehlerpolygon wird also in zwei Polygonzüge aufgeteilt; der eine Polygonzug besteht aus den Kanten der real gelaufenen Route (im Beispiel aus Abbildung 7.7  $e_5$  bis  $e_8$ ), der andere Polygonzug umfasst die Kanten der simulierten Route (im Beispiel aus Abbildung 7.7  $e_1$  bis  $e_4$ ). Die daraus resultierende Turning Angle Funktion der beiden Kantenzüge ist in Abbildung 7.8 dargestellt. Da die Kantenzüge der im Experiment gelaufenen Route und der mit der Simulation berechneten Route unterschiedlich lang sein können, wird das Maximum der Summe aller realen bzw. simulierten Kanten innerhalb eines Polygons auf 1 gesetzt. Die andere Route wird so skaliert, dass die Summe dieser Kantenzüge ebenfalls 1 ergibt. Die Länge kann hier vernachlässigt werden, da die Länge der Kantenzüge bereits in der Teilmetrik  $ratio_{length}$  mit einbezogen wird. Die beiden Kurven werden voneinander abgezogen, so dass der daraus resultierende Wert die Winkeldifferenz der beiden Kantenzüge beschreibt.

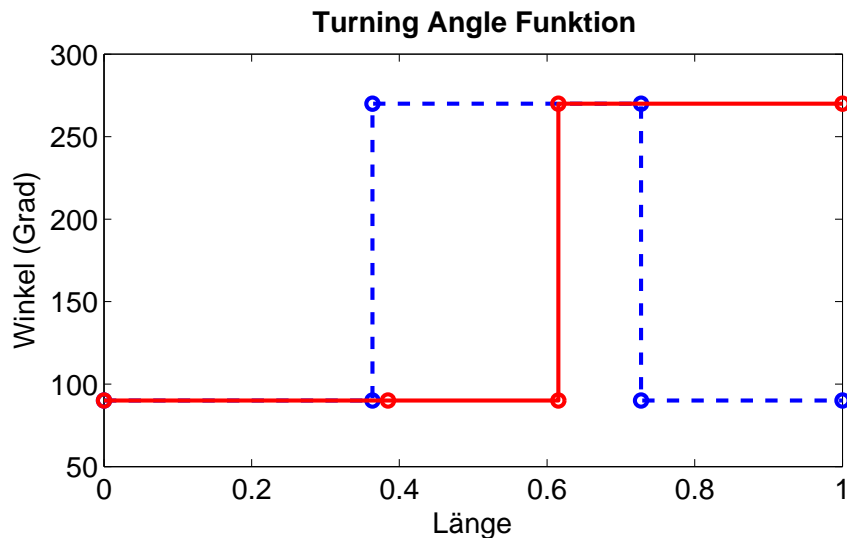


Abbildung 7.8: Turning Angle Funktion für das Beispiel aus Abbildung 7.7

Spannen zwei Routen mehrere Fehlerpolygone auf, so werden die Werte addiert. Um den Wert vergleichbar zu machen, wird das Maximum der Differenzen aller Routenpaare herangezogen, und alle anderen Werte anhand dieses Wertes skaliert. So liegt der Wert auch hier zwischen 0 (sehr unterschiedliche Anzahl der Richtungswechsel mit unterschiedlich großen Winkeln) und 1 (komplette Symmetrie). Um die Größe des Fehlerpolygons miteinzubeziehen (die Gesamtlänge aller Kanten), wird der Wert ermittelt, indem die Ratio der übereinstimmenden Weglänge  $ratio_{length}$  miteinbezogen wird. Der Wert des Turning Angle fließt nur für den Teil unterschiedlicher Weglänge mit ein, für den Rest der Strecke nimmt er den Wert 1 (vollkommene Übereinstimmung) an. Formal ausgedrückt:

$$TA_{gesamt} = (1 - ratio_{length}) \cdot TA + ratio_{length} \cdot 1$$

Die drei Einzelwerte werden in einer Gesamtsumme zusammengefasst. Die Ratio übereinstimmender Weglänge wird höher gewichtet, da dies der wichtigste Wert für die Ähnlichkeitsbetrachtung darstellt. Der Turning Angle dagegen wird weniger hoch gewichtet, da sich dieser nur auf die Bewertung der Fehlerpolygone bezieht.

$$similarity_{real,sim} = rel_{length} + 1.5 \cdot ratio_{length} + 0.5 \cdot TA_{gesamt}$$

Der Maximalwert liegt bei 3, bei völlig unterschiedlichen Routen nähert sich der Wert 0 an. Diese Metrik dient als Grundlage für die Ähnlichkeitsbetrachtung der Routen.

**Relative Häufigkeit der traversierten Kanten** – Die letzte Art von Vergleich überprüft die Häufigkeit der traversierten Kanten. Dies dient dazu, einen Überblick zu bekommen, wie häufig gewisse Teilstrecken während des Experiments entlang gelaufen wurden. Daraus kann die Wichtigkeit der Kanten ermittelt werden. Es wird gezählt, in wie vielen Wegen eine Kante enthalten ist. Dies wird

einmal für die Wege des Experiments und einmal für die Wege der Simulation gezählt. Der jeweils resultierende Wert wird ins Verhältnis zur Anzahl aller Wege gesetzt. Damit ergibt sich wiederum ein Wert zwischen 0 (in keinem Weg enthalten) und 1 (in allen Wegen enthalten). Die beiden daraus resultierenden Werte, einmal für die gelaufenen Wege und einmal für die berechneten Wege, werden miteinander verglichen. Ziel ist, dass beide Werte in derselben Größenordnung liegen.

### 7.3.5 Ergebnisse des Vergleichs

Im Folgenden werden die Ergebnisse der drei Vergleiche jeweils getrennt für Ziel 1 und für Ziel 2 vorgestellt.

#### 7.3.5.1 Visueller Vergleich

In den folgenden Abschnitten werden die im Experiment gelaufenen Routen mit den von den Simulationsläufen berechneten Routen visuell verglichen.

##### Visueller Vergleich der Routen zu Ziel 1

Die erste Gruppe lief in Richtung Ziel 1 ohne Zuhilfenahme einer Karte. Die Ergebnisse weisen eine hohe Variation auf und verlaufen in mehrere Richtungen (Abbildung 7.9). Es bilden sich keine eindeutigen Haupttrouten aus, und keine zwei Routen gleichen einander.

Die mit der Simulation berechneten Routen sind in Abbildung 7.10a und 7.10b dargestellt. Die Routen weisen ebenfalls eine hohe Variabilität auf und reproduzieren alle Richtungen der im Experiment gelaufenen Routen, sowohl ohne Landmarks als auch unter Miteinbeziehung von Landmarks. Für alle vier Startpunkte werden mehr Routen von der Simulation gefunden als die tatsächlich gelaufenen Routen. Dies resultiert daraus, dass für jeden Startpunkt 100 Personen generiert werden. Innerhalb eines Simulationslaufs werden mehr Routen berechnet als im Experiment gelaufen wurden.

Um die gelaufenen Wege besser visuell mit den berechneten Wegen vergleichen zu können, sind die Wege nach Startpunkt aufgeschlüsselt in den Abbildungen 7.11 bis 7.14 gegenübergestellt.

Von Start 1 aus führt eine Route (in Orange) zunächst weg vom Ziel. Dies ist in den Ergebnissen der Simulation nicht zu sehen. Auch die Route in Mint macht einen Bogen weg vom Ziel, was von der Simulation so nicht reproduziert wird. Generell sind die von der Simulation ohne Landmarks berechneten Routen mehr nach Osten ausgeprägt als die real gelaufenen Routen. Dennoch ist festzuhalten, dass die Hauptrichtungen nach Westen und nach Süden übereinstimmen. Auch die südlichste Route (in Blau) wird von beiden Simulationsläufen reproduziert. Für die Simulation mit Landmarks ändert sich das visuelle Bild nicht erheblich. Festzustellen ist, dass mehr Routen nun über das Landmark Karolinenplatz führen, was dem Bild der gelaufenen Routen mehr entspricht.

Die im Experiment von Start 2 gelaufenen Routen sind durch lange gerade Strecken und wenige Richtungswechsel charakterisiert. In der Simulation oh-

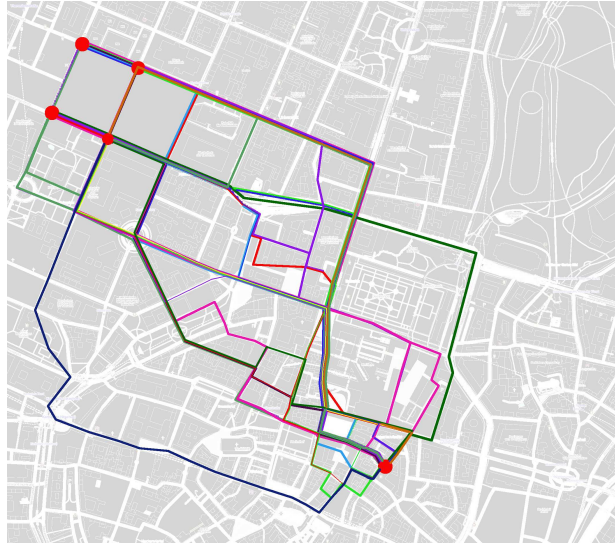
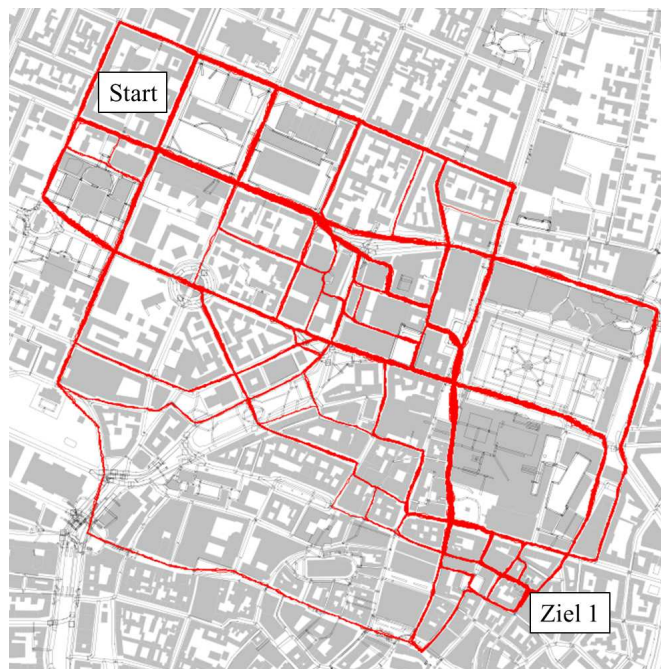
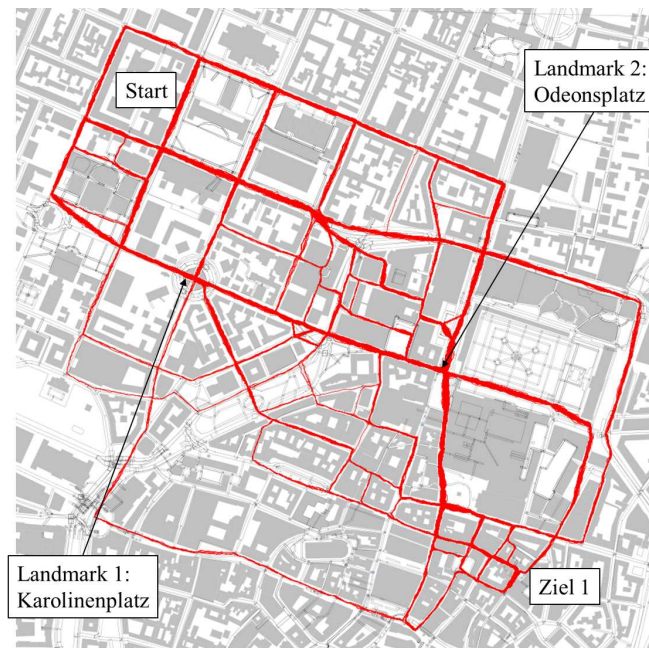


Abbildung 7.9: gelaufene Wege der Teilnehmer zu Ziel 1



(a) Ergebnis der Simulation ohne Landmarks



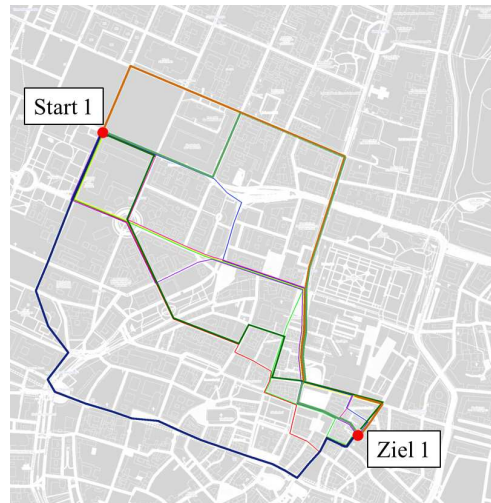
(b) Ergebnis der Simulation unter Miteinbeziehung von Landmarks

Abbildung 7.10: Visueller Vergleich der Abbildung 7.9 im Experiment gelaufenen Routen mit den Simulationsergebnissen

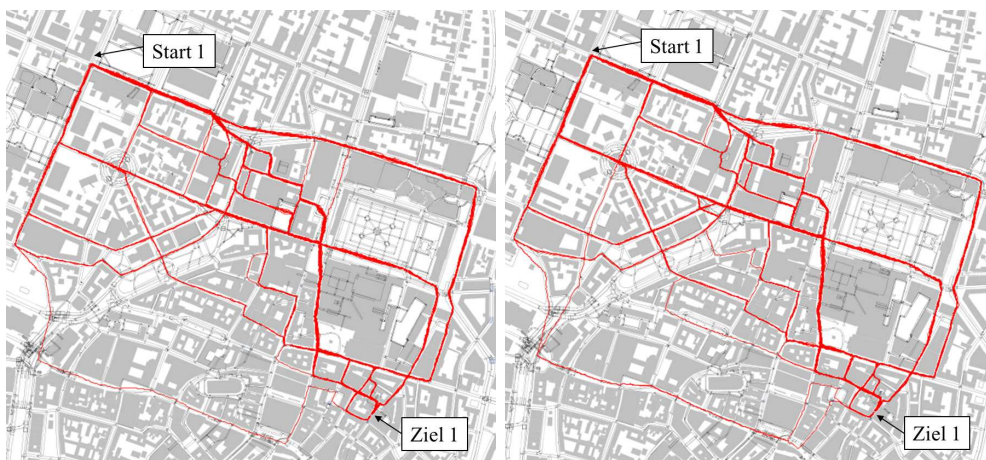
ne Landmarks existieren zusätzlich zu den gelaufenen Hauptachsen auch viele Routen, die aus vielen Richtungswechseln bestehen, vor allem im Mittelteil der Strecke. Dies resultiert aus den Algorithmen „Fastest Path“ und „Heuristic Path“, die möglichst entlang der Luftlinie führen bzw. den schnellsten Weg suchen. Auch hier unterscheidet sich das Bild der Simulation mit Landmarks nicht erheblich. Tendenziell verlaufen mehr Routen über den Karolinenplatz als bei der Simulation ohne Landmarks.

Die Wege von Start 3 weisen ebenfalls gute Übereinstimmungen auf, sowohl für die Simulationsergebnisse ohne Landmarks als auch für die Simulationsergebnisse mit Landmarks. Von diesem Startpunkt aus existiert eine gelaufene Route, die sehr lange in Richtung Osten führt, bevor der Richtungswechsel Richtung Süden erfolgt. Beide Simulationsläufe reproduzieren diese Route. Die zwei vom Start ausgehenden Parallelachsen von Westen nach Osten sind in allen drei Bildern zu erkennen und auch der Mittelteil der Routen ähnelt sich. Hier sind die Personen im Experiment durch die verwinkelten Straßenzüge gelaufen. Bei der Simulation mit Landmarks fällt die sehr südlich verlaufene Route auf. Diese wurde von keiner der Gruppen im Experiment gewählt. Diese südliche Route ist vom SALL-Algorithmus erzeugt und verläuft über das Landmark Karolinenplatz.

