

FAKULTÄT FÜR INFORMATIK
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Bachelor-Thesis in Wirtschaftsinformatik

**Erweiterung einer web-basierten Postvisualisierung
zur Darstellung und Analyse von Fußgängersimula-
tionsergebnissen**

Minh-Quang Nguyen





FAKULTÄT FÜR INFORMATIK
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Bachelor-Thesis in Wirtschaftsinformatik

**Erweiterung einer web-basierten Postvisualisierung
zur Darstellung und Analyse von Fußgängersimula-
tionsergebnissen.**

**Extension of a web-based post-visualization for dis-
play and analysis of pedestrian simulation results.**

Bearbeiter:	Minh-Quang Nguyen
Aufgabensteller:	Prof. Dr. rer. nat. Florian Matthes Prof. Dr.-Ing. André Bormann
BetreuerIn:	Dipl.-Inf. Angelika Kneidl B.Sc. Mustafa Isik
Abgabedatum:	11.Oktober.2012



Ich versichere, dass ich diese Abschlussarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

Garching b. München, den

Ort, Datum

Unterschrift

Zusammenfassung

„Das Gedränge sei immer dichter geworden, Menschen seien in der Menge kollabiert, in Panik geraten, hätten geschrien, versucht, den Kopf oben zu behalten. Luft zu bekommen.“ Schrieb der Spiegel am 24.07.2010, als es in Duisburg bei der Love-Parade in einem 300 Meter langen Tunnel zu einer Massenpanik kam. Solche Zwischenfälle zeigen, wie wichtig es ist, Verkehrsanlagen so zu gestalten, dass bei Extremsituationen die Sicherheit immer noch gewährleistet ist. Ob Katastrophen oder Großveranstaltungen - wenn große Menschenmassen zusammen kommen, müssen die Leistungskapazitäten der Fußgängeranlagen groß genug sein, eine Massenpanik zu vermeiden. Eine gute Methode, um Verkehrsanlagen auf ihre Leistungskapazität zu testen oder kritische Bereiche wie Engpässe zu identifizieren, ist das Simulieren von Fußgängerströmen. Fußgängersimulationen versuchen im Ansatz möglichst realistische Fußgängerströme und -verhalten darzustellen, um Informationen für die Planung von Verkehrsanlagen herzuleiten. Bei diesen Simulationen spielt das menschliche Verhalten eine große Rolle, das noch schwer vorhersehbar ist. Deshalb müssen Erfahrungen mehrerer Fachrichtungen wie Physik, Psychologie, Physiologie, Soziologie usw. vereint werden, um aussagekräftige Modelle zu entwickeln, die ausgereift genug sind, bezugsnahe Simulationen zu erzeugen. Diese Abschlussarbeit beschäftigt sich neben Modellansätzen der Fußgängersimulation mit der Erweiterung eines web-basierten Postvisualisierungstools für die Darstellung und Analyse von Fußgängersimulationsergebnissen. Für die Erweiterung werden dem Postvisualisierungstool zusätzliche Funktionen implementiert, mit denen es möglich ist, Simulationsdaten umfangreicher zu analysieren und auszuwerten.

Inhaltsverzeichnis

Zusammenfassung.....	III
Inhaltsverzeichnis	IV
Abbildungsverzeichnis.....	V
Abkürzungsverzeichnis	VI
1 Einleitung	2
1.1 Problemstellung.....	2
1.2 Aufbau der Abschlussarbeit.....	2
2 Modellierung	3
2.1 Verhaltensebenen in der Theorie der Fußgängerdynamik	3
2.2 Einstufung der Modellierungsansätze.....	4
2.3 Bedeutung von Fußgängersimulationen.....	5
3 Postvisualisierungstool.....	7
3.1 Desktop-Tool.....	7
3.2 Web-basiert.....	9
4 Evaluationsfunktionalität	14
4.1 Datenaufbereitung für die Evaluations-Funktionen.....	14
4.2 Farbliche Darstellung der Personendichte	14
4.3 Search Person	15
4.4 Identify Person.....	17
4.5 Evaluate Rectangle.....	18
4.6 Evaluate-Rectangle-Diagram	19
4.7 Evaluate Flow	20
4.8 Evaluate Flow Diagram	23
4.9 Source / Target	24
5 Zusätzliche Funktionen.....	25
5.1 Path Direction.....	25
5.2 Zoom Funktion	25
6 Ausblick	27
Literaturverzeichnis	28

Abbildungsverzeichnis

Abbildung 1 Die verschiedenen Modell-Ebenen von Fußgänger-Verhalten	3
Abbildung 2 Darstellung verschiedener Fussgängerdichten	5
Abbildung 3 Abhängigkeit der Fussgängergeschwindigkeit von der Fussgängerdichte in der Ebene	6
Abbildung 4 Fundamental Diagramme für Fußgängerbewegungen	6
Abbildung 5 Textdatei mit Fußgängerdateien	7
Abbildung 6 Farbliche Darstellung der Personendichte	7
Abbildung 7 Abbildung des Fußgängerstroms	8
Abbildung 8 Darstellung von Diagrammen	8
Abbildung 9 SumoViz ohne evaluations Funktionen	9
Abbildung 10 Architektur von SumoViz	10
Abbildung 11 Eigene Architektur von SumoViz	11
Abbildung 12 eingelesene Fußgängerdateien in Datenbank aufbereitet	12
Abbildung 13 eingelesene Geometrydateien in Datenbank aufbereitet	12
Abbildung 14 Struktur der Datenarrays, wie sie an den Client gesendet werden.	13
Abbildung 15 Zugriff auf Daten über Index	14
Abbildung 16 Dichte-Farbstufen	15
Abbildung 17 Eingabeaufforderung für Search Person Funktion	16
Abbildung 18 zurück gelegter Weg	17
Abbildung 19 Koordinatensysteme	18
Abbildung 20 Rechteck ziehen	19
Abbildung 21 Evaluate Rectangle Funktion	19
Abbildung 22 Evaluate Rectangle Diagram Funktion	20
Abbildung 23 Eingabeaufforderung nach Flussrichtung	22
Abbildung 24 Angezeigte Werte bei Evaluate Flow	23
Abbildung 25 Flow Diagramme	23
Abbildung 26 Source / Target Funktion	24
Abbildung 27 Path Direction Funktion	25
Abbildung 28 Zoom Funktion	26

Abkürzungsverzeichnis

b	Breite
i	Index
ID	Identifikationsnummer
IVT	Institut für Verkehrsplanung, Transporttechnik, Strassen- und Eisenbahnbau
<i>J</i>	Flussdichte
J_s	spezifische Flussdichte
JSON	JavaScript Object Notation
JSONP	JavaScript Object Notation Pattern
m/s	Meter pro Sekunde
P	Person
P/m^2	Dichte
RiMEA	Richtlinien für Mikroskopische Entfluchtungs-Analysen
s. Abb.	siehe Abbildung
v	Durchschnittsgeschwindigkeit

1 Einleitung

1.1 Problemstellung

In der heutigen Zeit finden immer mehr Ereignisse statt, bei denen sich eine große Menschenmenge auf geringem Raum zusammenfindet. Büro- und Wohngebäude werden immer größer und komplexer. Großereignisse wie Sport-, Kultur- oder religiöse Veranstaltungen stellen Organisatoren vor immer größere Herausforderungen bezüglich der Sicherheitsvorkehrungen. Vor allem in Notfall- oder anderen kritische Situationen. Um zukünftig mit großen Menschenmengen angemessen umzugehen, muss deshalb die Dynamik von großen Gruppen verstanden werden. Dabei muss diese auf mehreren Ebenen, physisch, physiologisch, psychologisch und soziologisch, verstanden und beschrieben werden. Laut [Schadschneider et al., 2008], gibt es mindestens drei Aspekte der Evakuierungsdynamik, die eine wissenschaftliche Untersuchung motivieren:

1. Wie in den meisten Vielteilchensystemen können verschiedene interessante Kollektivphänomene beobachtet werden, für die eine Erklärung gefunden werden sollte.
2. Es müssen Modelle entwickelt werden, die fähig sind, die Fußgängerdynamik möglichst realistisch nachzubilden.
3. Ergebnissen der Fußgängerdynamik bei Anlagen-Design und Notfallvorbereitungen müssen angewendet werden.