Von Start 4 aus tendieren die Personen dazu, lange gerade Strecken zu laufen. Die süd-westlichen Wege können von der Simulation ohne Landmarks nicht reproduziert werden. Da das Landmark „Karolinenplatz“ weder auf Luftlinie zum

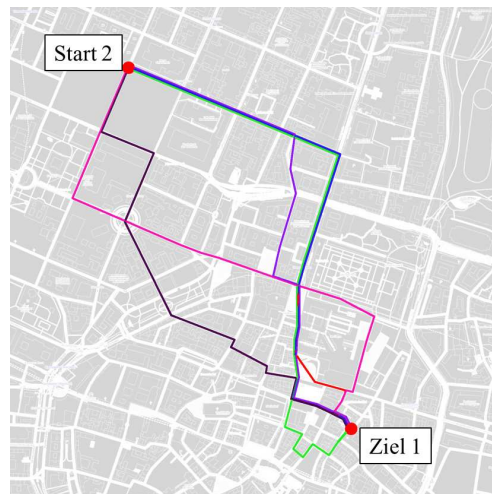


(a) Im Experiment gelaufene Routen

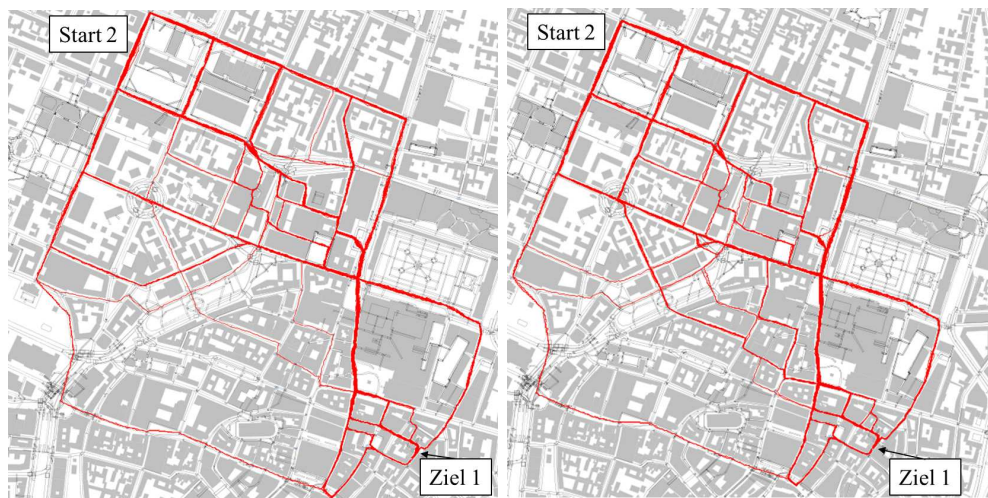


(b) Simulationsergebnis ohne Landmarks (c) Simulationsergebnis mit Landmarks

Abbildung 7.11: Visueller Vergleich der gelaufenen und simulierten Routen von Startpunkt 1 zu Ziel 1



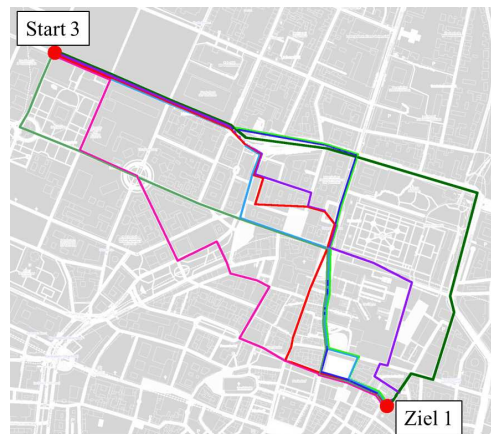
(a) Im Experiment gelaufene Routen



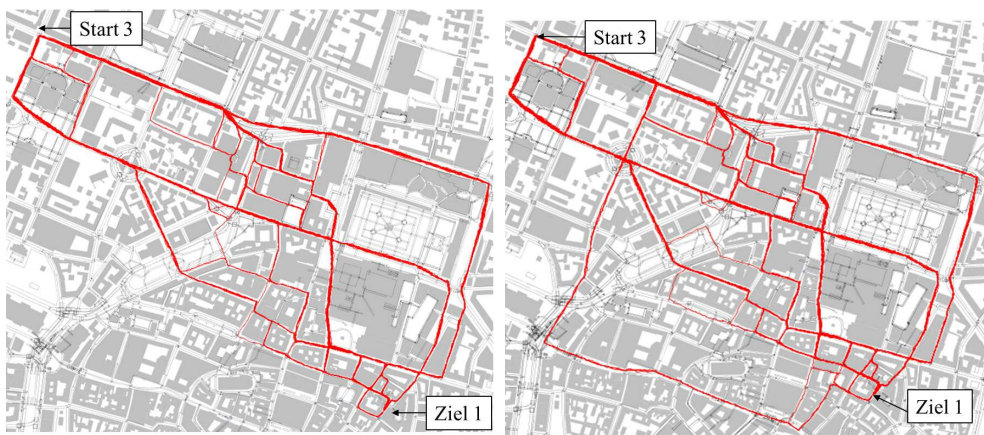
(b) Simulationsergebnis ohne Landmarks (c) Simulationsergebnis mit Landmarks

Abbildung 7.12: Visueller Vergleich der gelaufenen und simulierten Routen von Startpunkt 2 zu Ziel 1



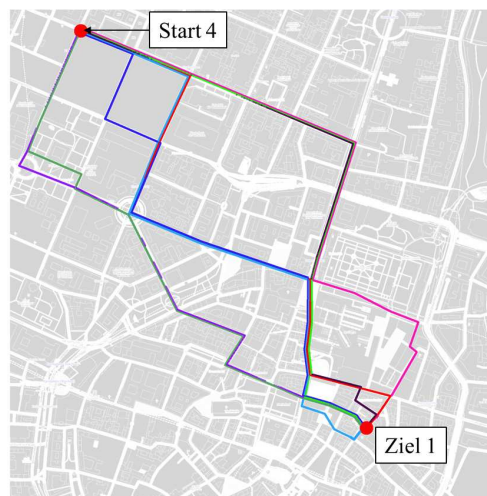


(a) Im Experiment gelaufene Routen

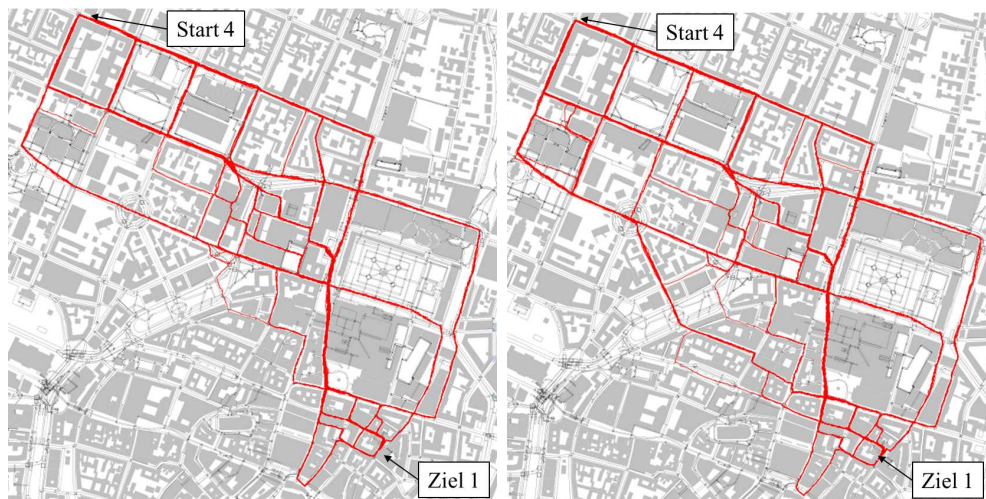


(b) Simulationsergebnis ohne Landmarks (c) Simulationsergebnis mit Landmarks

Abbildung 7.13: Visueller Vergleich der gelaufenen und simulierten Routen von Startpunkt 3 zu Ziel 1



(a) Im Experiment gelaufene Routen



(b) Simulationsergebnis ohne Landmarks (c) Simulationsergebnis mit Landmarks

Abbildung 7.14: Visueller Vergleich der gelaufenen sowie simulierten Routen von Startpunkt 4 zu Ziel 1

Ziel noch auf dem schnellsten Weg liegt, werden diese Wege nicht berechnet. Hier erkennt man deutlich die Verbesserung der Ergebnisse unter Miteinbeziehung von Landmarks. Damit werden auch die süd-westlichen Routen reproduziert.

Insgesamt lässt sich rein aus dem visuellen Vergleich schlussfolgern, dass von der Simulation die meisten gelaufenen Routen reproduziert werden können. Bei den Vergleichen der einzelnen Startpunkte weisen die Simulationsergebnisse ohne Miteinbeziehung von Landmarks sowie die Simulationsergebnisse unter Miteinbeziehung von Landmarks keine großen Unterschiede auf. Dies liegt daran, dass von den Startpunkten 1 bis 3 das Landmark Karolinenplatz sehr nahe der Luftlinie zum Ziel und das Landmark Odeonsplatz sehr nahe dem kürzesten Weg zum Ziel liegt. Anhand der von Startpunkt 4 gelaufenen Routen wird deutlich, dass die Personen im Experiment über das Landmark Karolinenplatz gelaufen sind,

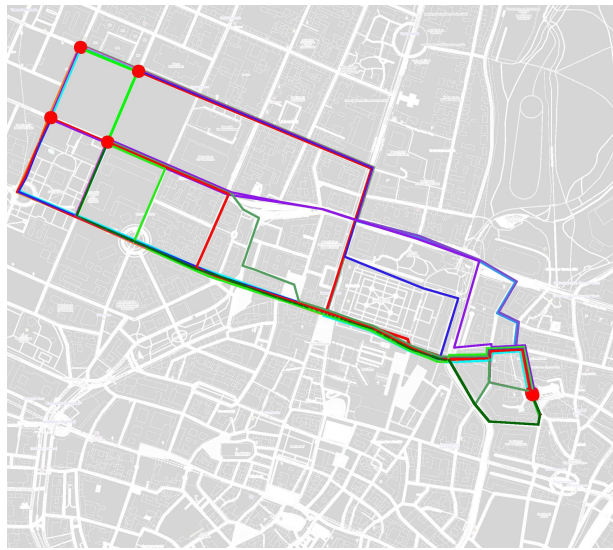


Abbildung 7.15: Im Experiment gelaufene Wege der Teilnehmer zu Ziel 2

obwohl es nicht auf der Luftlinie zum Ziel liegt. Landmarks spielen durchaus in der Wegefindung eine Rolle und die Simulation mit Landmarks liefert ein besseres visuelles Ergebnis. Ob auch die Werte der Ähnlichkeitsbetrachtung unter Einsatz von Landmarks verbessert werden, wird in Unterunterabschnitt 7.3.5.2 betrachtet.

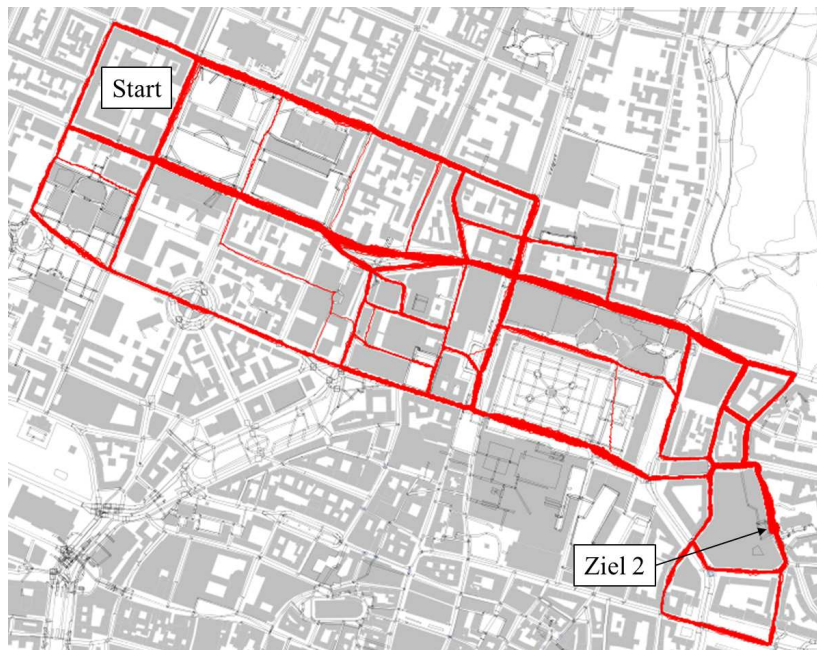
### Visueller Vergleich der Routen zu Ziel 2

Die zweite Gruppe bekam einen Stadtplan zur Orientierung. In der Umfrage gaben ein Drittel der Personen an, sich anhand der Karte orientiert zu haben. Die Ergebnisse der real gelaufenen Routen weisen daher keine so große Variabilität auf wie die Routen zu Ziel 1 (vgl. Abbildung 7.15).

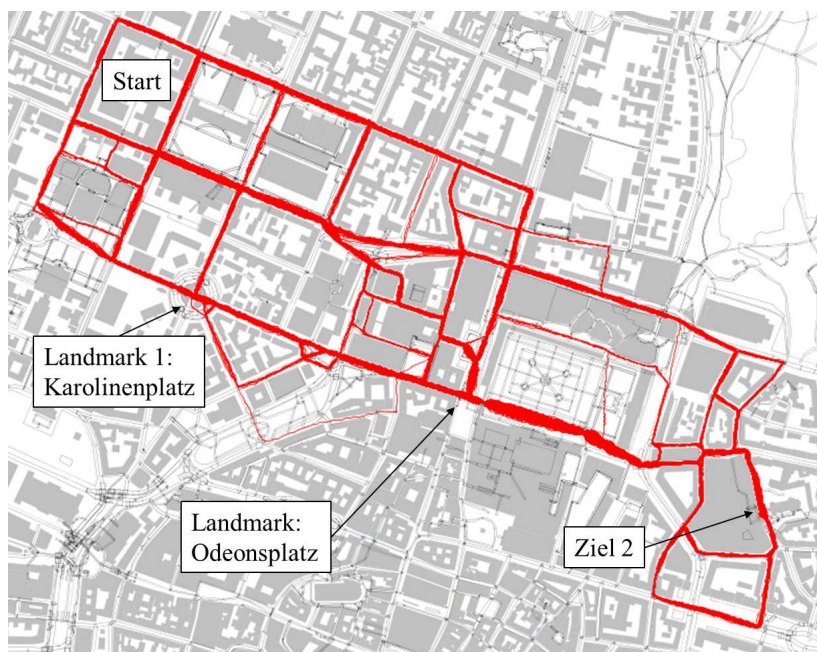
Die Ergebnisse des Simulationslaufs ohne Landmarks sind in Abbildung 7.16a, die Ergebnisse des Simulationslaufs unter Miteinbeziehung von Landmarks in Abbildung 7.16b dargestellt.

Allein aus dem visuellen Vergleich ist zu erkennen, dass die meist begangenen Teilstrecken von der Simulation ebenfalls reproduziert werden. Die drei gelaufenen Hauptachsen in West-Ost Richtung sind auch in den Simulationsergebnissen wiederzufinden. Interessant ist, dass nur sehr wenige Gruppen den kürzesten Weg gelaufen sind, trotz der Zuhilfenahme einer Karte. Die Ergebnisse der Simulation mit Landmarks zeigen zudem, dass nicht so viele Routen über die nord-östliche Ecke führen. In der Simulation mit Landmarks werden mehr Routen berechnet, die über diese nord-östliche Ecke führen. Dies entspricht eher dem Bild der gelaufenen Routen.

In den Abbildungen 7.17 bis 7.20 sind die gelaufenen Wege den Simulationsergebnissen mit und ohne Einsatz von Landmarks nach Start aufgeschlüsselt gegenübergestellt. Die Simulationsläufe berechnen weit mehr unterschiedliche Wege,

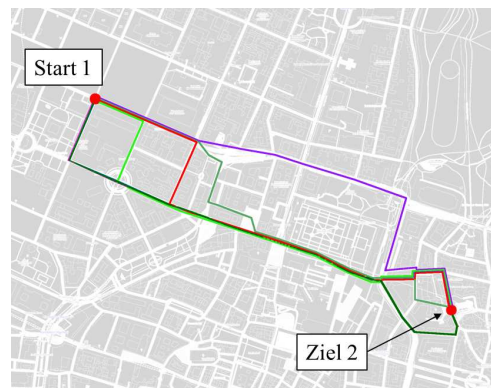


(a) Ergebnis der Simulation ohne Landmarks

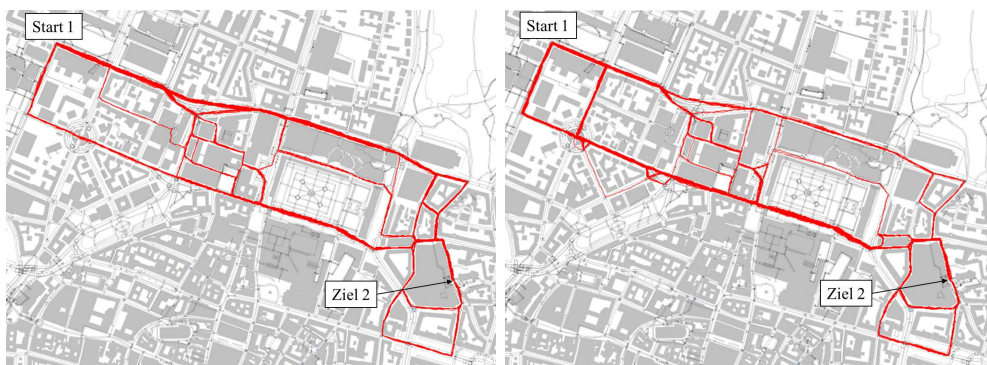


(b) Ergebnis der Simulation mit Landmarks

Abbildung 7.16: Visueller Vergleich der Abbildung 7.15 im Experiment gelaufenen Wege zu Ziel 1



(a) Im Experiment gelaufene Routen



(b) Simulationsergebnis ohne Landmarks (c) Simulationsergebnis mit Landmarks

Abbildung 7.17: Die real gelaufenen Routen sowie die simulierte Routen von Startpunkt 1 zu Ziel 2

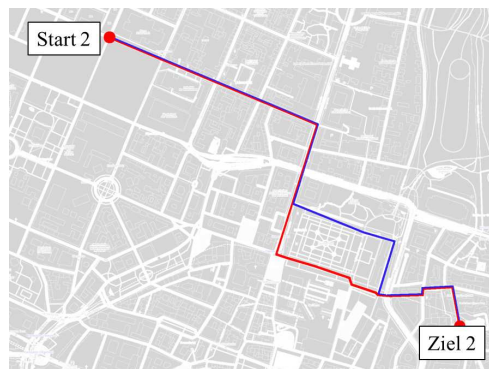
als die Personen gelaufen sind. Dies liegt daran, dass in diesem Experiment nur 20 Routen erfasst wurden.