Die Schwierigkeit stellen Vielteilchensysteme und deren komplexe Interaktionen, die noch nicht vollständig verstanden sind, dar. Das System ist typischerweise unausgeglichen und beinhaltet oft inhomogene Menschenansammlungen, deren Bewegung und Verhalten von vielen externen Faktoren beeinflusst werden. (Schadschneider et al. 2008, 1-2)

Für die Untersuchung dieser Thematik, spielen Simulationen eine große Rolle. Simulationen liefern zusätzliche Daten, die zur Beurteilung und Vorhersage von menschlichen Verhalten, insbesondere das Verhalten von großen Menschenmassen, helfen. Auch können verschiedene Szenarien wie Katastrophenfälle durchgespielt werden. Das ist wichtig, da empirische Beobachtungen zu realen Szenarien nur geringfügig vorhanden sind und somit die Auswertung aller beeinflussenden Parametern erschwert ist. Doch zuerst müssen Simulationen entwickelt werden, die realistische oder realitätsnahe Daten liefert. Diese Entwicklung ist nicht trivial, da viele Parameter berücksichtigt und deren Einfluss richtig zugeordnet werden müssen.

1.2 Aufbau der Abschlussarbeit

Mit dieser Abschlussarbeit werden zusätzliche Evaluationsfunktionen für ein web-basiertes Postvisualisierungstool implementiert. Dafür werden zuerst die Ebenen der Fußgängerdynamik für eine Simulation erläutert. Danach werden Merkmale von verschiedenen Modellansätzen zur Fußgängersimulationen anhand einer möglichen Klassifizierung genannt. Nach der allgemeinen Erläuterung zur Modellierung wird konkret ein Postvisualisierungstool vorgestellt. Dieses Postvisualisierungstool dient als Basis für das web-basierte Postvisualisierungstool *SumoViz*, das um zusätzliche Funktionen erweitert werden soll. Schließlich werden die zu erweiternden Funktionen einzeln vorgestellt und deren Implementierung erläutert.

2 Modellierung

2.1 Verhaltensebenen in der Theorie der Fußgängerdynamik

Eine umfassende Theorie der Fußgängerdynamik hat drei verschiedene Verhaltensebenen zu berücksichtigen. In der strategische Ebene entscheiden Fußgänger, welche Aktivitäten und in welcher Reihenfolge sie diese unternehmen wollen. Die taktische Ebene beschäftigt sich mit den kurzfristigen Entscheidungen der Fußgänger, die für die Ausführung der Aktivitäten in der strategischen Ebene gefällt wurden. Ist zum Beispiel die Aktivität von Punkt A zu Punkt B zu gelangen, wird in der taktischen Ebene festgelegt, über welchen Weg das Ziel erreicht werden soll. Die operative Ebene beschreibt das Gehverhalten des Fußgängers zum Beispiel wie Kollisionen ausgewichen werden. Die Prozesse in der strategischen und taktischen Ebene sind gewöhnlich als exogen zur Fußgängersimulation zu betrachten. Modelle in der operativen Ebene basieren normalerweise auf Variationen von physikalischen Modellen. (Schadschneider et al. 2008, 22-23)

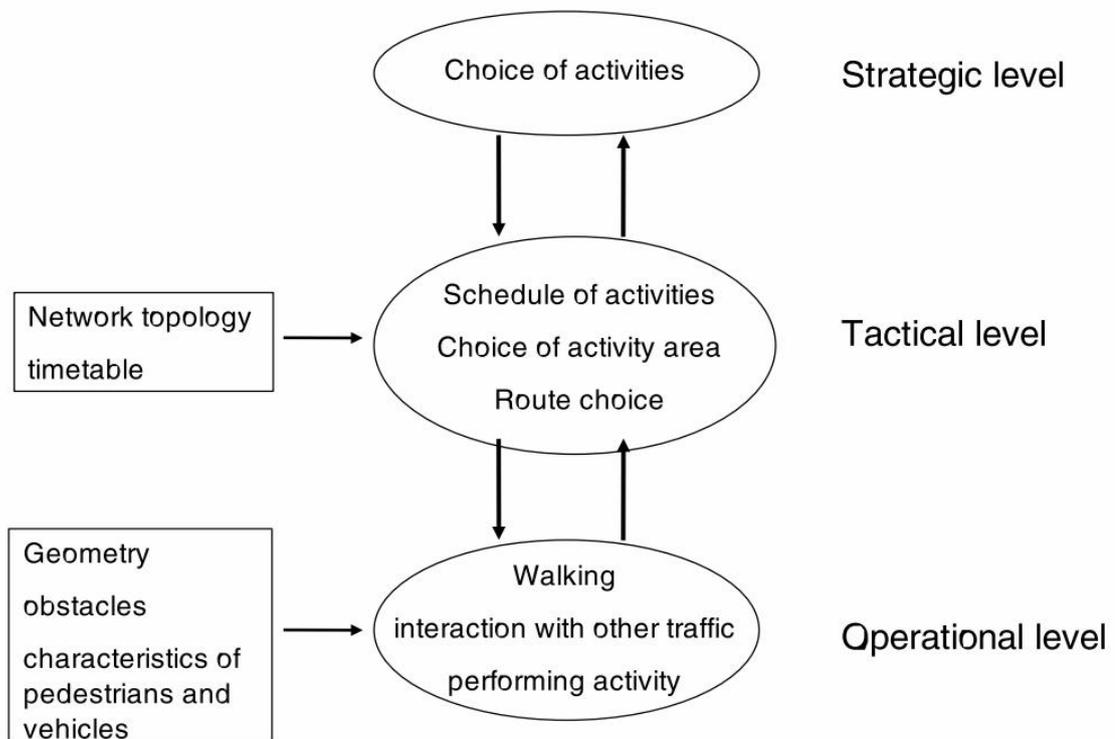


Abbildung 1 Die verschiedenen Modell-Ebenen von Fußgänger-Verhalten

Quelle: Schadschneider et al, 2008, 22

Das Ziel ist es, Modelle zu finden, die möglichst einfach sind, aber gleichzeitig realistisches Verhalten im Sinne von empirischer Beobachtung nachbilden. Deswegen werden, basierend auf den Erfahrungen aus der Physik, Fußgänger oft als einfache Teilchen, die mit einander interagieren, modelliert. (Schadschneider et al. 2008, 23)

2.2 Einstufung der Modellierungsansätze

Es gibt verschiedene Merkmale, die zur Klassifizierung von Modellierungsansätzen benutzt werden können.

Mikroskopisch – Makroskopisch

In mikroskopischen Modellen wird jedes Individuum einzeln modelliert. Dieser Ansatz erlaubt eine Einführung von verschiedenen Fußgängertypen mit individuellen Eigenschaften sowie Aspekte einer individuellen Routenwahl. Im Gegenteil dazu werden in makroskopischen Modellen nicht zwischen den verschiedenen Individuen unterschieden. Stattdessen wird der Zustand des Systems mit Dichtewerten beschrieben. In der Regel wird die Massendichte von den Positionen der Personen und den entsprechenden lokalen Durchschnittsgeschwindigkeiten abgeleitet. (Schadschneider et al. 2008, 23)

Diskret – Kontinuierlich

Jede der drei Basisvariablen für eine Beschreibung eines Fußgängersystems, nämlich Raum-, Zeit- und Statusvariable, kann entweder diskret oder stetig sein. Hier sind alle Kombinationen möglich. Beim zellulären Automaten sind im Ansatz per Definition alle Variablen diskret, während in hydrodynamischen Modellen alle stetig sind. Das sind die am häufigsten gewählten Möglichkeiten. Aber auch andere Kombinationen werden genauso verwendet. Manchmal werden für einen zellulären Automaten auch stetige Variablen verwendet. Eine diskrete Zeit wird gewöhnlich durch ein paralleles oder synchrones Update, in dem alle Teilchen oder Anlagen gleichzeitig bewegt werden, realisiert. So entsteht bildet eine Zeitskala. In sogenannten gekoppelten Kartengittern ist Zeit diskret, während Raum- und Statusvariablen stetig sind. (Schadschneider et al. 2008, 23)