Von Start 1 aus werden die gelaufenen Hauptachsen auch von den Algorithmen reproduziert. Durch die Einstellung, dass alle Wege mit Gesamtlängenabweichung kleiner als 10 % zum kürzesten Weg gefunden werden, existieren bei den Simulationsergebnissen ungefähr auf der Hälfte der Strecke unterschiedliche Teilstrecken, die alle in etwa gleich lang sind. Diese Variation ist in dem Maß nicht in den real gelaufenen Wegen zu finden.

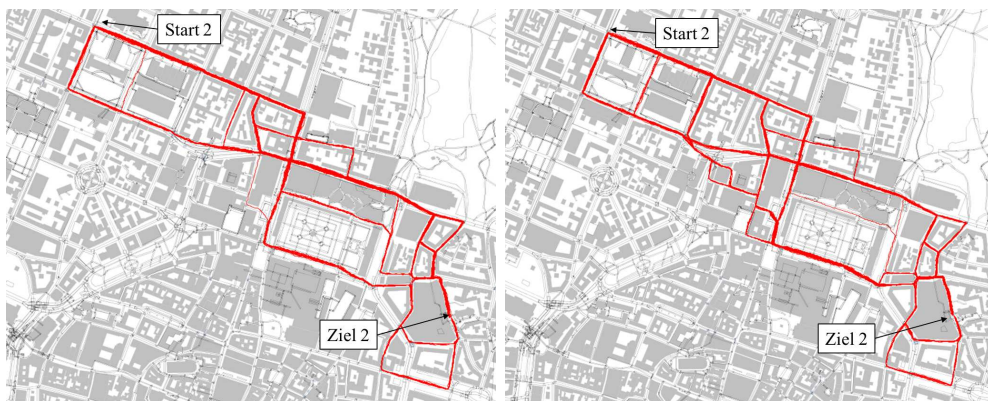
Die berechneten Routen des Simulationslaufs mit Einsatz von Landmarks verlaufen vermehrt entlang der südlichen Achse, was der Tendenz der real gelaufenen Strecken eher entspricht. Eine wesentliche Verbesserung der Routenähnlichkeit ist jedoch aus dem rein visuellen Vergleich nicht erkennbar.

Von Start 2 gibt es nur zwei Routen, mit denen die Simulationsergebnisse mit und ohne Landmarks zu vergleichen sind. Damit kann keine fundierte Aussage über die Qualität der Simulationsergebnisse getroffen werden. Jedoch ist ersichtlich, dass die Algorithmen zumindest visuell die gelaufenen Strecken beinhalten.

Dasselbe gilt für Start 3; auch hier sind nur zwei Routen zum Vergleich vorhanden. Diese werden von beiden Simulationsläufen reproduziert.

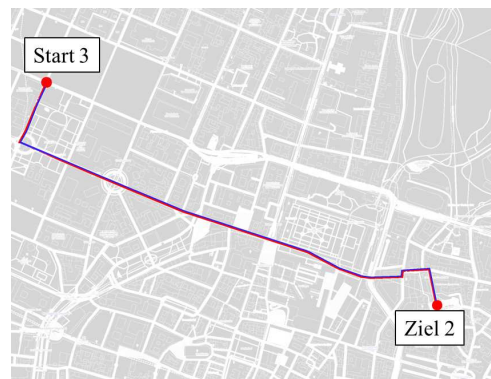


(a) Im Experiment gelaufene Routen

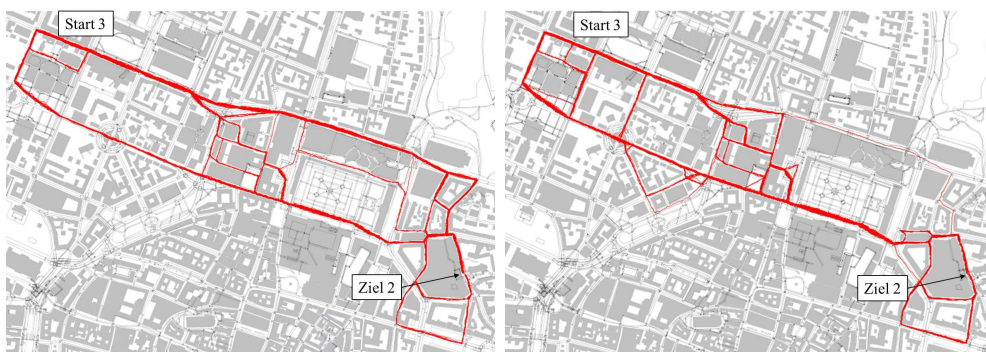


(b) Simulationsergebnis ohne Landmarks (c) Simulationsergebnis mit Landmarks

Abbildung 7.18: Die real gelaufenen Routen sowie die simulierten Routen von Startpunkt 2 zu Ziel 2



(a) Real gelaufene Routen



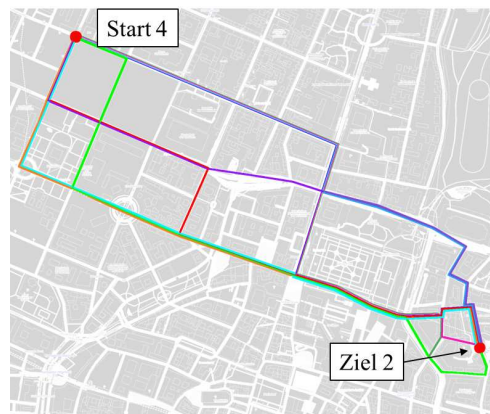
(b) Simulationsergebnis ohne Landmarks (c) Simulationsergebnis mit Landmarks

Abbildung 7.19: Die real gelaufenen Routen sowie die simulierten Routen von Startpunkt 3 zu Ziel 2.

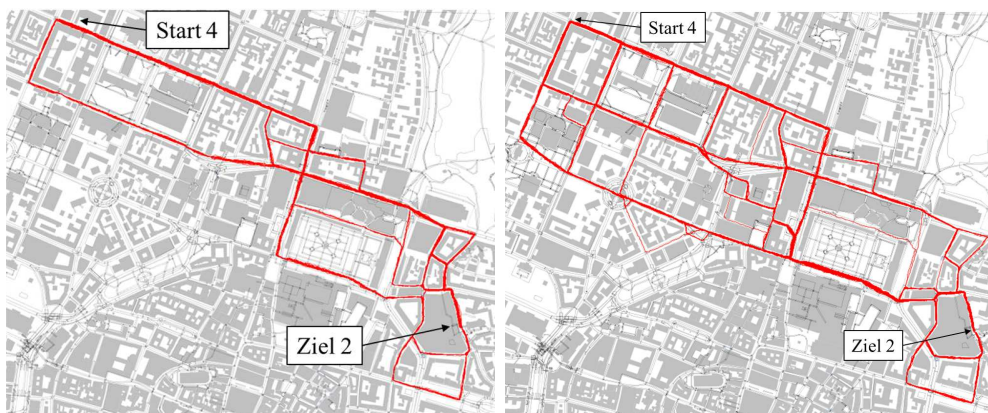
Für Start 4 steht die größte Anzahl von Routen zur Verfügung. Die Ergebnisse aus der Simulation ohne Landmarks können die südlichste Achse nicht reproduzieren. Dies liegt an der Lage von Start 4 zu Ziel 2. Die südlichste Achse liegt relativ weit weg von der Luftlinie und entspricht auch nicht einer Teilstrecke des kürzesten Wegs. Ein weiterer Grund dafür ist, dass auf der südlichsten Achse das Landmark „Karolinenplatz“ liegt.

Die Simulation mit Landmarks berechnet Wege entlang der südlichsten Achse und die Ergebnisse stimmen visuell besser mit den gelaufenen Wegen überein.

Zusammenfassend lässt sich auf Basis des visuellen Vergleichs sagen, dass die Routen zu Ziel 2 von der Simulation reproduziert werden können. Die Simulationsergebnisse zeigen eine höhere Variabilität auf, was daran liegt, dass nur 20 real gelaufene Routen zum Vergleich vorhanden sind, die Simulation aber mit 100 Personen pro Startpunkt durchgeführt wird. Die Ergebnisse aus dem Simulationslauf unter Miteinbeziehung von Landmarks kann die Routen von Start 4 besser reproduzieren. In Unterunterabschnitt 7.3.5.2 wird die Ähnlichkeit der Routen untersucht, um bessere Rückschlüsse über die Qualität der Simulationsergebnisse zu ziehen.



(a) Real gelaufene Routen



(b) Simulationsergebnis ohne Landmarks (c) Simulationsergebnis mit Landmarks

Abbildung 7.20: Die real gelaufenen Routen sowie die simulierten Routen von Startpunkt 4 zu Ziel 2.



### 7.3.5.2 Ähnlichkeitsbetrachtung

Dieser Vergleich zielt auf die Ähnlichkeit unterschiedlicher Routenpaare ab. Dazu werden die in Unterabschnitt 7.3.4 definierten Metriken für alle Routenpaare berechnet. Jede real gelaufene Route wird mit allen von der Simulation berechneten Routen verglichen. Das Paar mit der größten Übereinstimmung wird als Vergleichswert gewählt. Die Routen werden einmal mit den Simulationsergebnissen ohne Landmarks und einmal mit die Simulationsergebnisse mit Landmarks verglichen.

#### Ähnlichkeitsbetrachtung für Ziel 1

Die daraus resultierenden Werte für die Metrik *similarity* sind in Abbildung 7.21 dargestellt. Zum Einen sind die Ergebnisse der gelaufenen Routen im Vergleich zu den Routen des Simulationslaufs ohne Miteinbeziehung von Landmarks zu sehen und zum Anderen im Vergleich mit den Ergebnissen aus dem Simulationslauf mit dem Einsatz von Landmarks an Karolinenplatz und Odeonsplatz.

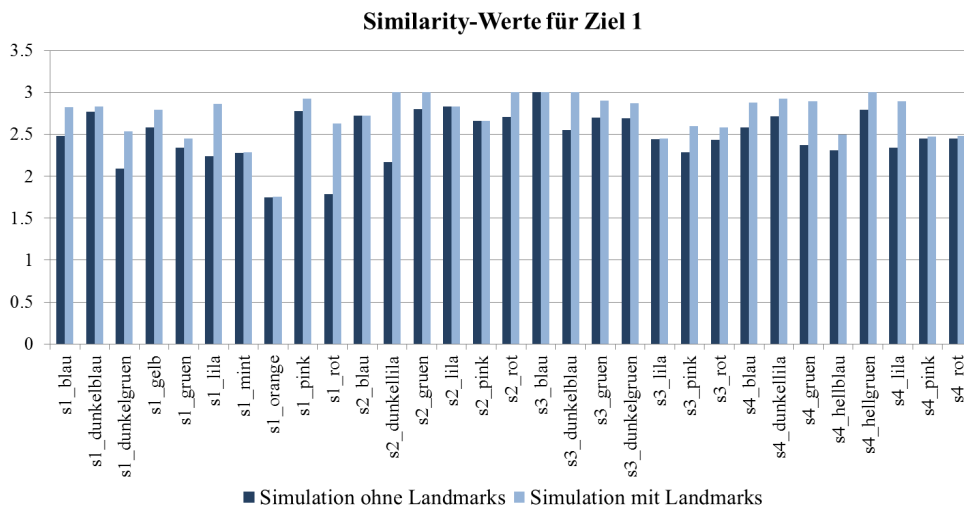


Abbildung 7.21: Similarity-Werte für alle Routen zu Ziel 1

Alle Routenpaare ohne Landmarks erreichen einen Wert größer als 1.5. Mit dem Einsatz von Landmarks verbessern sich die Werte auf über 2, bis auf eine Ausnahme, dem Weg „s1\_orange“. Dieser Weg verläuft zunächst weg vom Ziel, was die Simulation so nicht wiedergibt.

Um zu analysieren, welcher der drei Teilwerte wie hoch ist, sind in Abbildung 7.22 die Werte aus der Simulation ohne Landmarks, in Abbildung 7.23 die Werte aus der Simulation mit Landmarks pro Startpunkt aufgelistet. Die Routenpaare werden nach dem Muster  $\langle \text{Startpunkt} \rangle \_ \langle \text{Farbe der Route} \rangle \_ \langle \text{Algorithmus} \rangle$  bezeichnet. So ist direkt ersichtlich, welcher Algorithmus die besten Werte erzielt hat.

Die Anzahl der von den unterschiedlichen Algorithmen berechneten Routen mit den besten Werten ist in Tabelle 7.6 dargestellt. Zwei Routen wurden sowohl vom Fastest Path Algorithmus als auch vom Heuristic Path Algorithmus berechnet.

Algorithmus	Anzahl der Routen in der Simulation ohne Landmarks	Anzahl der Routen in der Simulation mit Landmarks
Fastest Path	5	20
Heuristic Path	18	20
SALL	10	2

Tabelle 7.6: Anteil der aus den unterschiedlichen Algorithmen resultierenden Routen zu Ziel 1 mit den größten Werten für die Metrik *similarity*

Der Grund dafür ist, dass der kürzeste Weg sehr nahe entlang der Luftlinie führt. Durch die Variationen der Gewichtung entsprechen diese Wege beiden Kriterien. Bei den Simulationsergebnissen unter Miteinbeziehung von Landmarks verschiebt sich das Bild etwas; dort erreichen nur noch zwei mit dem SALL Algorithmus berechnete Routen die beste Übereinstimmung. 11 Routen sind wieder jeweils das Ergebnis sowohl des Fastest Path Algorithmus als auch des Heuristic Algorithmus. Der Grund für die Erhöhung dieses Wertes liegt in den kürzeren Teilrouten zwischen den Landmarks; gerade die Startpunkte und das Landmark Karolinenplatz liegen nicht weit auseinander und zwischen Karolinenplatz und dem Ziel führt der kürzeste Weg ebenfalls entlang der Luftlinie. Der Grund für die Veränderung der Anteile der einzelnen Algorithmen wird im Folgenden bei der Untersuchung der einzelnen Startpunkte erklärt.

Von Start 1 aus existieren zwei Routenpaare für die Simulation ohne Landmarks mit einem sehr geringen Wert für den Anteil gleicher Strecke: das Routenpaar *s1\_dunkelgruen-Heuristic* und das Routenpaar *s1\_rot-SALL*. Das Routenpaar *s1\_orange-Fastest* wird dabei nicht betrachtet. Wie bereits erwähnt verläuft dieser Weg über einen Umweg, der von der Simulation nicht reproduziert wird. Das erste Routenpaar ist in Abbildung 7.24a dargestellt. Es ist erkennbar, dass die von der Simulation berechnete und die im Experiment gelaufene Route über weite Teile unterschiedlich verlaufen. Die im Experiment gelaufene Route orientiert sich zunächst am Karolinenplatz, während die von der Simulation berechnete Route global nach dem kürzesten Weg sucht. Mit dem Einsatz der Landmarks wird der Wert erheblich verbessert (siehe Abbildung 7.24b). Die beiden Routen verlaufen nun nahezu gleich und spannen nur noch ein sehr kleines Fehlerpolygon auf. Die von der Simulation berechnete Route entspricht sowohl dem Heuristic Path als auch dem Fastest Path.

Das Routenpaar *s1\_rot-SALL* ist in Abbildung 7.25a dargestellt. Obwohl die aus der Simulation ohne Landmarks resultierende Route bereits über den Karolinenplatz führt, sind die Charakteristiken der beiden Routen unterschiedlich. Die von der Simulation erstellte Route wurde mit dem SALL-Algorithmus berechnet, während die im Experiment gelaufene Route aus sehr vielen Richtungswechseln besteht. Vergleicht man die von der Simulation berechnete Route unter Miteinbe-

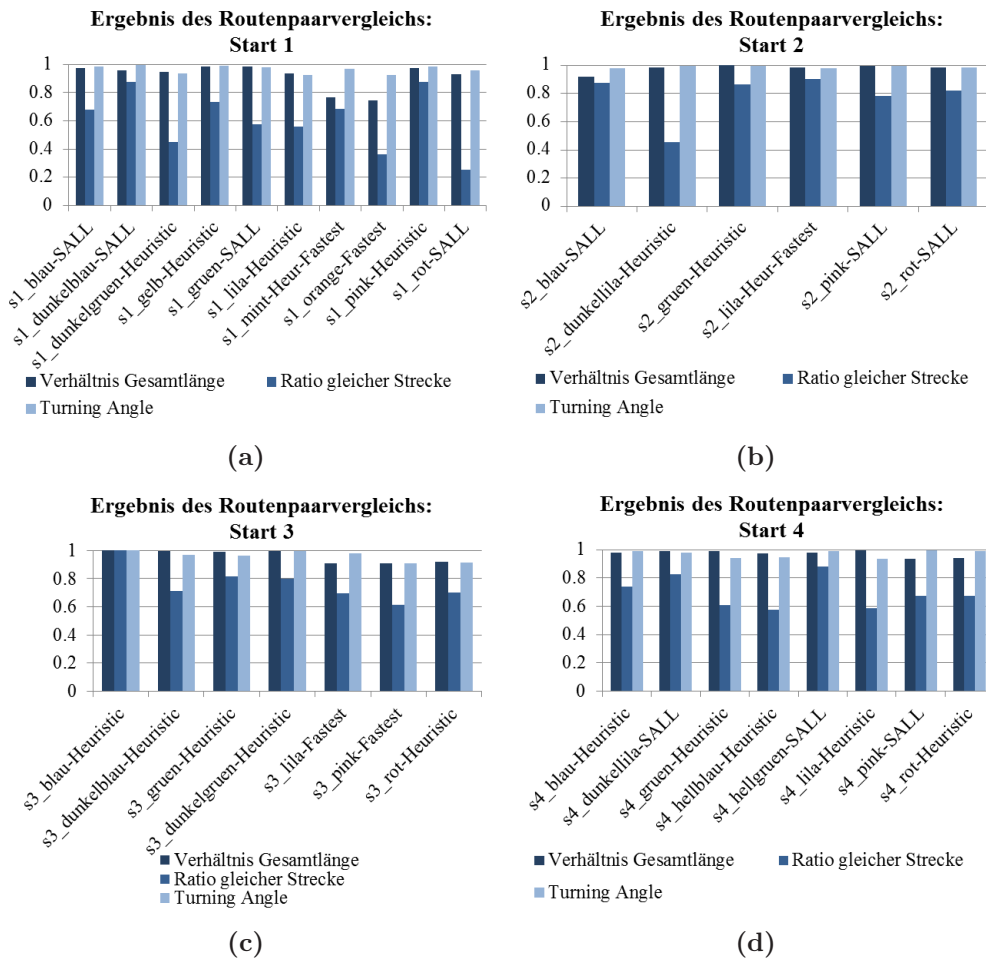


Abbildung 7.22: Die drei Teilmetriken sind für alle Routenpaare und jeden Startpunkt in Richtung Ziel 1 dargestellt. Es wird jeweils der Algorithmus mit dem Maximalwert für *similarity* als Wert gewählt. Aus dem Namen der Routenpaare ist der gewählte Algorithmus ablesbar.