Deterministisch – Stochastisch

Die Fußgängerdynamik kann entweder deterministisch oder stochastisch sein. Im ersten Fall ist das Verhalten zu einer bestimmten Zeit komplett vom aktuellen Zustand bestimmt. In stochastischen Modellen wird das Verhalten mit bestimmten Wahrscheinlichkeiten kontrolliert, die den Agenten bei gleichen Situationen ermöglicht, anders zu reagieren. In vielen anderen Beispielen in der Theorie der komplexen Systeme konnte gezeigt werden, dass durch die Einführung von Wahrscheinlichkeiten in eher simplen Systemen sehr komplizierte Verhalten generiert werden können. Auf der anderen Seite spiegeln Wahrscheinlichkeiten in Modellen auch den Mangel an Wissen über die zugrunde liegenden physikalischen Prozesse wieder. (Schadschneider et al. 2008, 23)

Regelbasiert – Einflussbasiert

Interaktionen zwischen den Agenten kann auf zwei verschiedene Arten implementiert werden. Bei einem regelbasierten Ansatz fällen Agenten ihre Entscheidungen nach ihrer aktuellen Situation in ihrer Nachbarschaft genauso nach ihren Zielen usw. Der Ansatz konzentriert sich auf die Charaktereigenschaften der Agenten, die oft aus der Psychologie gerechtfertigt sind. In einflussbasierten Modellen werden die Agenten von außen, also von Anderen und der Infrastruktur, beeinflusst. Es ist ein physikalischer Ansatz der auf Beobachtungen basiert, dass die Präsenz von Anderen zu einer Abweichung der geradlinigen Bewegung führt. Zellulare Automaten sind typischerweise regelbasierte Modelle. (Schadschneider et al. 2008, 24)

Hohe – Niedrige Genauigkeit

Die Genauigkeit bezieht sich auf den Realismus des Modellansatzes. Modelle mit hoher Genauigkeit versuchen die Komplexität von Entscheidungsfindung, Aktionen usw. zu erfassen und die Fußgängerbewegungen in einer realistischen Weise darstellen. Im Gegensatz dazu werden in simplen Modellen Fußgänger als Teilchen ohne Intelligenz dargestellt. Gewöhnlich wird das Teilchenverhalten über Einflüsse bestimmt. Dieser Ansatz kann mit einem internen Status erweitert werden. Der interne Status ermöglicht dem Fußgänger, unterschiedlich auf gleiche Einflüsse zu reagieren. Das kann als eine Art Intelligenz interpretiert werden, die zu einem komplexeren Ansatz, führt vergleichbar mit Multi-Agent Modellen. Hohe Genauigkeit bedeutet jedoch nicht unbedingt, dass empirische Beobachtungen besser nachgebildet werden können. (Schadschneider et al. 2008, 24)

Zu beachten ist, dass die Klassifizierungen nicht immer klar getrennt werden können.

2.3 Bedeutung von Fußgängersimulationen

Sind Fußgängersimulationen ausgereift genug, können sie realitätsnahe Informationen liefern, die von großem Nutzen für die Bewertung von Fußgängeranlagen sind. Computermodelle sollen vor allem extreme Situationen wie zum Beispiel Katastrophen oder ein sehr großes Fußgängeraufkommen simulieren, um Fußgängeranlagen auf ihre Leistungskapazität zu testen. Dabei werden besonders kritische Bereiche wie zum Beispiel Engstellen, Notausgänge, schmale Gänge, usw. betrachtet. Dazu werden vor allem die Dichte und die Geschwindigkeit der Fußgänger ausgewertet, die den Fußgängerfluss beeinflussen. Die Personendichte gibt dafür die Anzahl der Individuen pro Flächeneinheit an.

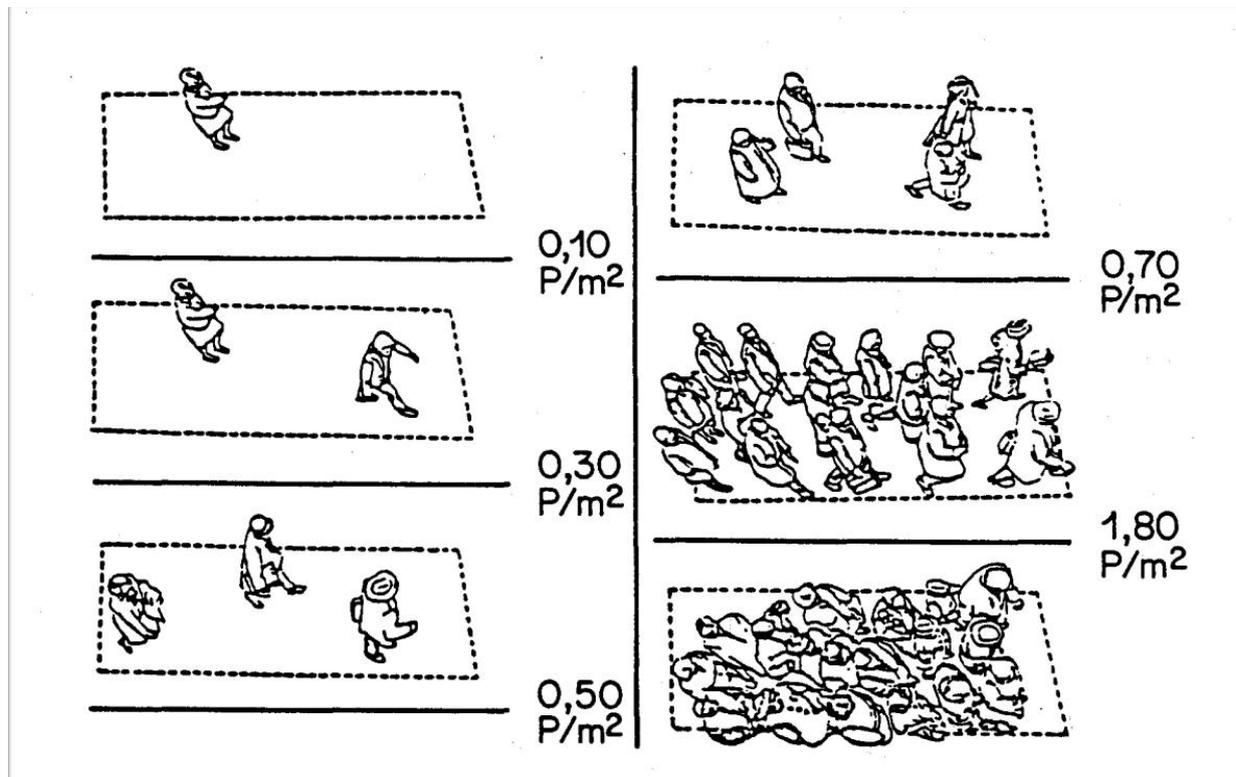


Abbildung 2 Darstellung verschiedener Fußgängerdichten

Quelle: Transportation Research Board, 1985

Umso höher die Dichte (P/m^2), desto höher der Personenfluss. Dabei kann eine zu hohe Dichte zu einer Überlastung der Fußgängeranlage führen, während der Fußgängerfluss abreißt. Der Fußgängerfluss ist auch von der Geschwindigkeit (m/s) der Personen abhängig. „Die Abhängigkeit zwischen der Fussgängerdichte und der Geschwindigkeit geht in die Berechnung der Leistungsfähigkeit von Fussgängeranlagen ein und ist besonders dort zu beachten, wo der Zeitbedarf eines Fussgängers von Belang ist.“ (Weidmann, 1993, 89). Zusammenfassend gilt, umso höher die Dichte, die Geschwindigkeit und die Breite des zu messenden Bereichs, desto höher der Fußgängerfluss. Allerdings sinkt die Geschwindigkeit bei steigender Dichte, da sich die Fußgänger bei höheren Dichten gegenseitig beim Bewegen behindern. „Der entscheidende Geschwindigkeitsabfall in der Ebene erfolgt zwischen etwa 0.5 und 2.0 P/m^2 . Bei 1.5 P/m^2 sinkt die Fußgängergeschwindigkeit bereits auf die Hälfte. Ueber etwa 5.4 P/m^2 kommt jegliche Fussgängerbewegung zum Erliegen“ (Weidmann, 1993, 88)

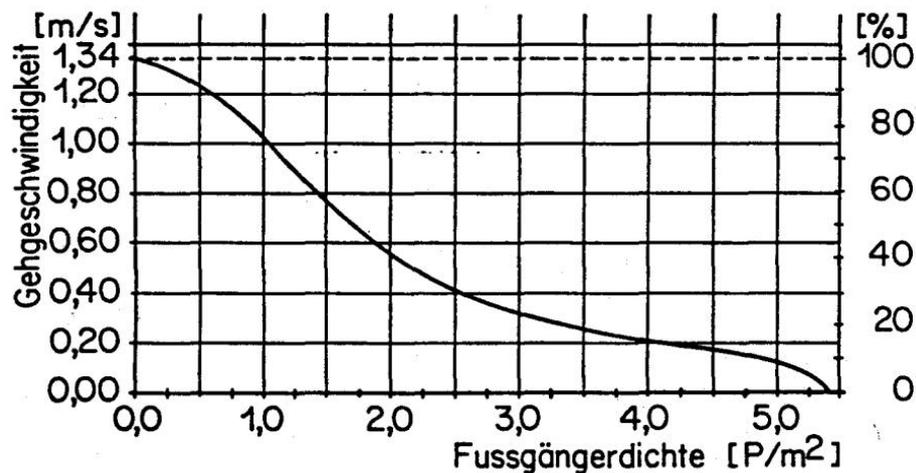


Abbildung 3 Abhängigkeit der Fussgängergeschwindigkeit von der Fussgängerdichte in der Ebene

Quelle: Weidmann, 1993, 5)

Mit der Dichte steigt der Fluss bis zu einem bestimmten Grad, der als Leistungsgrenze bezeichnet wird. Mit Simulationen wird versucht diese Leistungsgrenzen der Anlagenkapazität zu bestimmen. In der rechten Grafik in Abbildung 4 werden die Fußgängerflüsse zur Dichte in unterschiedlichen Simulationsmodellen dargestellt. Bei allen Modellen steigt der Fußgängerfluss mit der Dichte bis zu einem Hochpunkt und fällt dann wieder.

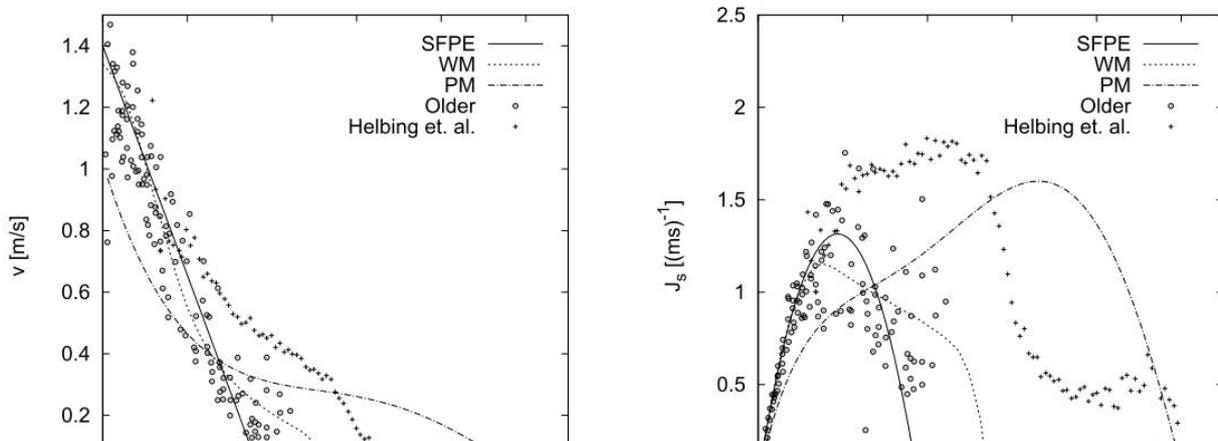


Abbildung 4 Fundamental Diagramme für Fußgängerbewegungen

Quelle: Schadschneider et. al., 2008, 10

3 Postvisualisierungstool

3.1 Desktop-Tool

Mit einem Postvisualisierungstool werden Daten von Fußgängersimulation veranschaulicht. Dazu werden Text-Dateien ausgewertet, die zu jedem Zeitschritt die Personenkennummer (IDs) der Individuen, deren Koordinaten und Personendichte beinhalten (s. Abb. 5). Die diskreten Zeitschritte werden dann nacheinander abgearbeitet, während bei jedem Schritt alle Positionen und Dichtewerte der bereits generierten Personen synchron aktualisiert werden. Dadurch wird eine dynamische Visualisierung erzeugt.

Time	ID	x	y	z	density
1.04	0	40.23	40.05	0.00	0 0.00
1.04	1	40.69	37.64	0.00	0 0.00
1.04	2	40.92	36.44	0.00	0 0.00

Abbildung 5 Textdatei mit Fußgängerdateien

Zusätzlich können Postvisualisierungstools viele Funktionen besitzen, wie zum Beispiel eine Farbliche Darstellung der Personendichte (s. Abb. 6), die Identifizierung der Fußgängerströme (s. Abb. 7) oder eine Parameterdarstellung durch Diagramme (s. Abb. 8). Diese Funktionen ermöglichen eine Vielzahl an Analysemöglichkeiten. Mit den gewonnen Informationen können Fußgängeranlagen auf ihre Leistungskapazitäten überprüft, kritische Bereiche aufgedeckt und Einflüsse von Parametern untersucht werden.