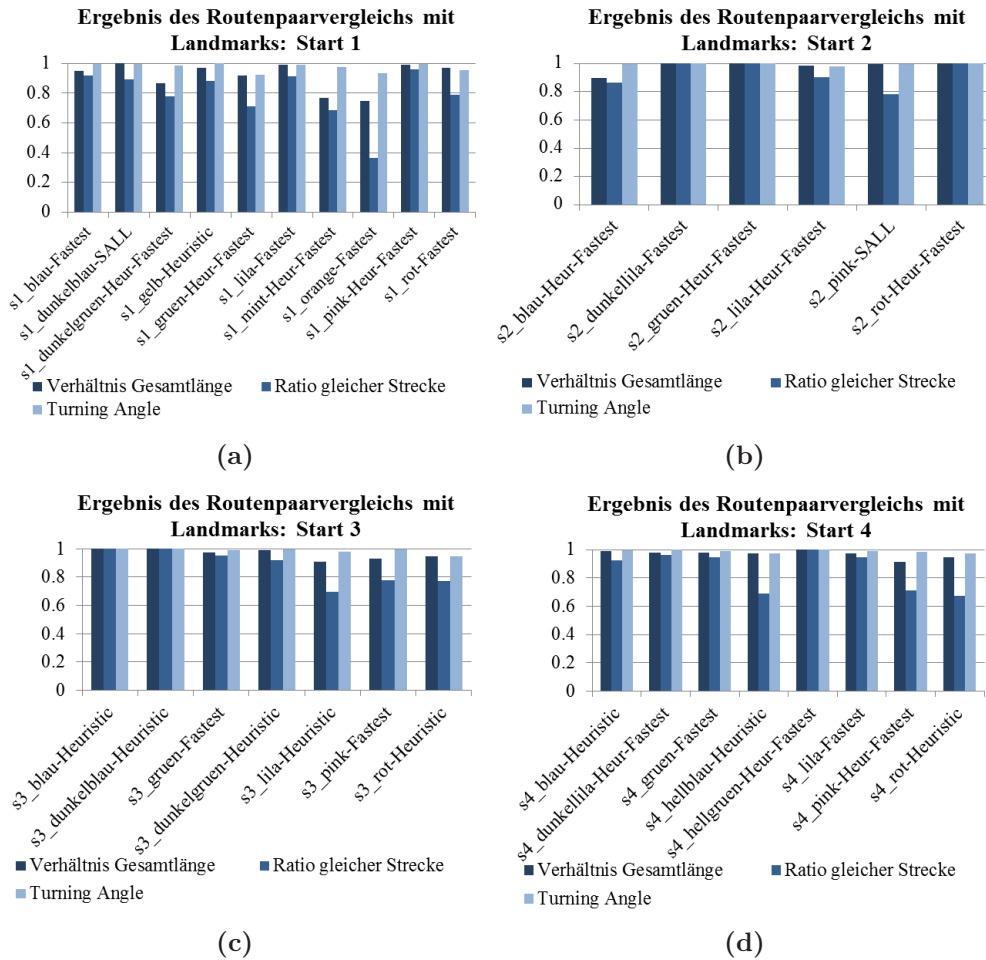
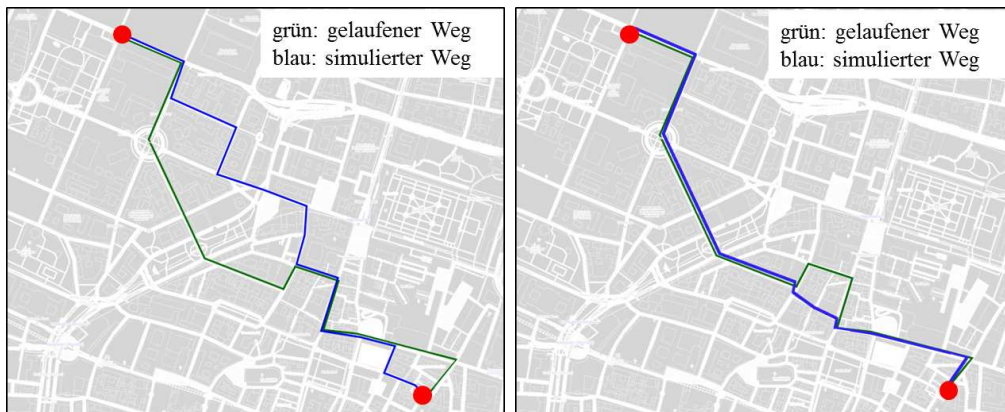


Abbildung 7.23: Die drei Teilmetriken sind für alle Routenpaare und jeden Startpunkt in Richtung Ziel 1 dargestellt. Alle Ergebnisse sind mit Einsatz von Landmarks produziert. Es wird jeweils der Algorithmus mit dem Maximalwert für *similarity* als Wert gewählt. Aus dem Namen der Routenpaare ist der gewählte Algorithmus ablesbar.

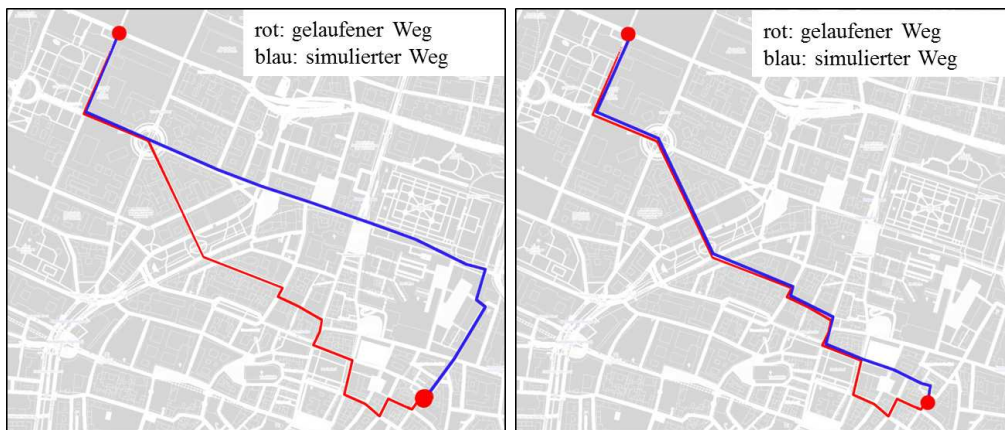


(a) Vergleich der gelaufenen Route mit der aus der Simulation ohne Landmarks berechneten Route in Richtung Ziel 1

(b) Vergleich der gelaufenen Route mit der aus der Simulation mit Landmarks berechneten Route in Richtung Ziel 1

Abbildung 7.24: Routenpaare a) s1\_dunkelgruen-Heuristic und b) s1\_dunkelgruen-Heur-Fastest

ziehung des Karolinenplatzes als Landmark mit der real gelaufenen Route (vgl. Abbildung 7.25b), so ergibt sich eine weitaus größere Übereinstimmung. Diese Route wurde mit dem Fastest Path Algorithmus erzeugt, was der Charakteristik der real gelaufenen Route weit mehr entspricht.



(a) Vergleich der gelaufenen Route mit der aus der Simulation ohne Landmarks berechneten Route in Richtung Ziel 1

(b) Vergleich der gelaufenen Route mit der aus der Simulation mit Landmarks berechneten Route in Richtung Ziel 1

Abbildung 7.25: Routenpaare a) s1\_rot-SALL und b) s1\_rot-Fastest

Die Werte für Startpunkt 2 liegen alle über 2. Auch alle Einzelwerte erreichen relativ hohe Werte, nur das Routenpaar *s2\_dunkellila-Heuristic* besitzt einen niedrigeren Wert für die Ratio. Dieses Routenpaar ist in Abbildung 7.26 dargestellt. Die Route ohne Miteinbeziehung von Landmarks verläuft östlicher als die real gelaufene Route. Dies liegt daran, dass die Luftlinie zum Ziel östlicher verläuft als die Route. Dennoch verlaufen die beiden Routen innerhalb des Fehlerpolygons

sehr symmetrisch. Dies wird im hohen Wert für den  $TA$  wiedergespiegelt. Werden Landmarks in die Simulation miteinbezogen, so wird die real gelaufene Route zu 100 % reproduziert. Betrachtet man die beiden Teilwege bis zum Landmark und zwischen Landmark und Ziel, so ist festzustellen, dass diese Wege sehr nahe entlang der Luftlinie verlaufen. Daher bleibt hier die Strategie die gleiche, nur die Orientierung erfolgt nicht global, sondern besteht aus zwei Schritten.

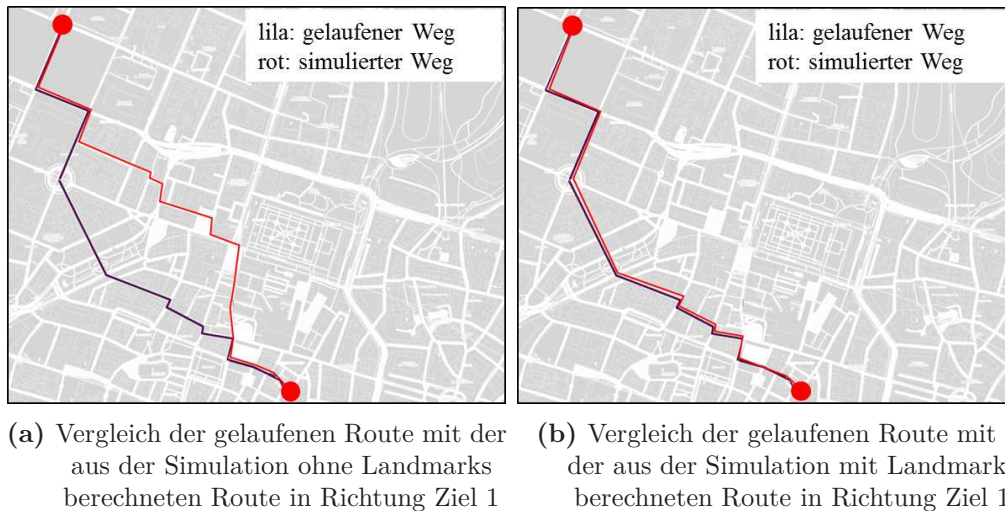


Abbildung 7.26: Routenpaar  $s2\_dunkellila$ -Heuristic

Die Routen von Startpunkt 3 aus können von der Simulation ebenfalls gut reproduziert werden. Die geringsten Werte erreicht das Routenpaar  $s3\_pink\_Fastest$ . Dort ist die Ratio relativ gering. Die Route verläuft ähnlich der Route  $s1\_dunkelgruen$  westlicher als die von der Simulation berechnete Route und macht einen nördlichen Abstecher nach Erreichen des Karolinenplatzes. Die von der Simulation berechnete Route führt zwar ebenfalls über den Karolinenplatz, allerdings optimaler in Richtung Ziel als die gelaufene Route. Daher wird auch der Wert mit Einsatz von Landmarks nicht wesentlich erhöht.

Bei den Routen von Startpunkt 4 liegen die Werte für die Ergebnisse der Simulation ohne Landmarks alle über 2. Hier fällt auf, dass der Einsatz von Landmarks die Hälfte der Routen erheblich verbessern kann. Dies ist bereits beim visuellen Vergleich aufgefallen. Da der Karolinenplatz nicht auf der Luftlinie zwischen Start und Ziel liegt, führen kaum von der Simulation berechnete Wege über diesen Platz. Im Experiment führen 5 der 8 Routen über diesen Platz. Durch den Einsatz von Landmarks können deshalb die Werte dieser Routen verbessert werden.

Insgesamt ist eine Verbesserung der Übereinstimmung mit dem Einsatz der Landmarks zu erkennen, auch für die hier nicht explizit dargestellten Routen. Dies liegt daran, dass die Routen ohne Landmarks teilweise östlicher verlaufen als die real gelaufenen. Blickt man auf die einzelnen Teilwerte der Metrik *similarity*, so ist festzustellen, dass vor allem der Wert für die Ratio übereinstimmender Streckenabschnitte verbessert wurde. Dies ist mit dem Einsatz von Landmarks und den dadurch westlicheren Verlauf der Routen zu erklären. Vergleicht man die Werte

Algorithmus	Anzahl der Routen in der Simulation ohne Landmarks	Anzahl der Routen in der Simulation mit Landmarks
Fastest Path	5	9
Heuristic Path	8	8
SALL	9	6

Tabelle 7.7: Anteil der aus den unterschiedlichen Algorithmen resultierenden Routen zu Ziel 2 mit den größten Werten für die Metrik *similarity*

derjenigen Routen, die mit dem Simulationslauf ohne Landmarks sehr schlechte Übereinstimmungswerte hatten, so ist eine erhebliche Verbesserung der Werte zu erkennen.

Daraus erschließt sich die Bedeutung von Landmarks in der Wegefindung. Konnte der Simulator mit den entwickelten Algorithmen die gelaufenen Wege nur teilweise gut abbilden, ergibt sich eine wesentlich bessere Übereinstimmung mit dem Einsatz von Landmarks.

### Ähnlichkeitsbetrachtung für Ziel 2

In Abbildung 7.27 sind die Ergebnisse für die Routenpaare zu Ziel 2 dargestellt, einmal für die Simulation ohne Landmarks und einmal für die Simulation mit Landmarks. Zunächst ist zu erkennen, dass alle Routenpaare der Simulation ohne Landmarks bereits einen besseren Wert erreichen als die Routenpaare von Ziel 1. Erklärt werden kann das durch die Zuhilfenahme der Karte, anhand derer die Zielrichtung immer abzulesen und eine bessere Orientierung möglich ist. Die Routen, die mit dem Einsatz von Landmarks berechnet wurden, erreichen ebenfalls bessere Werte wie auch bei Ziel 1.

Die meisten Routen mit der größten Übereinstimmung des *similarity*-Wertes in der Simulation ohne Landmarks erzielen die mit dem SALL-Algorithmus berechneten Routen. Dies überrascht nicht, da die Routen durch lange Strecken geradeaus charakterisiert sind. Eine Route wird sowohl vom Heuristic Path Algorithmus als auch Fastest Path Algorithmus erzeugt. Dies liegt wieder in der Variation in den Kantengewichten sowie an der Tatsache, dass der kürzeste Weg nahe entlang der Luftlinie verläuft.

Unter der Miteinbeziehung von Landmarks ändern sich die Übereinstimmungen mit den Algorithmen leicht. Die Anzahl der Routen, die mit dem Fastest Path die größte Übereinstimmung erzielen, erhöht sich. Dies ist ein interessantes Ergebnis. Erklärt werden kann dies damit, dass die Probanden trotz Zuhilfenahme einer Karte zunächst bekannte Orte ansteuerten, und nur zwischen diesen Punkten jeweils den kürzesten Weg herausgelesen haben. Von diesen 9 Wegen wurden 3 Wege sowohl vom Fastest Path Algorithmus als auch vom Heuristic Path Algorithmus berechnet.

Insgesamt werden ohne Einsatz von Landmarks bereits 4 von 20 Routen exakt von der Simulation reproduziert, mit dem Einsatz von Landmarks erhöht sich der Wert auf 12 der 20 Routen.

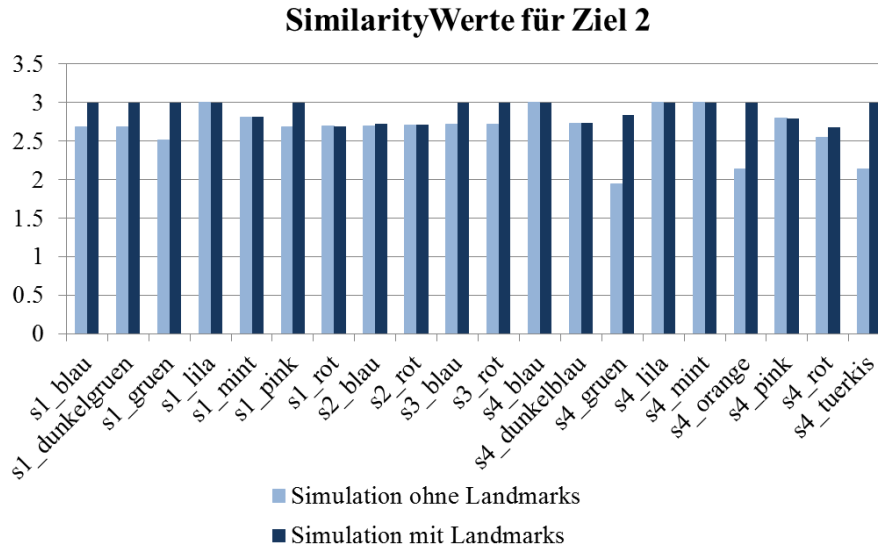


Abbildung 7.27: Similarity-Werte für alle Routen zu Ziel 2

Blickt man genauer auf die in Abbildung 7.28 aufgelisteten Teilmetriken für die Routenpaare aus der Simulation ohne Landmarks, so ist ersichtlich, dass alle Teilmetriken hohe Werte erreichen. Eine Ausnahme bilden die Wege von Start 4. Hier existieren teilweise sehr geringe Werte für die Ratio.