Abbildung 6 Farbliche Darstellung der Personendichte

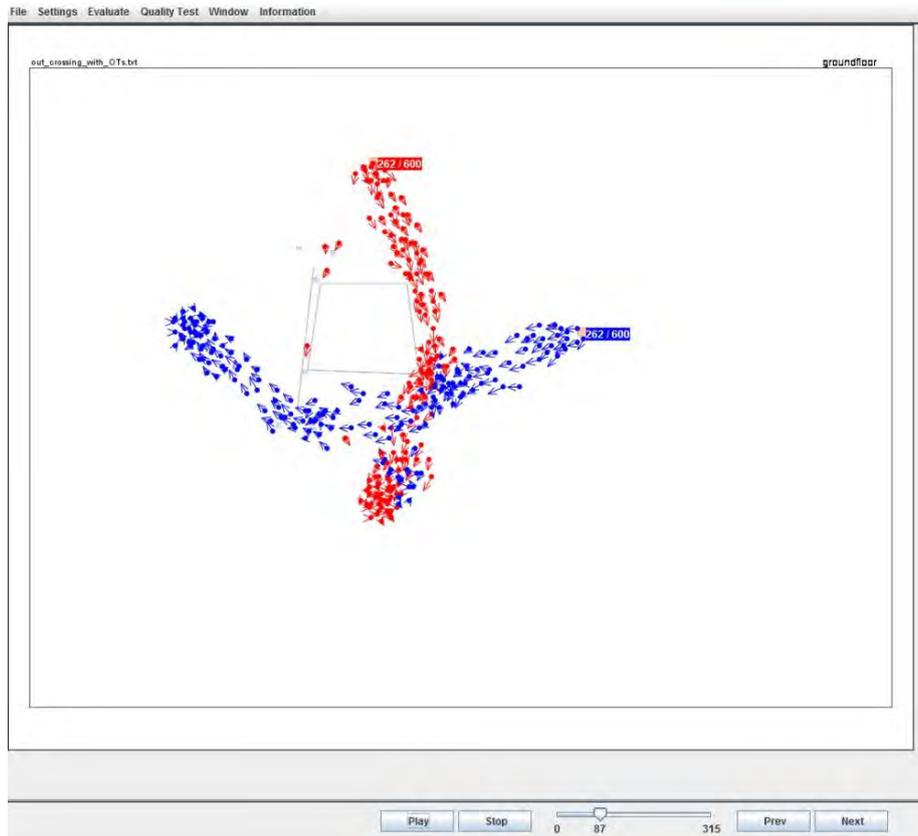


Abbildung 7 Abbildung des Fußgängerstroms

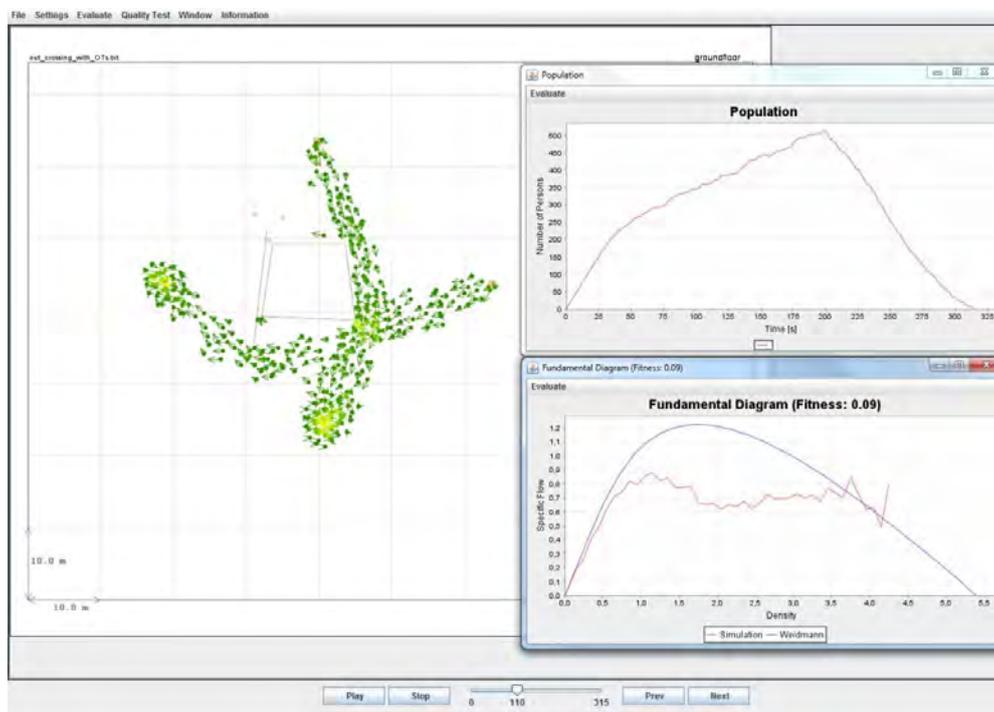


Abbildung 8 Darstellung von Diagrammen

3.2 Web-basiert

Eine web-basierte Implementierung ermöglicht dem Benutzer ohne selber das Programm und die Daten zu besitzen, *SumoViz* zu benutzen. Für die Anwendung werden lediglich ein Internetanschluss und ein Browser für Zugriff und Darstellung benötigt. Somit können Nutzer, selbst ohne das Tool zu besitzen, *SumoViz* mit fast jedem Rechner und von jedem Ort aus benutzen. Aufgabe war es im Zuge dieser Abschlussarbeit die noch sehr eingeschränkte Funktionalität zur Ausgangssituation zu erweitern.

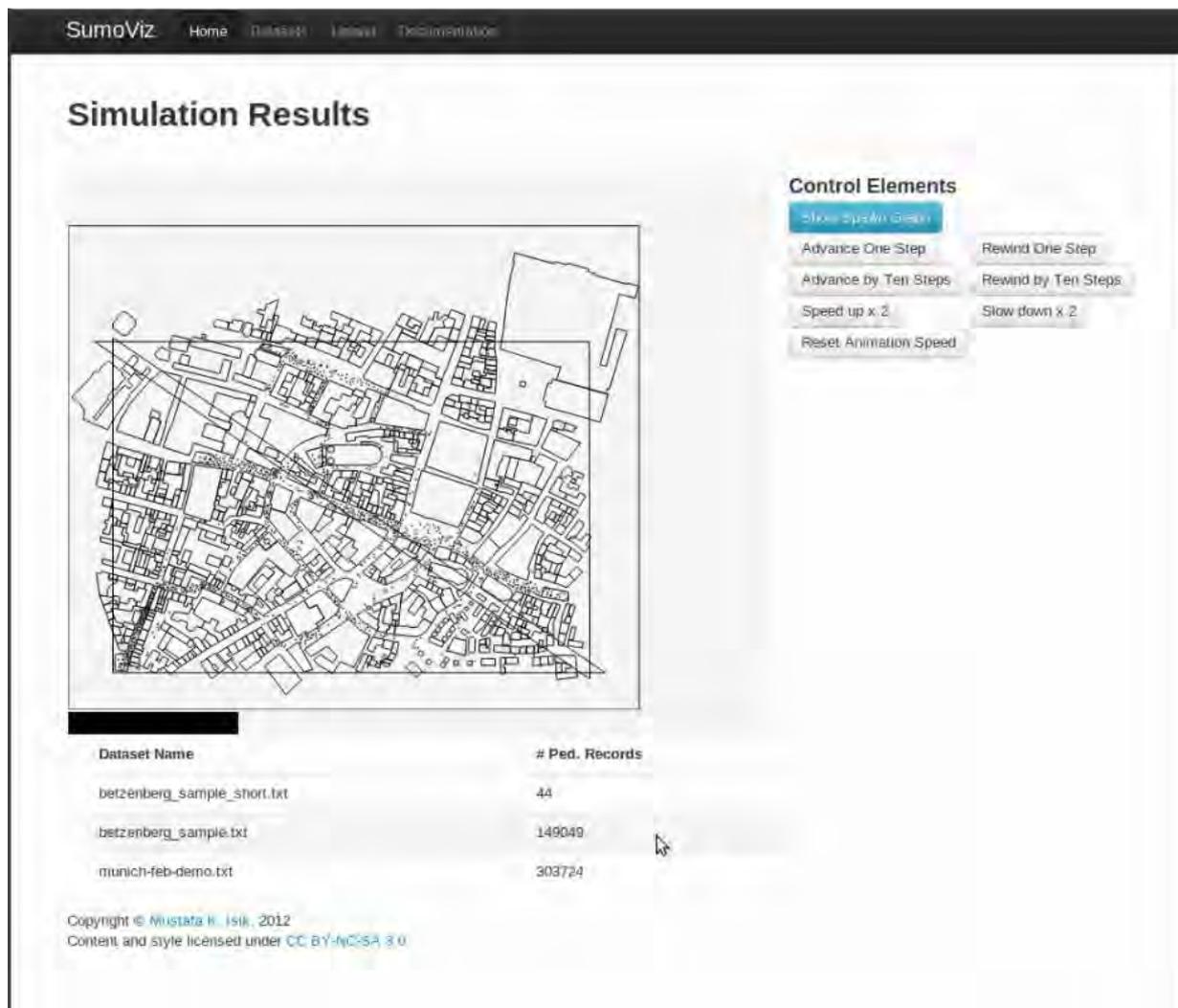


Abbildung 9 SumoViz ohne evaluations Funktionen

Quelle: Isik, 2012, 6

Vorherige Architektur

SumoViz ist eine Client/Server-Applikation mit einer Webbrowser basierten Präsentation. Dabei werden auf der Clientseite das HTML5 Canvas-Element und auf der Serverseite node.js und CouchDB verwendet. Der Vorteil bei der Verwendung dieser Technologien ist, dass die Kommunikation zwischen den Komponenten und der auszuführende Quellcode in JavaScript geschrieben sind. Das verringert die Komplexität und erleichtert die Entwicklung und die Kommunikation über die Schnittstellen enorm. CouchDB ist eine dokumentenorien-

tierte und schemalose Datenbank. Im Gegensatz zu SQL-Datenbanken, die zur Speicherung von stark strukturierten zusammenhängenden Daten design sind, wurde CouchDB für große Mengen von semi-strukturierten dokumentorientierten Daten entwickelt. Mit CouchDB ist kein Schema erforderlich, das bedeutet, dass neue Dokumententypen sicher neben alte hinzugefügt werden kann.(RandallLeeds, 2012) Die benötigten Daten werden mit MapReduce-Funktionen, die in JavaScript geschrieben sind, aufbereitet und weiterversendet.