Die Ergebnisse der Teilmetriken für die Simulation mit Landmarks sind in Abbildung 7.29 aufgelistet. Hier nehmen die Teilmetriken höhere Werte an, vor allem die Werte von Start 4 können verbessert werden.

Die Routenpaare von Start 4 mit den geringsten Werten für die Ratio werden näher betrachtet.

In Abbildung 7.30 sind die Routenpaare *s4\_orange-Heuristic* und *s4\_tuerkis-Heuristic* eingezeichnet. Für den Simulationslauf ohne Landmarks wird für die mit dem Heuristic Path Algorithmus berechnete Route die größte Übereinstimmung erzielt. Weite Teile der Strecke verlaufen zwar unterschiedlich, jedoch parallel zueinander. Dies ist auch im Wert der Metrik wiederzufinden; der Wert für den „Turning Angle“ ist relativ hoch. Vergleicht man die Routenpaare mit den Simulationsergebnissen unter Miteinbeziehung von Landmarks, so kann die Simulation die gelaufenen Routen zu 100 % reproduzieren (vgl. Abbildung 7.30b). Durch den Einsatz von Landmarks führt dieser Weg über den Odeonsplatz und nicht eine Straße nördlicher entlang.

Der Weg *s4\_gruen-SALL* ist zusammen mit dem von der Simulation ohne Landmarks berechneten Weg in Abbildung 7.31a eingezeichnet. Hier ist noch deutlicher zu erkennen, dass die beiden Wege über weite Strecken unterschiedlich verlaufen,



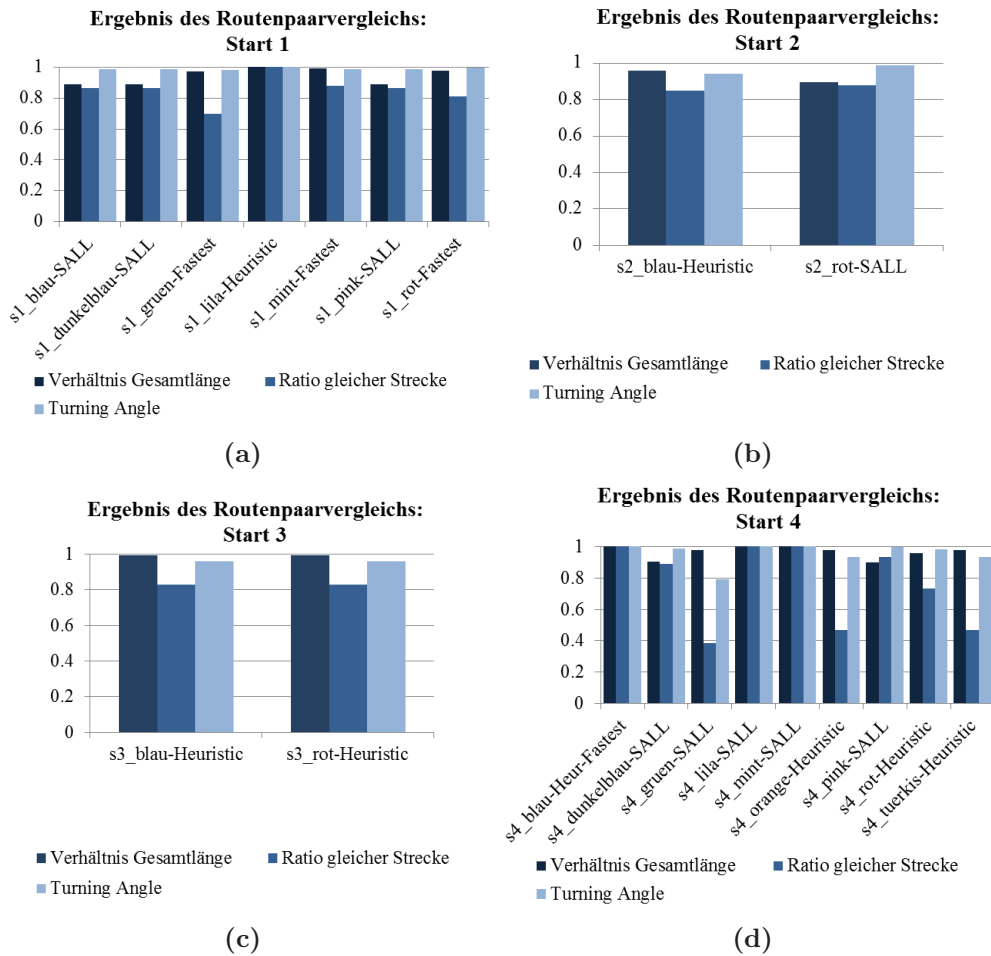


Abbildung 7.28: Die drei Teilmetriken sind für alle Routenpaare und jeden Startpunkt in Richtung Ziel 2 dargestellt. Es wurde jeweils der Algorithmus mit dem Maximalwert für *similarity* als Wert gewählt. Aus den Namen der Routenpaare ist der gewählte Algorithmus ablesbar.

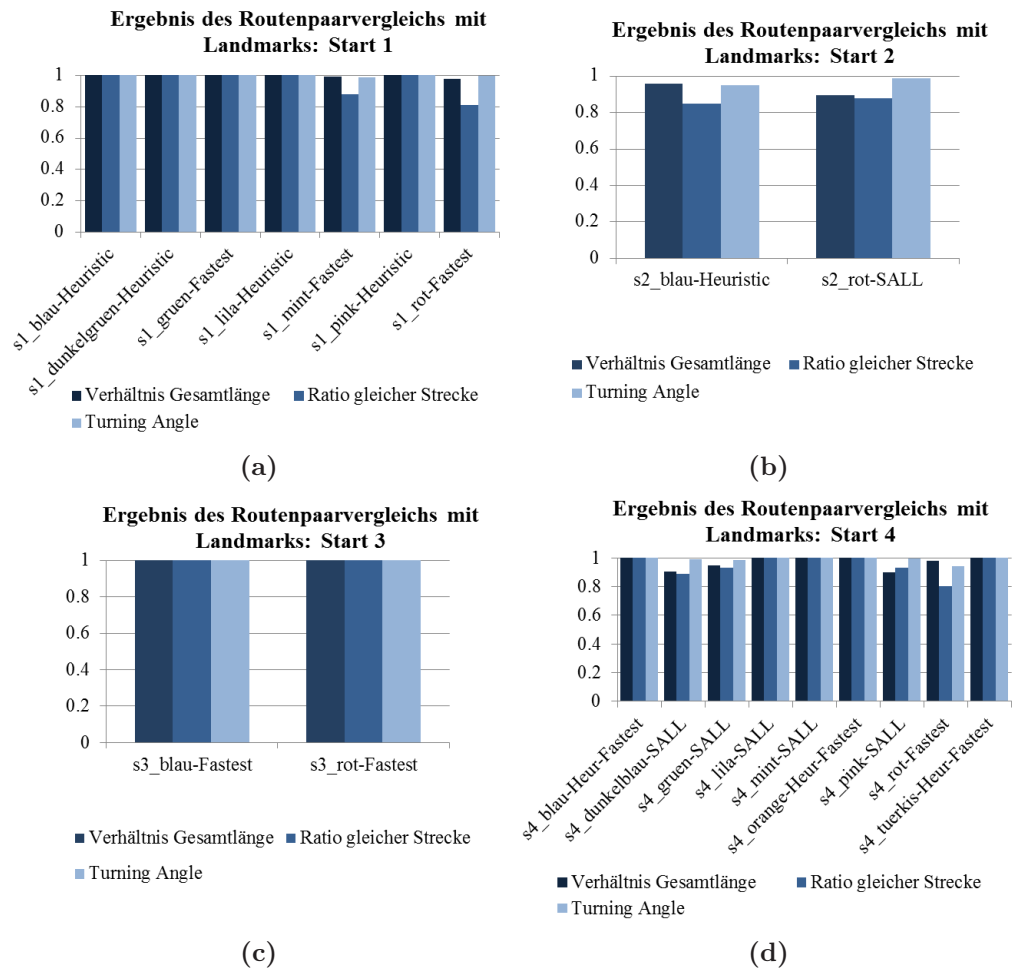
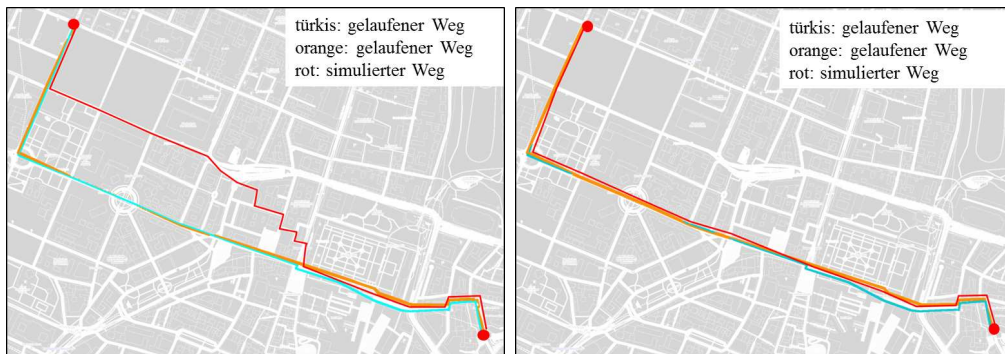


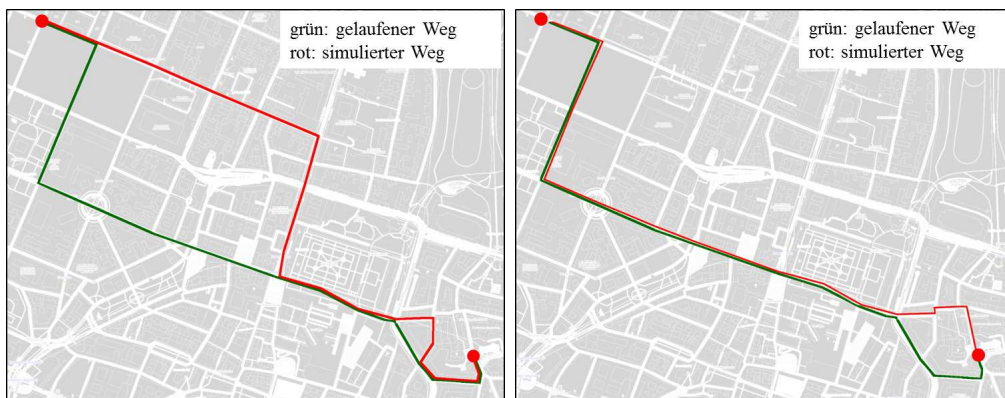
Abbildung 7.29: Die drei Teilmetriken sind für alle Routenpaare und jeden Startpunkt in Richtung Ziel 2 dargestellt. Die von der Simulation erstellten Routen sind mit Einbeziehung von Landmarks. Es wird jeweils der Algorithmus mit dem Maximalwert für *similarity* als Wert gewählt. Aus den Namen der Routenpaare ist der gewählte Algorithmus ablesbar.



(a) Vergleich der gelaufenen Route mit der aus der Simulation ohne Landmarks berechneten Route

(b) Vergleich der gelaufenen Route mit der aus der Simulation mit Landmarks berechneten Route

Abbildung 7.30: Routenpaare s4\_orange-Heuristic und s4\_tuerkis-Heuristic



(a) Vergleich der gelaufenen Route mit der aus der Simulation ohne Landmarks berechneten Route

(b) Vergleich der gelaufenen Route mit der aus der Simulation mit Landmarks berechneten Route

Abbildung 7.31: Routenpaar s4\_gruen-SALL

aber exakt parallel zu einander. Der Abstand der beiden Wege wird durch den Einsatz von Landmarks eliminiert, nur kurz vor dem Erreichen des Ziel spalten sich die beiden Routen des Routenpaares noch einmal symmetrisch auf. Die Charakteristik des von der Simulation mit Landmarks erzeugten Wegs entspricht nach wie vor dem SALL Algorithmus. Zusammenfassend lässt sich sagen, dass für Ziel 2 bereits ohne den Einsatz von Landmarks eine gute Übereinstimmung der gelaufenen und der von der Simulation berechneten Wege existiert. Diese kann durch den Einsatz der Landmarks Karolinenplatz und Odeonsplatz weiter verbessert werden; gerade die südlichste Achse wird nun auch von der Simulation reproduziert. Insgesamt erzielt die Simulation die besseren Ergebnisse für Ziel 2, was durch die Tatsache begründet ist, dass die Teilnehmer einen Stadtplan zur Verfügung hatten und zudem die Strecke nicht so verwinkelt ist wie die Strecke zu Ziel 1.

### Vergleich der Algorithmen zu konventionellen Algorithmen

Um die Ergebnisse der Algorithmen in Relation zu Routingalgorithmen zu setzen,

die in den meisten Simulationsmodellen eingesetzt werden, werden die gelaufenen Routen mit dem klassischen Dijkstra-Algorithmus mit euklidischer Distanz als Kantengewicht verglichen. Hier ist anzumerken, dass dies in diesem Szenario dem Fastest Path entspricht, da keine Staus auftreten und die kürzesten Wege den schnellsten Wegen entsprechen. Durchgeführt wird die Simulation mit dem Fastest Path Algorithmus ohne Variation der zehn schnellsten Wege. Er wird im Vergleich als *Fastest Path* bezeichnet.

Ein weiterer Vergleich wird mit einem Algorithmus durchgeführt, der die Kanten zufällig auswählt. Dies soll belegen, dass Fußgänger bei der Wegefindung einer Metrik folgen und nicht rein zufällig eine Kante wählen. Der Algorithmus wird im Folgenden als *Random Path* bezeichnet. Die Ergebnisse des Vergleichs werden nicht nach Ziel 1 und Ziel 2 aufgeschlüsselt, sondern gemeinsam betrachtet.

In Abbildung 7.32 sind die Bandbreiten der *similarity*-Werte für die Simulationen der drei hier verwendeten Algorithmen mit und ohne dem Einsatz von Landmarks den Werten des Fastest Path Algorithmus und des Random Path Algorithmus gegenübergestellt. Pro Algorithmus wird der jeweils kleinste Wert, der Mittelwert und der Maximalwert dargestellt.

Die zufällig gewählten Wege des Random Path schneiden erwartungsgemäß am schlechtesten ab. Dies belegt, dass Menschen einen Weg nicht zufällig finden, sondern eine Metrik im Kopf haben, nach der sie den Weg suchen. Dass diese Metrik nicht der allein der euklidischen Distanz entspricht, ist aus dem Wert des Vergleichs mit dem schnellsten Weg ersichtlich. Der Mittelwert liegt unter 2 und der Unterschied zwischen Minimum und Maximum ist sehr groß. Zwar wird mit dem Fastest Path eine gute Übereinstimmung für einen Teil der Wege erzielt, aber es existieren auch einige Wege, die der Metrik des Fastest Path nicht entsprechen. Die Kombination der drei unterschiedlichen Algorithmen „Schnellster Weg“, „entlang der Luftlinie“ und „SALL“ erzielt einen deutlich höheren Mittelwert bei 2.57. Die Spanne zwischen dem Minimum und Maximum ist nicht mehr so groß und im Optimalfall kann einer der drei Algorithmen den gelaufenen Weg komplett abbilden.

Noch bessere Werte erzielen die Ergebnisse der Algorithmen unter Miteinbeziehung von Landmarks. Mit der korrekten Platzierung von Landmarks für jedes der beiden Ziele liegt der Mittelwert bei 2.81. Insgesamt werden so 18 von den 51 gelaufenen Routen exakt reproduziert. Dies erklärt den relativ hohen Mittelwert. Daraus erschließt sich, dass sich Personen nicht ausschließlich entlang des kürzesten Wegs orientieren. Viele mit dem Heuristic Path Algorithmus berechneten Wege erzielen zu beiden Zielen die beste Übereinstimmung. Zudem orientieren sich Personen an markanten Punkten und bevorzugen auch – zumindest in diesem Szenario, einem städtischen Umfeld – möglichst gerade Strecken ohne häufigem Abbiegen.