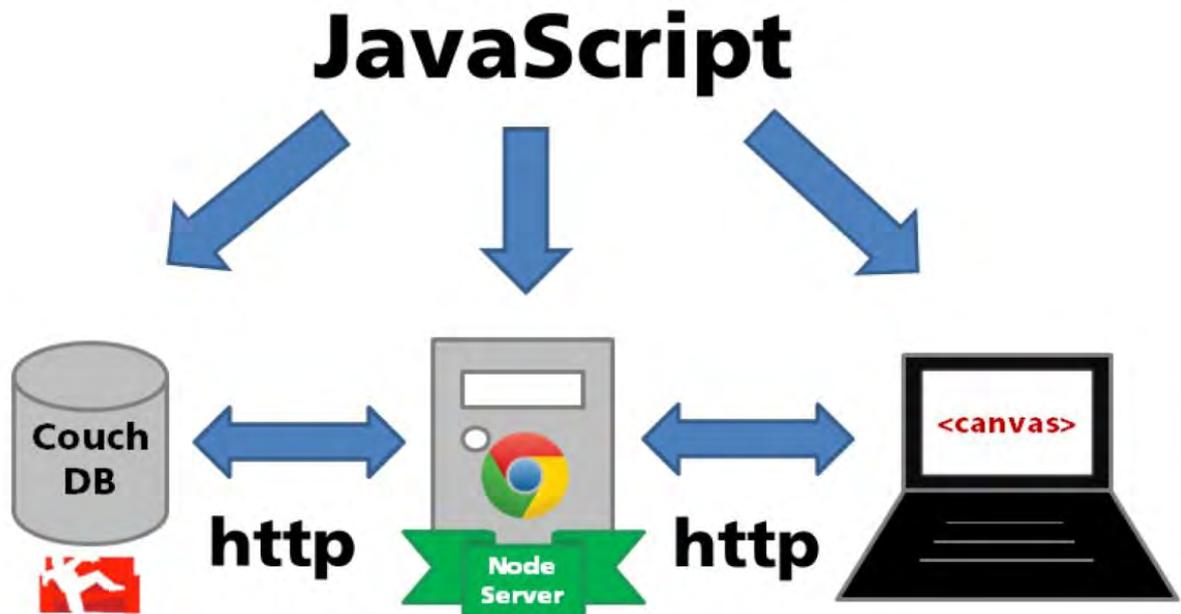


Abbildung 10 Architektur von SumoViz
Quelle: Isik, 2012, 3

Für die Visualisierung werden zuerst Simulationsdateien mittels node.js Server eingelesen und in der CouchDB Datenbank abgespeichert. Die aufbereiteten Daten werden mit dem http-Protokoll an den Node.js Server übertragen, der als serverbasierten JavaScript Interpreter fungiert. Das Node.js Ausführungsmodell ist um eine *Single-Threaded, run-to-completion* und *non-blocking* Event-Schleife aufgebaut, die es sehr ressourcenfreundlich skalierbar macht. (Isik, 2012, 6). Denn Thread basierte Netzwerke sind relativ ineffizient und sehr schwierig zu benutzen (nodejs.org). Beim *Single-Threading* wird nur der Prozess eines Befehls gleichzeitig ausgeführt anstatt wie beim Multithreading mehrere (Menéndez R., Lowe D. 2001, 512). *Deadlocks* werden auf zwei Arten vermieden. Prozesse die mit node.js gestartet wurden, können nicht mehr unterbrochen werden. Das wird als *run-to-completion* bezeichnet und vermeidet *Deadlocks*. Desweiteren werden beim *non-blocking* keine *Locks* verwendet die Prozesse nicht *verklemmt* (nodejs.org). Der Node.js Server, der vorher die Anfrage des Clients verarbeitet und die Datenbank weiter geschickt hat, schickt nun die angeforderten Daten von der Datenbank mittels http weiter an den Client. Der Client visualisiert mit Hilfe von Canvas die angeforderten Simulationsdateien.

Aktuelle Architektur

Für die Implementierung der Evaluationsfunktion wurde die Architektur von SumoViz verändert. Der node.js Server wurde ausgebonden, um die Komplexität zu reduzieren. Deshalb kann der Client mittels JSONP eine direkte Anfrage an die Datenbank schicken und somit Daten auch direkt von der Datenbank erhalten. JSONP (JavaScript Object Notation Pattern) ist eine Möglichkeit Daten im JSON-Format über mehrere Domänen anzufragen. Das ist nötig, da bestimmte Arten der Datenübertragung in der Browser Schicht, zum Schutz der Benutzer vor unerwünschten oder unsicheren und böswärtigen JavaScript Verhalten, beschränkt sind. Deshalb können eigentlich nur Daten übertragen werden, wenn die Ziel-Ressource-Domäne mit der Client-Domäne identisch sind. (Simpson) Die angefragten Daten werden dann im JSON-Format von der Datenbank gesendet. JSON (JavaScript Object Notation) ist ein Datenaustauschformat, das sich einer Teilmenge der JavaScript-Syntax ähnelt, obwohl es sich nicht um eine strikte Untergruppe handelt. Die Verwendung von JSON ist für jede Art von JavaScript basierten Anwendungen, einschließlich Websites und Browser-Erweiterungen sinnvoll. (Mortchek et al. 2012)

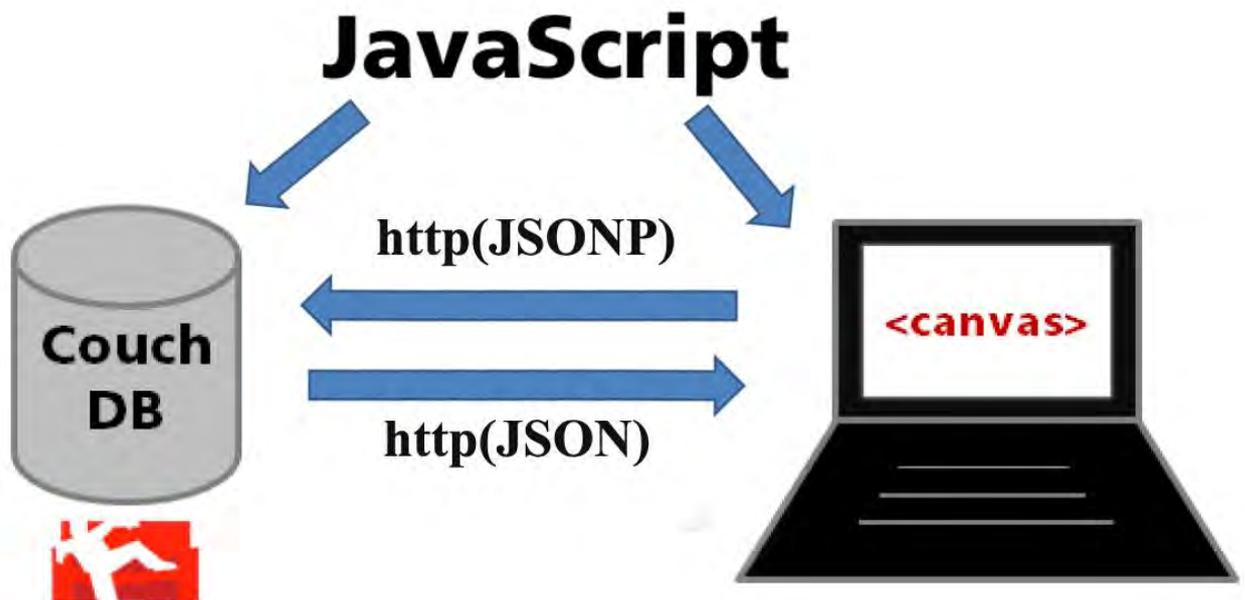


Abbildung 11 Eigene Architektur von SumoViz

Zur bisherigen Visualisierung wurden nur die Personen- und Geometrie-Koordinaten benötigt. Für die Evaluationsfunktionen werden aber weitere Daten benötigt. Die MapReduce-Funktion wird dazu so umgeschrieben, dass zu den Koordinaten auch die Personen IDs ausgegeben werden um die Koordinaten den Personen zuordnen zu können (s. Abb. 12). Auch müssen die Geometrie-Koordinaten den Objekten zuzuordnen sein, um Quellen und Ziel-Gebiete zu unterscheiden (s. Abb. 13). Zusätzlich werden Daten über die Personendichte benötigt.