### 7.3.5.3 Vergleich der relativen Häufigkeit aller traversierten Kanten

Der letzte Vergleich untersucht die relative Häufigkeit aller traversierten Kanten. Er soll Auskunft darüber geben, ob die gelaufenen Hauptwege auch in den Simulationsergebnissen erkennbar sind. Dies ist bei der Simulation eines Eva-

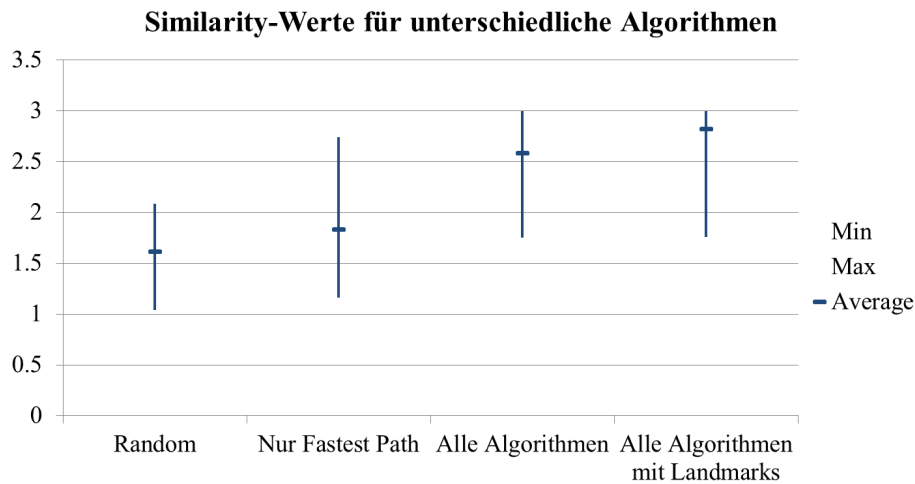


Abbildung 7.32: Bandbreiten des *Similarity*-Wertes für unterschiedliche Algorithmen für das Szenario. Alle Wege fließen hier in die Bandbreite mit ein.

kuierungsszenarios von Bedeutung: Gerade die meist gewählten Routen können schnell zu einem Flaschenhals werden. Kann die Simulation häufig begangene Wege nicht reproduzieren, wird sie auch nicht in der Lage sein, Flaschenhalse an den korrekten Stellen zu prognostizieren.

Da sich aus den beiden ersten Vergleichen herauskristallisiert hat, dass die Simulationsläufe mit dem Einsatz von Landmarks in jedem Fall die realistischeren Ergebnisse erzielen, wird dieser Vergleich ausschließlich mit den Simulationsergebnissen mit dem Einsatz von Landmarks durchgeführt.

Die prozentuale Verteilung der drei verwendeten Algorithmen wird so gewählt, wie in Tabelle 7.6 und Tabelle 7.7 in der Spalte „Anzahl der Routen in der Simulation mit Landmarks“ dargestellt.

### Relative Häufigkeit der Kanten auf dem Weg zu Ziel 1

In Abbildung 7.33 sind alle Kanten des Experiments, die in mehr als 20 % aller Wege enthalten sind und die entsprechenden relativen Häufigkeiten, die in der Simulation erreicht werden, gegenübergestellt. Für jeden Startpunkt wurde die zehnfache Anzahl von Personen generiert als im Experiment Routen gelaufen wurden. Damit wird das Verhältnis der Wege von den einzelnen Startpunkten bewahrt.

Die Kanten, die in den meisten im Experiment gelaufenen Wegen enthalten sind, werden in gleicher Größenordnung auch von den berechneten Wegen der Simulation reproduziert. Die maximale Abweichung zwischen relativer Häufigkeit einer Kante in Bezug auf die gelaufenen Wege und relativer Häufigkeit einer Kante in Bezug auf die von der Simulation berechneten Wege beträgt 17 %, die durchschnittliche Abweichung liegt bei 5 %. Dies ist ein akzeptables Ergebnis.

Zum visuellen Vergleich der Häufigkeit aller Kanten sind die Kanten in Abbildung 7.34 nach relativer Häufigkeit eingefärbt. Eine Übereinstimmung liegt für

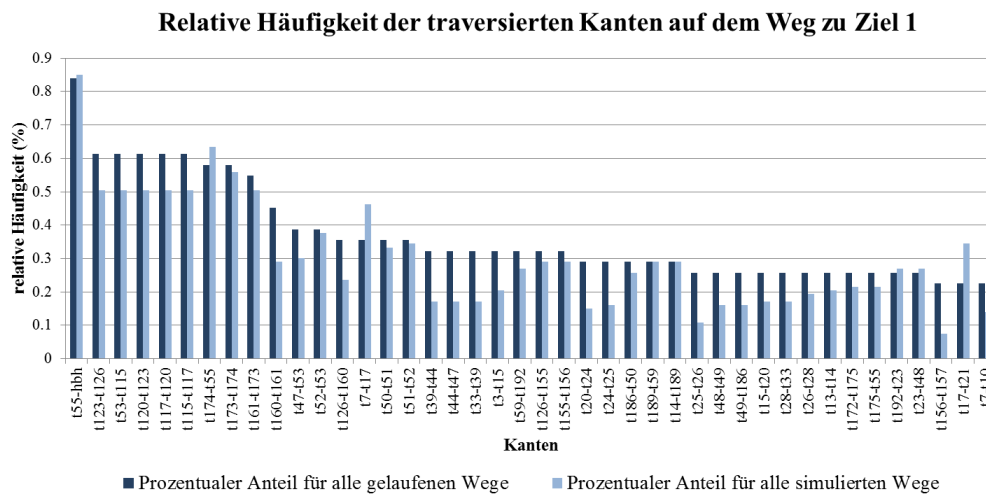


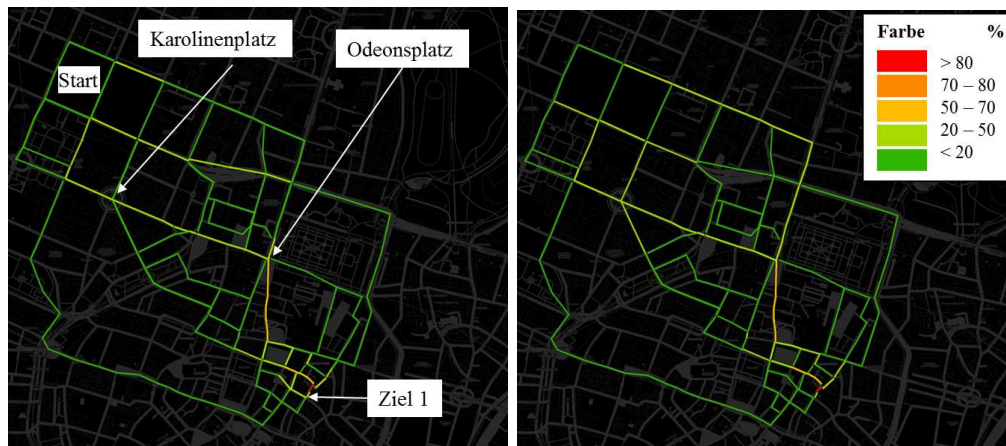
Abbildung 7.33: Vergleich der relativen Häufigkeit für die im Experiment gelaufenen Wege und die von der Simulation berechneten Wege

die Kanten südlich des Odeonsplatzes vor, sowie für einige Kanten in der Nähe des Ziels.

Von den drei parallelen Hauptachsen nörd-westlich des Odeonsplatzes, die sich im Experiment herausgebildet haben, können die südlichen zwei Achsen ebenfalls von der Simulation reproduziert werden, die nördlichste Achse ist in den berechneten Wegen nicht so häufig enthalten. Dennoch weisen die von der Simulation berechneten Routen auch im Bezug auf die relative Häufigkeit eine Ähnlichkeit auf.

### Relative Häufigkeit der Kanten auf dem Weg zu Ziel 2

Für Ziel 2 wird ebenfalls die relative Häufigkeit der einzelnen Kanten bestimmt. Da von Start 2 und Start 3 aus nur jeweils zwei Wege aus dem Experiment existieren, werden diese Startpunkte aus dem Vergleich ausgeschlossen. Für die beiden anderen Startpunkte wird jeweils die Anzahl der im Experiment gelaufenen Routen verzehnfacht. Dies ermöglicht eine Variabilität in den Ergebnissen bei gleichzeitiger Beibehaltung der Verhältnisse. Die Werte der relativen Häufigkeit für alle Kanten aus dem Experiment, die einen Wert größer als 20 % erzielen, sind in Abbildung 7.35 den Werten der relativen Häufigkeit für die Kanten aus den Simulationsergebnissen gegenübergestellt. Auch hier ist der Unterschied der Werte für die meisten Kanten sehr gering. Die am häufigsten traversierte Kante im Experiment ist in gleichem Maße in den von der Simulation berechneten Wegen enthalten. Auch für die weiteren Kanten mit einem Wert von 81 % liegen die Werte der Simulation in der Nähe, jedoch etwas überhalb der Werte aus dem Experiment. Die maximale Abweichung der Häufigkeit liegt bei 20 %, im Mittel weicht der Wert der Häufigkeit nur um 1 % ab. Die Extremwerte sind aus dem Diagramm ersichtlich. Die Kante  $t25-t26$  sowie  $t26-t28$  haben einen Wert in den real gelaufenen Wegen von 20 %, während in der Simulation nur 4 % aller Wege



(a) relative Häufigkeit der Kanten für die von der Simulation berechneten Wege (b) relative Häufigkeit der Kanten für die im Experiment gelaufenen Wege

Abbildung 7.34: Visueller Vergleich der relativen Häufigkeit

über diese Kante führen.

Blickt man auf den visuellen Vergleich in Abbildung 7.36, so ist zu erkennen, dass die die Kanten mit dem größten Anteil kurz vor dem Ziel liegen. Auch die Hauptachse westlich und östlich des Odeonsplatzes wird von der Simulation reproduziert. Die restlichen Kanten zeigen weitgehend ähnliche Werte für die relative Häufigkeit auf. Hier ist – wie auch bei Ziel 1 – zu erkennen, dass die Wege der Simulation nicht so häufig entlang der nördlichsten West-Ost-Achse führen wie die Wege des Experiments. Dies kann ganz einfach mit der längeren Wegstrecke begründet werden. Der Weg über die nördlichste Achse ist erheblich weiter als die Wege über die beiden anderen Achsen.

Insgesamt stimmen die Simulationsergebnisse sowohl visuell, bezüglich der Ähnlichkeitsbetrachtung als auch bezüglich der relativen Häufigkeit der Kanten gut mit den Routen des Experiments überein. Die berechneten Routen der Simulation erreichen keine 100 %-ige Übereinstimmung mit den im Experiment gelaufenen Routen. Dies ist aber auch keine Notwendigkeit. Da Menschen keine deterministischen Entscheidungen treffen, sehen die Ergebnisse bei einem neuen Experiment mit großer Wahrscheinlichkeit etwas anders aus. Wichtig ist es, dass die Simulationsergebnisse die richtigen Trends widerspiegeln kann, wie die Charakteristik einer Route und die Häufigkeit der traversierten Kanten. Somit kann gewährleistet werden, dass die Ergebnisse der Simulation die Realität reflektieren. Und dies ist bei eingehender Analyse auf die Ergebnisse der Fall.

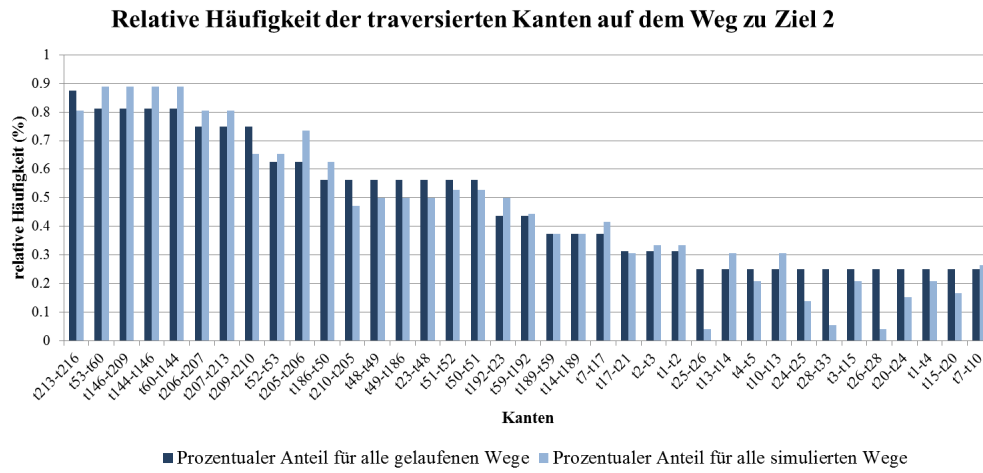
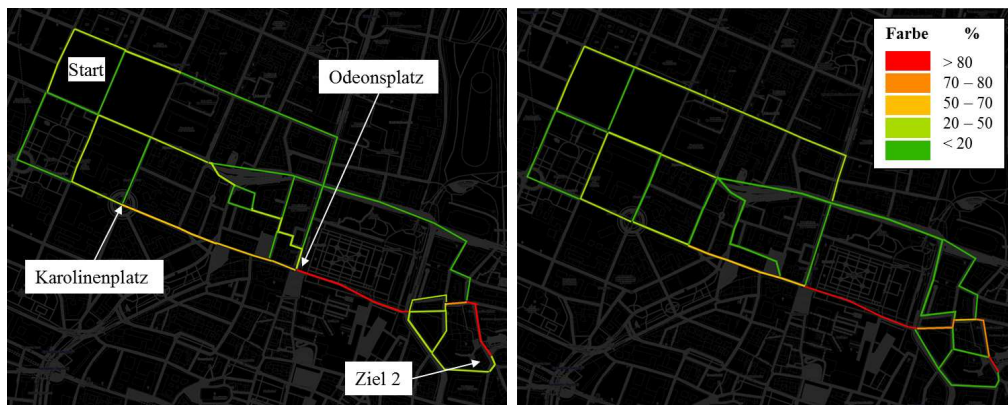


Abbildung 7.35: Vergleich der relativen Häufigkeit für die im Experiment gelaufenen Wege und die von der Simulation berechneten Wege



(a) relative Häufigkeit der Kanten für die von der Simulation berechneten Wege (b) relative Häufigkeit der Kanten der real gelaufenen Wege

Abbildung 7.36: Visueller Vergleich der relativen Häufigkeit



## 7.4 Zusammenfassung

In diesem Kapitel wurden die entwickelten Wegefindungsalgorithmen anhand eines durchgeführten Experiments zur Wegefindung validiert. Während des Experiments liefen Personen in einem städtischen Umfeld zu einem bekannten Platz, einmal mit Zuhilfenahme einer Karte und einmal ohne Hilfsmittel. Die gelaufenen Routen sind sehr verschieden, es ist keine klare Hauptroute erkennbar. Viele der Routen führen über dieselben bekannten Plätze. Es hat sich herausgestellt, dass diese Landmarks eine wichtige Orientierungshilfe bei der Wegefindung sind. Besonders auffällig ist dies beim Blick auf die Routen der Gruppe, die keine Hilfsmittel zur Orientierung zur Verfügung hatten.

Die Simulationsläufe wurden unter Anwendung von drei unterschiedlichen Algorithmen durchgeführt. Dabei wurden einmal keine Landmarks verwendet und einmal Landmarks an den bekannten Orten integriert. Zum Vergleich der simulierten Routen mit den gelaufenen Wegen wurde eine neue Metrik *similarity* eingeführt, die aus dem Verhältnis der Länge, der Ratio gleicher Wegstrecke sowie dem Turning Angle besteht.

Die Ergebnisse des Vergleichs zeigen nicht nur visuell eine gute Übereinstimmung, auch bei der Ähnlichkeitsbetrachtung mit Hilfe der Metrik *similarity* kann nachgewiesen werden, dass die Routen der Simulation den im Experiment gelaufenen Routen ähnlich sind.

Desweiteren wurden die gelaufenen Wege mit den kürzesten Wegen sowie rein zufällig berechneten Wegen verglichen und die *similarity*-Werte ermittelt. Hier zeigt sich, dass eine Kombination der Algorithmen die Abbildung der im Experiment gelaufenen Wege erheblich verbessert und in Kombination mit Landmarks noch weiter annähert.

Betrachtet wurde auch, in wie vielen Wegen eine Kante enthalten ist. Bei diesem Vergleich spiegeln sich die gelaufenen Hauptwege auch in den Simulationsergebnissen wieder.

Die Vorgehensweise des Vergleichs bezüglich der Metrik *similarity* kann zukünftig auch als Kalibrierung eingesetzt werden: Durch die getrennten Simulationsläufe der einzelnen Algorithmen und den anschließenden Vergleich der Ergebnisse mit allen vorliegenden Routen können die Gewichte der einzelnen Algorithmen abgeleitet werden. Dies wurde hier bereits für den Vergleich der relativen Häufigkeit durchgeführt.

Insgesamt ist aus den Ergebnissen des Experiments zu schließen, dass Personen nicht ausschließlich die kürzesten Wege finden, sondern sich auch an bekannten Orten orientieren und versuchen, möglichst nahe entlang der Luftlinie oder lange ohne Richtungswechsel zu laufen. Um realistische Simulationsergebnisse zu erhalten, ist es daher von großer Bedeutung, zum Einen bekannte Orte vorab zu identifizieren und diese in die Simulation zu integrieren und zum Anderen verschiedene Algorithmen anzuwenden, um unterschiedliche Strategien bei der Wegefindung abzubilden.



## Kapitel 8

# Zusammenfassung und Ausblick

Die Modellierung und Simulation von Fußgängerverhalten spielt eine wichtige Rolle bei der Planung von Gebäuden, bei der Unterstützung von Sicherheitskräften zur Planung und Durchführung von Großveranstaltungen, für Kurzzeitprognosen basierend auf Videoaufnahmen sowie zur Berechnung von Evakuierungszeiten von Gebäuden, Flugzeugen, Schiffen oder ganzen Regionen. In den letzten Jahren wurden eine Reihe unterschiedlicher Modelle entwickelt, um den verschiedenen Anforderungen des breiten Einsatzgebietes bestmöglich gerecht zu werden. Modelle zur Fußgängersimulation bestehen in der Regel aus drei verschiedene Modellierungsebenen. Auf der Lokomotionsebene wird das lokale Laufverhalten abgebildet, die taktische Ebene modelliert globale Planungsprozesse sowie Wegfindungsstrategien und die strategische Ebene dient zur Definition des Grund Sachverhalts, aus dem Parameter für die Simulation abgeleitet werden.

In dieser Arbeit wurde die taktische Ebene näher betrachtet. Es wurden Methoden zur Abbildung von Wegfindungsstrategien bei Fußgängern vorgestellt, um globale Planungsprozesse von Fußgängern realitätsnah abzubilden und damit mikroskopische Modelle, deren Fokus bisher auf lokalem Verhalten lag, zu erweitern. Dazu wurde ein bestehendes mikroskopisches Simulationsmodell um die taktische Ebene ergänzt. Auf Basis eines Navigationsgraphen wurden unterschiedliche Algorithmen vorgestellt und bewertet, die die kognitiven Prozesse bei der Wegfindung unter Einbeziehung der Umgebung, des Orientierungssinns und Ortskenntnis abbilden.

### 8.1 Zusammenfassung der Arbeit

Die Arbeit hat im Wesentlichen vier Schwerpunkte:

**Automatisiertes Erstellen eines Navigationsgraphen aus einer Szenariogeometrie** – Als Grundlage zur Modellierung der taktischen Ebene kann ein Sichtbarkeitsgraph eingesetzt werden. Dieser Sichtbarkeitsgraph soll auf der einen

Seite so ausgedünnt wie möglich sein, gleichzeitig soll er automatisiert aus einer Szenariogeometrie zu erstellen sein, um auch große komplexe Szenarien modellieren zu können. Die Forderung nach einem ausgedünnten Graphen resultiert aus den auf ihm implementierten Routingalgorithmen, deren Performanz von der Anzahl der Kanten und Knoten des Graphen abhängt.

Um diese beiden Voraussetzungen zu erfüllen, wurde ein kegelbasierter Algorithmus eingeführt. Dieser Algorithmus erstellt einen Sichtbarkeitsgraph, der auf der einen Seite so ausgeünnt wie möglich und auf der anderen Seite detailliert genug ist, um als Wegfindungsbasis zu dienen. In einer Studie wurden die so erstellten Graphen mit vollständigen Sichtbarkeitsgraphen verglichen. Die Ergebnisse belegen, dass der ausgedünnte Graph vor allem bei komplexen Szenarien um bis zu 80 % weniger Kanten enthält als ein vollständiger Sichtbarkeitsgraph bei gleichzeitig gegebener räumlicher Gebietsabdeckung. Somit eignet sich dieser Graph als Grundlage zur Abbildung menschlichen Wegfindungsverhaltens sehr gut.

**Verfahren zur Kopplung von mikroskopischen und makroskopischen Simulationsmodellen** – Evakuierungszeiten, die von mikroskopischen und makroskopischen Simulationsmodellen berechnet werden, unterscheiden sich in der Regel erheblich. Dies liegt daran, dass makroskopische Verfahren mikroskopische Effekte wie Interaktionen zwischen einzelnen Personen nicht miteinbeziehen. Daraus resultieren zu optimistische untere Schranken für die Evakuierungszeit. Auf der anderen Seite überschätzen mikroskopische Ansätze reale Evakuierungszeiten, da global optimale Routingentscheidungen nicht mit einfließen.

Um eine Annäherung der Zeiten zu erreichen, wurde ein neues Verfahren vorgestellt, das beide Ansätze auf Basis eines Regelkreises miteinander kombiniert. Grundlage hierzu bildet der Navigationsgraph, der für dieses Verfahren zu einem Netzwerk erweitert wurde. Mit dem Verfahren konnten die vom makroskopischen Modell berechneten unteren Schranken für Evakuierungszeiten angehoben werden, da mikroskopische Effekte miteinbezogen werden. Die Anhebung der Evakuierungszeiten führt zu einer verbesserten Abschätzung der berechneten unteren Schranken.

Auf der anderen Seite konnten die vom mikroskopischen Modell berechneten Evakuierungszeiten präziser angenähert werden, indem global optimale Routingentscheidungen aus dem makroskopischen Modell mit einfließen. Diese optimalen Routingentscheidungen repräsentieren Entscheidungen von Sicherheitskräften bei Veranstaltungen, die über globales Wissen verfügen, und deshalb Personen optimal leiten können.

Das Verfahren wurde an schematischen Testfällen überprüft und in einer leicht abgewandelten Form, der sog. „minimalen Kopplung“ an einem komplexeren Realbeispiel demonstriert. Die Ergebnisse zeigen eine Annäherung der berechneten Evakuierungszeiten auf weniger als 50 % der ursprünglichen Abweichungen und damit eine erheblich genauere Abschätzung dieser Zeiten.

Daraus kann geschlossen werden, dass eine solche Kopplung Einsatz in der Planung und Durchführung von Großveranstaltungen finden kann.

**Algorithmen zur Abbildung von menschlichem Wegfindungsverhalten** – Die Abbildung und Integration von realistischem menschlichen Wegfin-

dungsverhalten in mikroskopische Simulationsmodelle ist die Voraussetzung für aussagekräftige Simulationsergebnisse. Bisherige mikroskopische Simulationsmodelle wenden - wenn überhaupt - sehr einfache Wegefindungsalgorithmen an, die das Spektrum der Aspekte von menschlichem Wegefindungsverhalten kaum abdecken.

Deshalb wurden in dieser Arbeit kognitive Prozesse, die in der Wegefindung stattfinden, in Algorithmik überführt. Es wird zwischen ortskundigen und ortsfremden Personen unterschieden, deren Verhalten jeweils in Normalsituationen und im Evakuierungsfall modelliert wird.

Zur Wegefindung verwenden Personen kognitive Karten. Diese beinhalten unter anderem Landmarks zur Orientierung. Landmarks bezeichnen bekannte Orte, die entweder individuell gewählt werden oder allgemein bekannt sind. Dieses Konzept wurde in die entwickelten Algorithmen integriert.

Die Algorithmen wurden zur Validierung mit Daten aus einem Experiment verglichen. Die Simulationsergebnisse stimmen visuell zu weiten Teilen mit den von den Personen gelaufenen Wegen überein. Bei der Ähnlichkeitsbetrachtung konnten die Ergebnisse der Simulation gute Werte erzielen, auch im Vergleich zu konventionellen Routingalgorithmen. Häufig gelaufene Teilstrecken wurden von den Simulationsergebnissen ebenfalls reproduziert.

Insgesamt konnte gezeigt werden, dass der Einsatz unterschiedlicher Wegefindungsalgorithmen maßgeblich zur Verbesserung der Simulationsergebnisse beiträgt und dass Personen unterschiedliche Strategien zur Wegefindung einsetzen.

**Definition einer Metrik zur Validierung der taktischen Ebene** – Bisher existierten keine Metriken und kaum Realdaten, um eine aussagekräftige Validierung der taktischen Ebene durchzuführen. Deshalb wurde zur Validierung der entwickelten Wegefindungsalgorithmen ein neues Konzept eingeführt. Basierend auf experimentellen Daten, die im Rahmen eines Experiments an der Technischen Universität München erhoben wurden, wurden mehrere Arten von Vergleichen durchgeführt: Zusätzlich zum visuellen Vergleich wurde eine neue Metrik „similarity“ zur Ähnlichkeitsbetrachtung eingeführt. Diese Metrik kann dazu eingesetzt werden, Routen, die im Experiment gelaufen wurden mit den von der Simulation berechneten Routen systematisch zu vergleichen. Ein dritter Vergleich zielt auf die Häufigkeit überquerter Teilstrecken ab.

Anhand einer ausführlichen Vergleichsstudie zur Validierung der in dieser Arbeit entwickelten Algorithmen konnte die Aussagekraft der Metriken belegt werden. Damit kann zukünftig die taktische Ebene durch ein Standardverfahren validiert werden.

Insgesamt tragen die vier in dieser Arbeit vorgestellten Schwerpunkte zu einer Verbesserung der Ergebnisse von mikroskopischen Simulationsmodellen bei. Durch die automatisierte Erzeugung eines ausgedünnten Sichtbarkeitsgraphen wurde die Grundlage für die Implementierung von Algorithmen zur Abbildung menschlichen Wegefindungsverhaltens geschaffen. Diese Algorithmen konnten durch Einführung einer aussagekräftigen Vergleichsmetrik validiert werden. Zudem ermöglicht der eingeführte ausgedünnte Navigationsgraph eine Kombina-

tion von mikroskopischen Modellen mit makroskopischen Netzwerkflussmodellen, um realistischere untere Schranken für Evakuierungszeiten zu ermitteln.

## 8.2 Ausblick

Die in dieser Arbeit vorgestellten Algorithmen bilden bereits ein breites Spektrum an menschlichem Wegefindungsverhalten ab. Es konnte jedoch nur ein Teil der Algorithmen mit einem Experiment validiert werden. Insbesondere für Evakuierungssituationen ist es schwierig, Realdaten zur Validierung zu erhalten. Auch ist festzuhalten, dass die Einstellung der korrekten Landmarks und der richtigen Parameter bis dato ein manueller Prozess ist. Hier ist es sinnvoll – ähnlich zur Kalibrierung der Lokomotionsebene – diese Parameter gegenüber Realdaten zunächst zu kalibrieren, um den manuellen Prozess abzulösen.

Ein ganz anderer Aspekt, der in das menschliche Wegefindungsverhalten mit einfließt, ist die Absicht der Bewegung. Menschen laufen in ihrer Freizeit mitunter nicht zielorientiert. Dies kann mit den bisherigen Algorithmen nicht abgedeckt werden. Dazu werden agentenbasierte Ansätze benötigt, die Personen individuell mit unterschiedlichen Charakteren ausstatten. Die Kombination der hier vorgestellten Wegefindungsalgorithmen mit agentenbasierten Ansätzen kann diese Lücke schließen.

Diese Arbeit stellt Methoden zur Abbildung von menschlichem Wegefindungsverhalten vor. Eine hundertprozentige Voraussage des Verhaltens ist dennoch nach dem aktuellen Stand der Wissenschaft nicht möglich. Daher stellt die Erforschung von menschlichem Verhalten unter Miteinbeziehung unterschiedlicher wissenschaftlicher Disziplinen auch in Zukunft eine spannende wissenschaftliche Herausforderung dar.

---

# Literaturverzeichnis

- Aguirre, B. E., El-Tawil, S., Best, E., Gill, K. B. & Fedorov, V. (2011). Contributions of social science to agent-based models of building evacuation. *Contemporary Social Science* 6(3), S. 415–432.
- Ahuja, R., Mananti, T. & Orlin, J. (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall.
- Allen, G. L. (1999). Spatial Abilities, Cognitive Maps, and Wayfinding: Bases for Individual Differences in Spatial Cognition and Behavior. In: R. G. Golledge (Hrsg.), *Wayfinding behavior*, S. 46–81. Baltimore: Johns Hopkins Univ. Press.
- Angus, D. Solving a unique Shortest Path problem using Ant Colony Optimisation.
- Antonini, G., Bierlaire, M. & Weber, M. (2006). Discrete choice models of pedestrian walking behavior. *Transportation Research Part B: Methodological* 40(8), S. 667–687.
- Arikan, O., Chenney, S. & Forsyth, D. A. (2001). Efficient multi-agent path planning. In: *In Proceedings of the 2001 Eurographics Workshop on Animation and Simulation*, S. 151–162.
- Arkin, E. M., Chew, L. P., Huttenlocher, D. P., Kedem, K. & Mitchell, J. S. B. (1991). An Efficiently Computable Metric for Comparing Polygonal Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13, S. 209–216.
- Aurenhammer, F. (1991). Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure. *ACM Comput. Surv.* 23, S. 345–405.
- Bäck, T. (1996). *Evolutionary algorithms in theory and practice: Evolution strategies, evolutionary programming, genetic algorithms*. New York: Oxford Univ. Press.
- Ballard, D. H. & Brown, C. M. (1982). *Computer vision*. Englewood Cliffs and N.J: Prentice-Hall.
- Bayer, R. & McCreight, E. (1972). Organization and Maintenance of Large Ordered Indexes. *Acta Informatica* 1, S. 173–189.

- Beckmann, N., Kriegel, H.-P., Schneider, R. & Seeger, B. (1990). The R\*-tree: an efficient and robust access method for points and rectangles. *SIGMOD Rec.* 19(2), S. 322–331.
- Berg, M. d., van Kreveld, M. & Overmars, M. H. (2000). Chapter 15: Visibility Graphs. In: O. Schwarzkopf (Hrsg.), *Computational geometry*, S. 307–317. Berlin: Springer.
- Blue, V. & Adler, J. (1999). Cellular Automata Microsimulation of Bidirectional Pedestrian Flows. *Transportation Research Record: Journal of the Transportation Research Board* 1678, S. 135–141.
- Blue, V., Embrechts, M. & Adler, J. (1997). Cellular automata modeling of pedestrian movements. *IEEE International Conference on Systems, Man and Cybernetics*, S. 2320.
- Borrmann, A., Kneidl, A., Köster, G., Ruzika, S. & Thiemann, M. (2012). Bidirectional coupling of macroscopic and microscopic pedestrian evacuation models. *Safety Science* 50(8), S. 1695–1703.
- Brogan, D. C. & Johnson, N. L. (2003). Realistic Human Walking Paths. In: *16th International Conference on Computer Animation and Social Agents*, S. 94–101. Los Alamitos and California: IEEE Computer Society.
- Bungartz, H.-J., Zimmer, S., Buchholz, M. & Pflüger, D. (2009). *Modellbildung und Simulation: Eine anwendungsorientierte Einführung*. Berlin: Springer-Verlag.
- Burkard, R., Dlaska, K. & Klinz, B. (1993). The quickest flow problem. *Mathematical Methods of Operations Research* 37(1), S. 31–58.
- Burstedde, C., Klauck, K., Schadschneider, A. & Zittartz, J. (2001). Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications* 295(3-4), S. 507–525.
- Butenuth, M., Burkert, F., Kneidl, A., Borrmann, A., Schmidt, F., Hinz, S. & Sirmacek, B. (2011). Integrating pedestrian simulation, tracking and event detection for crowd analysis. In: *1st IEEE Workshop on Modeling, Simulation and Visual Analysis of Large Crowds*.
- Chalmet, L., Francis, R. & Saunders, P. (1982). Network Models for Building Evacuation. *Management Science* 28, S. 86–105.
- Choset, H. M. (2005). *Principles of robot motion: Theory, algorithms, and implementation*. A Bradford book. Cambridge and Mass.: MIT Press.
- Chraïbi, M. (2012). *Validated force-based modeling of pedestrian dynamics*. Dissertation, Forschungszentrum Jülich, Jülich.
- Chraïbi, M., Wagoum, U. K., Schadschneider, A. & Seyfried, A. (2011). Force-based models of pedestrian dynamics. *Networks and Heterogeneous Media* 6, S. 425–442.



- Conroy, R. (2001). *Spatial Navigation in Immersive Virtual Environments*. Dissertation, The Faculty of Built Environment, London.
- Cormen, T. H. (2010). *Algorithmen - eine Einführung* (3 Aufl.). München [u.a.]: Oldenbourg.
- Dalton, C. R. (2001). The secret is to follow your nose. In: Wineman & Peponis (Hrsg.), *Proceedings of the 3rd International Symposium on Space Syntax*, S. 47.1 – 47.14. Atlanta.
- Davidich, M. & Köster, G. (2011). A Methodological Approach to Adjusting Pedestrian Simulations to Real Life Scenarios: a German Railway Station Example. In: *Proceedings of Traffic and Granular Flow (TGF)*. Moscow and Russia.
- Davidich, M. & Köster, G. (2012a). Towards automatic and robust adjustment of human behavioral parameters in a pedestrian stream model to measured data. *Safety Science* 50(5), S. 1253–1260.
- Davidich, M. & Köster, G. (2012b). Verification of Pedestrian Stream Models Based on Video Analysis. In: *Proceedings of the 6th International Conference on Pedestrian and Evacuation Dynamics*.
- Dickmeiß, T. (2005). *Zur graphtheoretischen Dilation der Delaunay-Triangulation und verwandter Graphen*. Diplomarbeit, Rheinische Friedrich-Wilhelms-Universität Bonn.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik* 1(1), S. 269–271.
- Dijkstra, J., Jessurun, A. J., Vries, B. d. & Timmermans, H. J. P. (2006). Agent Architecture for Simulating Pedestrians in the Built Environment. In: A. L. Bazzan, B. Chaib-draa, F. Klügl, & S. Ossowski (Hrsg.), *Forth International Workshop on Agents in Traffic and Transportation (ATT2006)*, S. 8–16. New York.
- Dorigo, M. (1992). *Optimization learning and natural algorithms*. Dissertation, Politecnico di Milano, Mailand.
- Downs, R. M. & Stea, D. (1977). *Maps in minds: Reflections on cognitive mapping*. New York: Harper & Row.
- Dressler, D., Groß, M., Kappmeier, J., Kelter, T., Kulbatzki, J., Plümpe, D., Schlechter, G., Schmidt, M., Skutella, M. & Temme, S. (2010). On the use of network flow techniques for assigning evacuees to exits. *Procedia Engineering* 3, S. 205–215.
- Driesel, M. (2012). *Simulation von Evakuierungsszenarien bei Großveranstaltungen am Beispiel eines Gebäudes der TU München*. Bachelorarbeit, Technische Universität München, München.