Time	ID	x	y	z	density		
1.04	0	40.23	40.05	0.00	0	0.00	1.04
1.04	1	40.69	37.64	0.00	0	0.00	
1.04	2	40.92	36.44	0.00	0	0.00	
1.04	3	40.92	38.85	0.00	0	0.00	2.08
1.04	4	40.69	35.24	0.00	0	0.00	
1.04	5	40.46	42.05	0.00	0	0.00	
2.08	0	38.61	39.65	0.00	0	0.00	2.08
2.08	1	39.77	37.64	0.00	0	0.06	
2.08	2	40.00	36.44	0.00	0	0.06	
2.08	3	40.23	38.44	0.00	0	0.06	3.01
2.08	4	39.31	35.24	0.00	0	0.00	
2.08	5	39.77	41.65	0.00	0	0.00	
2.08	6	40.46	34.84	0.00	0	0.09	
2.08	7	40.46	39.65	0.00	0	0.00	

Abbildung 12 eingelesene Fußgängerdateien in Datenbank aufbereitet

Key	Value
"entire_geometry"	[[[0, 0], [41.76, 0], [41.76, 54.04], [0, 54.04]], ["entire_geometry"]]
"o1"	[[[23.13, 40.36], [25.28, 40.67], [25.34, 40.21], [23.13, 39.85], [23.13, 40.33]], ["o1"]]
"o2"	[[[18.96, 42.27], [19.95, 42.09], [19.92, 41.58], [18.93, 41.67]], ["o2"]]
"o3"	[[[33.82, 41], [34.85, 41], [34.85, 41.64], [33.82, 41.64]], ["o3"]]
"o4"	[[[12.24, 25.65], [12.72, 25.42], [13.13, 26.43], [12.65, 26.65], [12.22, 25.69]], ["o4"]]
"o5"	[[[11.45, 21.85],

Abbildung 13 eingelesene Geometrydateien in Datenbank aufbereitet

Die Daten werden in Arrays an den Client gesendet. Das Array mit den Fußgängerdaten ist so aufgebaut, dass jedes Feld einen Zeitschritt repräsentiert, das alle Objekte in diesem Zeitschritt beinhaltet. Bei dem Array mit den Geometriedaten, ist jedes Geometrieobjekt einzeln in einem Feld gespeichert(s. Abb. 14).

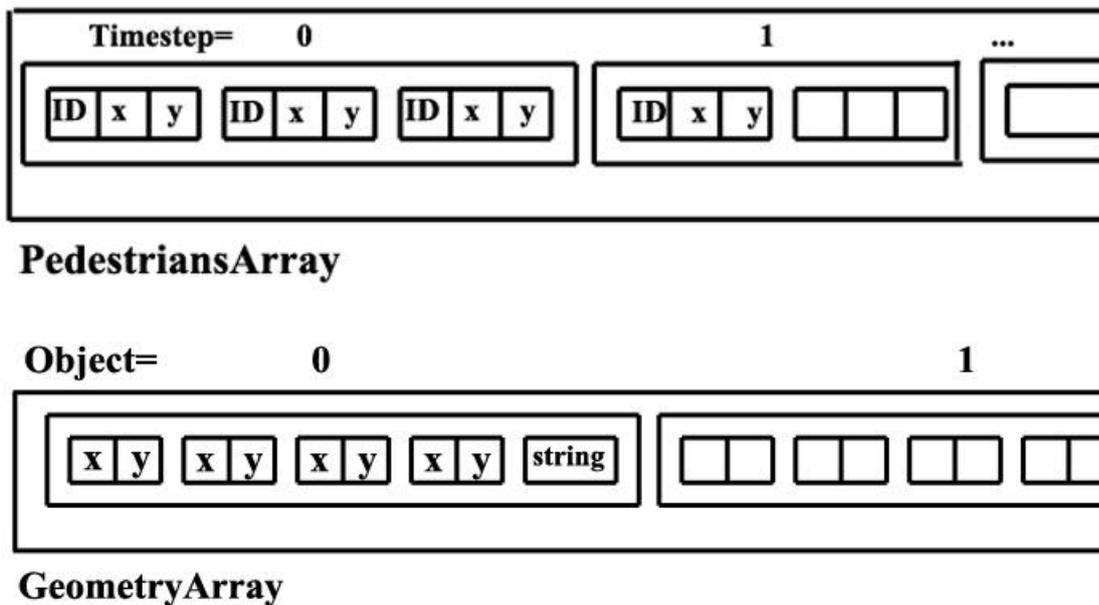


Abbildung 14 Struktur der Datenarrays, wie sie an den Client gesendet werden.

Eine weitere Änderung ist, dass zum Animieren der einzelnen Bilder, die vorherige implementierte recht komplexe Funktion, die diese Aufgabe übernommen hat, ersetzt wurde. Denn für die Animation wurde vorher eigens eine Funktion selber implementiert. Subjektiv gesehen, verringert die Verwendung einer bereits entwickelten, in JavaScript zur Verfügung stehenden, Funktion die Komplexität. Diese Funktion wird mit *window.requestAnimationFrame* aufgerufen. Da diese Funktion eine sehr neue Entwicklung ist, ist sie Browserspezifisch. Das bedeutet, dass der Browser zuerst identifiziert werden und dann je nach Browsertyp die dazugehörige Funktion aufgerufen werden muss. Zum Beispiel wird für den Firefox Browser die *window.mozRequestAnimationFrame(callback)* Funktion aufgerufen, während für den Chrome Browser *window.webkitRequestAnimationFrame(callback)* ausgeführt wird. Diese Methode teilt dem Browser mit, dass eine Animation ausgeführt werden soll. Mit dieser Anfrage wird mit dem Browser ein Zeitplan erstellt, in dem festgelegt wird wann ein neues Bild gezeichnet wird. (Zyunfei et al, 2012) Der Vorteil dieser Methode ist, dass nicht nur in bestimmten Zeitintervallen ein neues Bild gezeichnet wird, sondern die Methode prüft auch, ob der Browser bereit ist diese Aktion auszuführen. Bei einer Implementierung, die nur ein festes Zeitintervall für das Zeichnen eines neuen Bildes definiert, kann es zu Komplikationen kommen, wenn der Browser zu diesen Zeitpunkten gerade mit anderen Aufgaben beschäftigt ist.

4 Evaluationsfunktionalität

4.1 Datenaufbereitung für die Evaluations-Funktionen

Zur Visualisierung in *SumoViz* wurden bisher nur die Fußgänger- und Geometriekoordinaten benötigt. Für die Implementierung von Evaluationsfunktionen werden weitere Daten gebraucht. Diese Daten, wie etwa die Personendichte, werden teils von der Datenbank geliefert, während andere Daten, wie die Geschwindigkeit, erst ausgerechnet werden müssen. Alle Datensätze werden jeweils so in Arrays gespeichert, dass nur der Array-Index benötigt wird, um alle Informationen einer Person auszulesen. Liegt zum Beispiel eine Person P mit ihren Koordinatendaten im Array mit Index i, kann mit i auch auf die Dichte und Geschwindigkeit von P zugegriffen werden.