- Durupinar, F., Allbeck, J. M., Pelechano, N. & Badler, N. (2008). Creating crowd variation with the OCEAN personality model. In: *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 3*, AAMAS '08, Richland and SC, S. 1217–1220. International Foundation for Autonomous Agents and Multiagent Systems.
- Fukui, M. & Ishibashi, Y. (1999). Self-organized phase transitions in cellular automaton models for pedestrians. *J. Phys. Soc. Japan* 68(8), S. 2861.
- Funge, J., Tu, X. & Terzopoulos, D. (1999). Cognitive modeling: knowledge, reasoning and planning for intelligent characters. In: *Proceedings of the 26th annual conference on computer graphics and interactive techniques*, SIGGRAPH '99, New York and NY and USA, S. 29–38. ACM Press/Addison-Wesley Publishing Co.
- Geraerts, B. R. & Overmars, M. H. (2007). The corridor map method: a general framework for real-time high-quality path planning. *Computer Animation and Virtual Worlds* 18(2), S. 107.
- Gibbons, A. (1999). *Algorithmic graph theory* (Reprinted, transferred to digital reprinting. Aufl.). Cambridge: Cambridge Univ. Press.
- Gloor, C., Stucki, P. & Nagel, K. (2004). Hybrid Techniques for Pedestrian Simulations. In: P. Sloot, B. Chopard, & A. Hoekstra (Hrsg.), *Cellular Automata*, Volume 3305 of *Lecture Notes in Computer Science*, S. 581–590. Springer.
- Gluck, M. (1991). Making sense of human wayfinding: Review of cognitive and linguistic knowledge for personal navigation with a new research direction. In: A. U. Frank & D. M. Mark (Hrsg.), *Cognitive and linguistic aspects of geographic space*, S. 117–135. Dordrecht: Kluwer.
- Golledge, R. G. (1995). Path selection and route preference in human navigation: A progress report. In: A. U. Frank (Hrsg.), *Spatial information theory*, Volume 988 of *Lecture Notes in Computer Science*, S. 207–222. Berlin: Springer.
- Golledge, R. G. (1999a). Human Wayfinding and Cognitive Maps. In: R. G. Golledge (Hrsg.), *Wayfinding behavior*, S. 5–45. Baltimore: Johns Hopkins Univ. Press.
- Golledge, R. G. (Hrsg.) (1999b). *Wayfinding behavior: Cognitive mapping and other spatial processes*. Baltimore: Johns Hopkins Univ. Press.
- Graessle, F. & Kretz, T. (2011). An Example of Complex Pedestrian Route Choice. In: R. D. Peacock, E. D. Kuligowski, & J. D. Averill (Hrsg.), *Pedestrian and evacuation dynamics, 2011*, Berlin. Springer.
- Guttman, A. (1984). R-trees: A Dynamic Index Structure for Spatial Searching. In: *International Conference on Management of Data*, S. 47–57. ACM.
- Hamacher, H., Heller, S., Köster, G. & Klein, W. (2011). A Sandwich Approach for Evacuation Time Bounds. In: R. D. Peacock, E. D. Kuligowski, & J. D. Averill (Hrsg.), *Pedestrian and evacuation dynamics, 2011*, Berlin. Springer.

- Hamacher, H. W. & Tjandra, S. A. (2002). Mathematical modelling of evacuation problems: A state of the art. In: M. Schreckenberg (Hrsg.), *Pedestrian and Evacuation Dynamics*, S. 227–266. Berlin: Springer.
- Hart, P., Nilsson, N. & Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *Systems Science and Cybernetics* 4(2), S. 100–107.
- Hartmann, D. (2010a). Adaptive pedestrian dynamics based on geodesics. *New Journal of Physics* 12(4), S. 043032.
- Hartmann, D. (2010b). SinoSim - Microscopic Models of Pedestrian Dynamics.
- Helbing, D. (1992). A fluid-dynamic model for the movement of pedestrians. *Complex Systems* 6, S. 391–415.
- Helbing, D., Johansson, A. & Al-Abideen, H. Z. (2007). Dynamics of crowd disasters: An empirical study. *Physical Review E* 75, S. 046109.
- Helbing, D. & Molnár, P. (1995). Social Force Model for Pedestrian Dynamics. *Physical Review E* 51(5), S. 4282–4286.
- Henderson, L. F. (1971). The Statistics of Crowd Fluids. *Nature* (229), S. 381–383.
- Henderson, L. F. (1974). On the fluid mechanics of human crowd motion. *Transportation Research Part B: Methodological* 8(6), S. 509–515.
- Hochmair, H. & Frank, A. U. (2003). Influence of estimation errors on wayfinding-decisions in unknown street networks - analyzing the least-angle strategy. In: *Spatial cognition and computation*, Volume 2, S. 283–313. Mahwah: Lawrence Erlbaum.
- Höcker, M. (2010). *Modellierung und Simulation von Fußgängerverkehr*. Dissertation, Fakultät für Bauingenieurwesen und Geodäsie, Gottfried Wilhelm Leibniz Universität Hannover.
- Höcker, M., Berkahn, V., Kneidl, A., Borrmann, A. & Klein, W. (2010). Graph-based approaches for simulating pedestrian dynamics in building models. In: University College Cork (Hrsg.), *8th European Conference on Product & Process Modelling (ECPPM)*. Cork and Ireland.
- Hoogendoorn, S. P. & Bovy, P. H. L. (2004). Pedestrian route-choice and activity scheduling theory and models. *Transportation Research Part B: Methodological* 38(2), S. 169–190.
- Huang, H.-J. & Guo, R.-Y. (2008). Static floor field and exit choice for pedestrian evacuation in rooms with internal obstacles and multiple exits. *Physical Review E* 78(2), S. 021131.
- Hughes, R. L. (2000). The flow of large crowds of pedestrians. *Math. Comput. Simul.* 53(4-6), S. 367.

- Hughes, R. L. (2003). The flow of human crowds. *Annual Review of Fluid Mechanics* 35(1), S. 169–182.
- Kinkeldey, C. (2003). *Fußgängersimulation auf der Basis zellularer Automaten*. Dissertation, Universität Hannover, Hannover.
- Kirik, E., Yurgel'yan, T. B. & Krouglov, D. (2009). The shortest time and/or the shortest path strategies in a CA FF pedestrian dynamics model. *J. Siberian Fed. Univ. Math. Phys.* 2(3), S. 271–278.
- Klein, W., Köster, G. & Meister, A. (2010). Towards The Calibration of Pedestrian Stream Models. In: J. Weglarz, R. Wyrzykowski, & B. Szymanski (Hrsg.), *8th Int. Conf. on Parallel Processing and Applied Mathematics*, Lecture Notes in Computer Science, Berlin. Springer.
- Klüpfel, H. (2006). The simulation of crowd dynamics at very large events - Calibration, empirical data, and validation. In: P. Gattermann, N. Waldau, & M. Schreckenberg (Hrsg.), *Proceedings of the 3rd International Conference on Pedestrian and Evacuation Dynamics*, Berlin. Springer.
- Kneidl, A. & Borrmann, A. (2011). How Do Pedestrians find their Way? Results of an experimental study with students compared to simulation results. In: W. Jaskolowski & P. Kepka (Hrsg.), *EMEVAC*. Warsaw: The Main School of Fire Service.
- Kneidl, A., Borrmann, A. & Hartmann, D. (2010). Einsatz von graphbasierten Ansätzen in einer mikroskopischen Personenstromsimulation für die Wegewahl der Fußgänger. In: T. Krämer, S. Richter, F. Enge, & B. Kraft (Hrsg.), *Forum Bauinformatik 2010*, Volume Bd. 9 of *Heftreihe des Instituts für Bauingenieurwesen*. Aachen: Shaker.
- Kneidl, A., Borrmann, A. & Hartmann, D. (2011). Generating sparse navigation graphs for microscopic pedestrian simulation models. In: T. Hartmann, P. de Wilde, & Y. Rafiq (Hrsg.), *Proceedings of the 2011 eg-ice Workshop*.
- Kneidl, A., Hartmann, D. & Borrmann, A. (2012). Generation and use of sparse navigation graphs for microscopic pedestrian simulation models. *Advanced Engineering Informatics* 26, S. 669–680.
- Kneidl, A., Thiemann, M., Borrmann, A., Ruzika, S., Hamacher, H. W., Köster, G. & Rank, E. (2011). Bidirectional Coupling of Macroscopic and Microscopic Approaches for Pedestrian Behavior Prediction. In: R. D. Peacock, E. D. Kuligowski, & J. D. Averill (Hrsg.), *Pedestrian and evacuation dynamics, 2011*, Berlin. Springer.
- Kneidl, A., Thiemann, M., Hartmann, D. & Borrmann, A. (2011). Combining pedestrian simulation with a network flow optimization to support security staff in handling an evacuation of a soccer stadium. In: *Proceedings of European Conference Forum 2011, Cork*. Cork and Ireland.

- Köster, G., Hartmann, D. & Klein, W. (2010). Microscopic pedestrian simulations: From passenger exchange times to regional evacuation. In: *Operations Research - Mastering complexity*.
- Köster, G., Seitz, M., Treml, F., Hartmann, D. & Klein, W. (2011). On modelling the influence of group formations in a crowd: *Contemporary Social Science. Contemporary Social Science* 6(3), S. 397–414.
- Kretz, T. (2009). Pedestrian traffic: on the quickest path. *Journal of Statistical Mechanics: Theory and Experiment* 2009(03), S. P03012.
- Kretz, T., Bönisch, C. & Vortisch, P. (2008). Comparison of various methods for the calculation of the distance potential field. *Pedestrian and Evacuation Dynamics 2008*.
- Krötzing, W. (2011). *Erweiterung von Navigationsgraphen um Landmarks zur Abbildung der Fußgängerwegewahl in einem Personenstromsimulator*. Bachelorarbeit, Technische Universität München, München.
- Kuipers, B. (1983). The Cognitive Map: Could it have been any other way? In: H. L. Pick & L. P. Acredolo (Hrsg.), *Spatial Orientation: Research, Theory and Application*, S. 345–359. New York: Plenum Press.
- Langston, P. A., Masling, R. & Asmar, B. N. (2006). Crowd dynamics discrete element multi-circle model. *Safety Science* 44(5), S. 395–417.
- Lawton, C. A. & Kallai, J. (2002). Gender Differences in Wayfinding Strategies and Anxiety About Wayfinding: A Cross-Cultural Comparison. *Sex Roles* 47(9/10), S. 389–401.
- Lerner, A., Chrysanthou, Y. & Lischinski, D. (2007). Crowds by Example. *Computer Graphics Forum* 26(3), S. 655–664.
- Loomis, J. M., Klatzky, R. L., Golledge, R. G., Cicinelli, J. G., Pellegrino, J. W. & Phyllis, F. A. (1993). Nonvisual Navigation by blind and sighted: Assessment of path integration ability. *Journal of Experimental Psychology: General* 122(1), S. 73–91.
- Lynch, K. (1960). *The image of the city* (5 Aufl.). Cambridge and Mass: MIT Press.
- Mainzer, K. & Chua, L. O. (2012). *The universe as automaton: From simplicity and symmetry to complexity*. Heidelberg and New York: Springer.
- Malcher, A. (2004). *Beschreibungskomplexität von Zellularautomaten*. Dissertation, Johann Wolfgang Goethe-Universität, Frankfurt am Main.
- Meschede, D. (2006). *Gerthsen Physik* (23 Aufl.). Berlin: Springer.
- Montello, D. R. (1991). Spatial Orientation and the Angularity of Urban Routes: A Field Study. *Environment and Behavior* 23(3), S. 47–69.

- Moussaïd, M. & Helbing, D. (2011). How simple rules determine pedestrian behavior and crowd disasters. *Proceedings of the National Academy of Sciences* 108(17), S. 6884–6888.
- Moussaïd, M., Helbing, D., Garnier, S., Johansson, A., Combe, M. & Theraulaz, G. (2009). Experimental study of the behavioural mechanisms underlying self-organization in human crowds. *Proceedings of the Royal Society B- Biological Sciences* 276(1668), S. 2755–2762.
- Pelechano, N., Allbeck, J. M. & Badler, N. (2008). *Virtual crowds: Methods, simulation, and control*. [San Rafael and Calif.]: Morgan & Claypool Publishers.
- Pelechano, N. & Malkawi, A. (2008). Evacuation simulation models: Challenges in modeling high rise building evacuation with cellular automata approaches. *Automation in Construction* 17(4), S. 377–385.
- Peng, L. (2006). *A Dynamic Network Flow Optimization for Large-Scale Emergency Evacuation*. Dissertation, City University of Hong Kong, Hong Kong.
- Petres, C., Pailhas, Y., Patron, P., Petillot, Y., Evans, J. & Lane, D. (2007). Path Planning for Autonomous Underwater Vehicles. *IEEE Transactions on Robotics* 23, S. 331–341.
- Philbeck, J. W., Loomis, J. M. & Beall, A. C. (1997). Visually perceived location is an invariant in the control of action. *Perception & Psychophysics* 59(4), S. 601–612.
- Portz, A. & Seyfried, A. (2011). Analyzing Stop-and-Go Waves by Experiment and Modeling. In: R. D. Peacock, E. D. Kuligowski, & J. D. Averill (Hrsg.), *Pedestrian and Evacuation Dynamics*, S. 577–586. Springer US.
- Reynolds, C. (1999). Steering Behaviors for Autonomous Characters. In: *Game Developers Conference*.
- Rindsfuser, G. & Klügl, F. (2007). Agent-Based Pedestrian Simulation: A Case Study of the Bern Railway Station. In: K. Nagel & M. Koll-Schretzenmayr (Hrsg.), *disP*, Volume 170, S. 9–18. Zurich and Switzerland: ETH Zürich.
- Rodriguez, S. & Amato, N. M. (2010). Behavior-Based Evacuation Planning. In: *Proceedings of 2010 IEEE International Conference on Robotics and Automation*, Anchorage.
- Sadalla, E. K. & Montello, D. R. (1989). Remembering Changes in Direction. *Environment and Behavior* 21(3), S. 346–363.
- Samet, H. (1990). *The design and analysis of spatial data structures*. Addison-Wesley.
- Schadschneider, A., Klingsch, W., Klüpfel, H., Kretz, T., Rogsch, C. & Seyfried, A. (2009). Evacuation dynamics: Empirical results, modeling and applications. *Encyclopedia of Complexity and System Science*, S. 3142–3176.

- Seitz, M. & Köster, G. (2012). Natural discretization of pedestrian movement in continuous space. *Physical Review E* 86(4), S. 046108 – 046116.
- Sethian, J. A. (1999). *Level Set Methods and Fast Marching Methods*. Cambridge University Press.
- Seyfried, A., Steffen, B., Winkens, A., Rupprecht, T., Boltes, M. & Klingsch, W. (2009). Empirical Data for Pedestrian Flow Through Bottlenecks. In: C. Appert-Rolland, F. Chevoir, P. Gondret, S. Lassarre, J.-P. Lebacque, & M. Schreckenberg (Hrsg.), *Traffic and Granular Flow '07*, S. 189–199. Springer.
- Sterlin, P., Deconinck, F., Rakotaorison, T. & Buendia, A. (2010). Comfort driven navigation of individuals in a crowd simulation. In: *Computer Animation and Virtual Worlds*.
- Sud, A., Andersen, E., Curtis, S., Lin, M. C. & Manocha, D. (2008). Real-Time Path Planning in Dynamic Virtual Environments Using Multiagent Navigation Graphs. *IEEE Transactions on Visualization and Computer Graphics* 14, S. 526–538.
- Teknomo, K. (2008). Modeling mobile traffic agents on network simulation. In: *Transportation Science Society of the Phillippines 2008 – Proceedings of the 16th Annual of Transportation Science*.
- Teknomo, K. & Millonig, R. (2007). A navigation algorithm for pedestrian simulation in dynamic environments. In: *Proceedings of the 11th World Conference on Transport Research (WCTR)*, Berkeley and USA.
- Thiemann, M. (2012). *Cross-linking of robustness, parameter computation, and optimization in evacuation modeling* (1 Aufl.). München: Verl. Dr. Hut.
- Thorndyke, P. W. & Hayes-Roth, B. (1982). Differences in spatial knowledge acquired from maps and navigation. *Cognitive Psychology* 14(4), S. 560–589.
- Tjandra, S. (2003). *Dynamic Network Optimization with Application to the Evacuation Problem*. Dissertation, TU Kaiserslautern, Kaiserslautern.
- Tolman, E. C. (1948). Cognitive Maps in Rats and Men. *The Psychological Review* 55(4), S. 189–208.
- Tom, A. & Denis, M. (2003). Referring to Landmark or Street Information in Route Directions: What Difference Does It Make? In: W. Kuhn (Hrsg.), *Spatial Information Theory*, Volume 2825 of *Lecture Notes in Computer Science*. Berlin: Springer.
- Torrens, P. M., Nara, A., Li, X., Zhu, H., Griffin, W. A. & Brown, S. B. (2012). An extensible simulation environment and movement metrics for testing walking behavior in agent-based models. *Computers, Environment and Urban Systems* 36(1), S. 1–17.
- Treuille, A., Cooper, S. & Popović, Z. (2006). Continuum crowds. *ACM Trans. Graph* 25(3), S. 1160–1168.

- Varas, A., Cornejo, M. D., Mainemer, D., Toledo, B., Rogan, J., Munoz, V. & Valdivia, J. A. (2007). Cellular automaton model for evacuation process with obstacles. *Physica A: Statistical Mechanics and its Applications* 382(2), S. 631–642.
- Weidmann, U. (1993). Transporttechnik der Fussgänger: Transporttechnische Eigenschaften des Fussgängerverkehrs (Literaturauswertung).
- Wildgruber, J. (2011). *GIS2Simulation*. Bachelorarbeit, Technische Universität München, München.
- Yamamoto, K., Kokubo, S. & Nishinari, K. (2007). Simulation for pedestrian dynamics by real-coded cellular automata. *Physica A: Statistical Mechanics and its Applications* 379(2), S. 654.
- Zhang, J., Klingsch, W., Schadschneider, A. & Seyfried, A. (2012). Ordering in bidirectional pedestrian flows and its influence on the fundamental diagram. *Journal of Statistical Mechanics: Theory and Experiment* P02002.