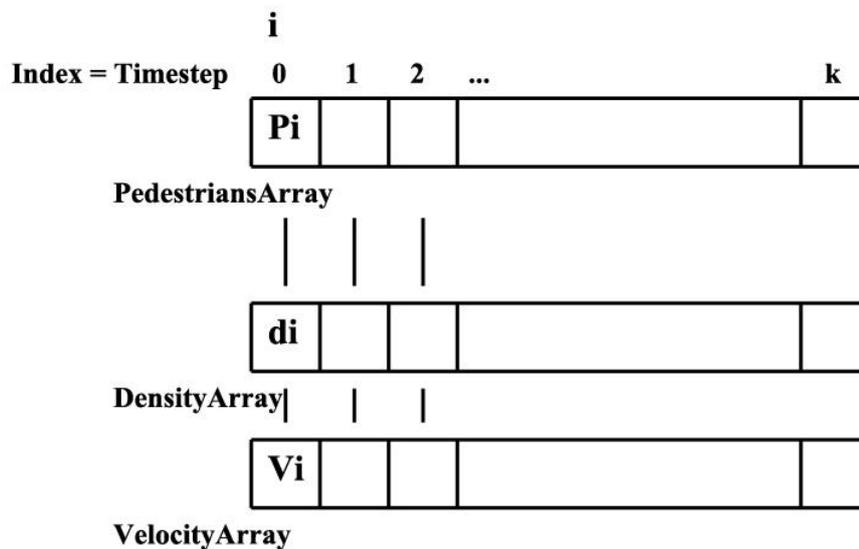


Abbildung 15 Zugriff auf Daten über Index

Bevor auf die Geschwindigkeit zugegriffen werden kann, muss diese zuerst berechnet werden. Hierzu wird die Strecke vom aktuellen Koordinatenpunkt zum Koordinatenpunkt des vorherigen Zeitschritts berechnet. Die Länge der Strecke S wird aus der Differenz der x - und y -Koordinaten und mit Hilfe des *Satz des Pythagoras* ermittelt:

$$S = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}$$

Mit der ermittelten Strecke und der Zeitschrittlänge kann die Geschwindigkeit berechnet werden.

4.2 Farbliche Darstellung der Personendichte

Ein gestufter Farbverlauf von Grün über Gelb nach Rot gibt die Höhe des Dichtewertes an. Somit werden kritische Bereiche in der betrachteten Simulation sichtbar. Für die Problemstellung ist dies von großer Bedeutung, da dort mögliche Engpässe bestehen, die eine zu hohe Dichte an Personen aufweisen.

Da die Datenbank bereits vorhandene Werte der Personendichte liefert, müssen diese nur noch, auf dem Wert basierend, eine Farbe zugeordnet bekommen. Dabei ist die Personendichte folgendermaßen definiert. Die Anzahl der belegten Zellen je Fußgänger wird mit der Anzahl der Nachbarzellen des ersten und zweiten Nachbarschaftsrings im Hexagonalgitter dividiert. Die Personendichte wird in einem 0.1er Schritten in 10 Stufen bis 1.0 in 10 Farben aufgeteilt. Alle Werte, die über 1.0 liegen werden rot markiert (s. Abb. 16).

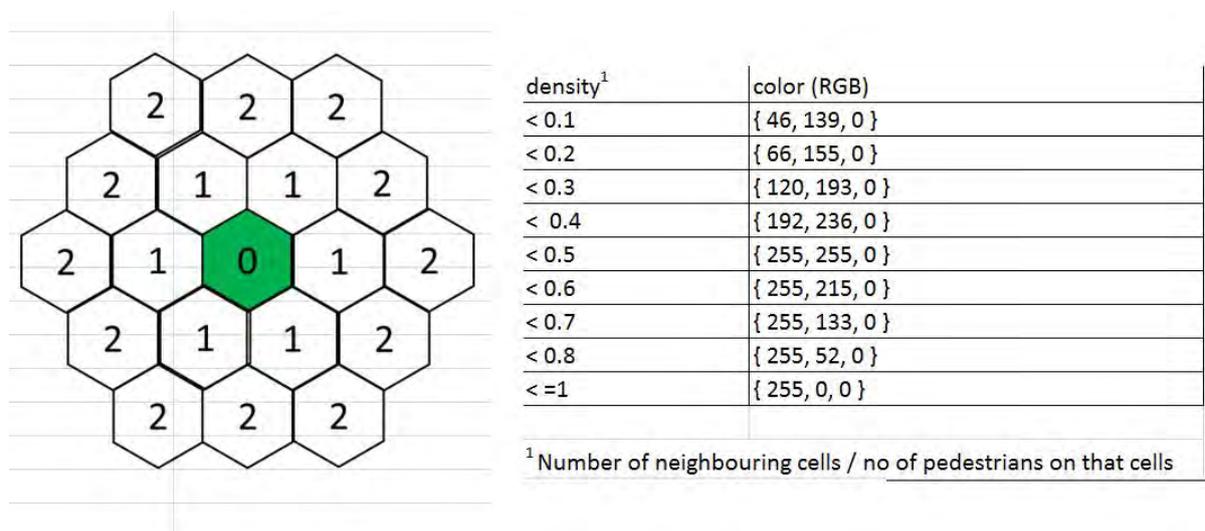


Abbildung 16 Dichte-Farbstufen

4.3 Search Person

Die Search Person Funktion ermöglicht dem Nutzer, eine bereits generierte Person zu identifizieren.

Der Nutzer gibt über Eingabeaufforderung die gesuchte Personen-ID ein. Es wird geprüft, ob die eingegebene ID im aktuellen Zeitschritt im Array vorhanden ist. Wird die ID nicht gefunden, so wird im Textfenster eine Fehlermeldung ausgegeben.

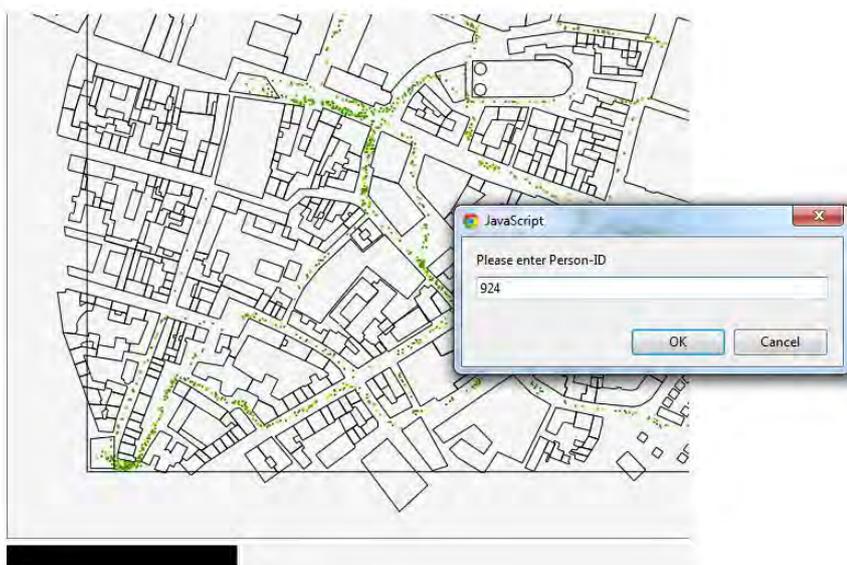


Abbildung 17 Eingabeaufforderung für Search Person Funktion

Wurde die ID gefunden, werden anhand des ID-Indexes die zugehörigen Werte wie Koordinaten, Dichte und Geschwindigkeit ausgelesen und im Textfeld angezeigt. Zusätzlich werden in allen vorherigen Zeitschritten die Koordinaten, die sich auf die ID beziehen, ausgelesen und miteinander verbunden, um einen Weg zu zeichnen, den die gesuchte Person zurückgelegt hat.



Abbildung 18 zurück gelegter Weg

4.4 Identify Person

Mit dieser Funktion wird es dem Nutzer ermöglicht, im aktuellen Zeitschritt eine Person durch Anklicken zu identifizieren. Das heißt, der Nutzer bekommt neben Index, Koordinaten, Geschwindigkeit und Dichte auch den bereits zurückgelegten Weg angezeigt. Anhand der Visualisierung des zurückgelegten Weges können beeinflussende Parameter nachgewiesen werden, die auf den Weg des Individuums zum Ziel einwirken. Dabei wird sichtbar wie weit die Einflussparameter sich auf die Person auswirken und sie vom optimalen Ziel-Weg abdrängen.

Die Implementierung ist zum größten Teil analog zur Search Person-Funktion. Der Unterschied besteht darin, dass die Personen nicht mittels Eingabeaufforderung ausgewählt werden, sondern über einen Mausklick auf die Person. Um das zu ermöglichen, muss eine Funktion geschrieben werden, welche die Mauskoordinaten auf das Canvas-Fenster umrechnet und ausgibt. Die Schwierigkeit besteht darin, dass die Position des Canvas-Fensters dynamisch ist. Auch muss für das Umrechnen der Koordinaten beachtet werden, dass das Canvas-Koordinatensystem anders ausgerichtet ist, als das Koordinatensystem, auf welches sich die Personen- und Geometriekoordinaten beziehen.

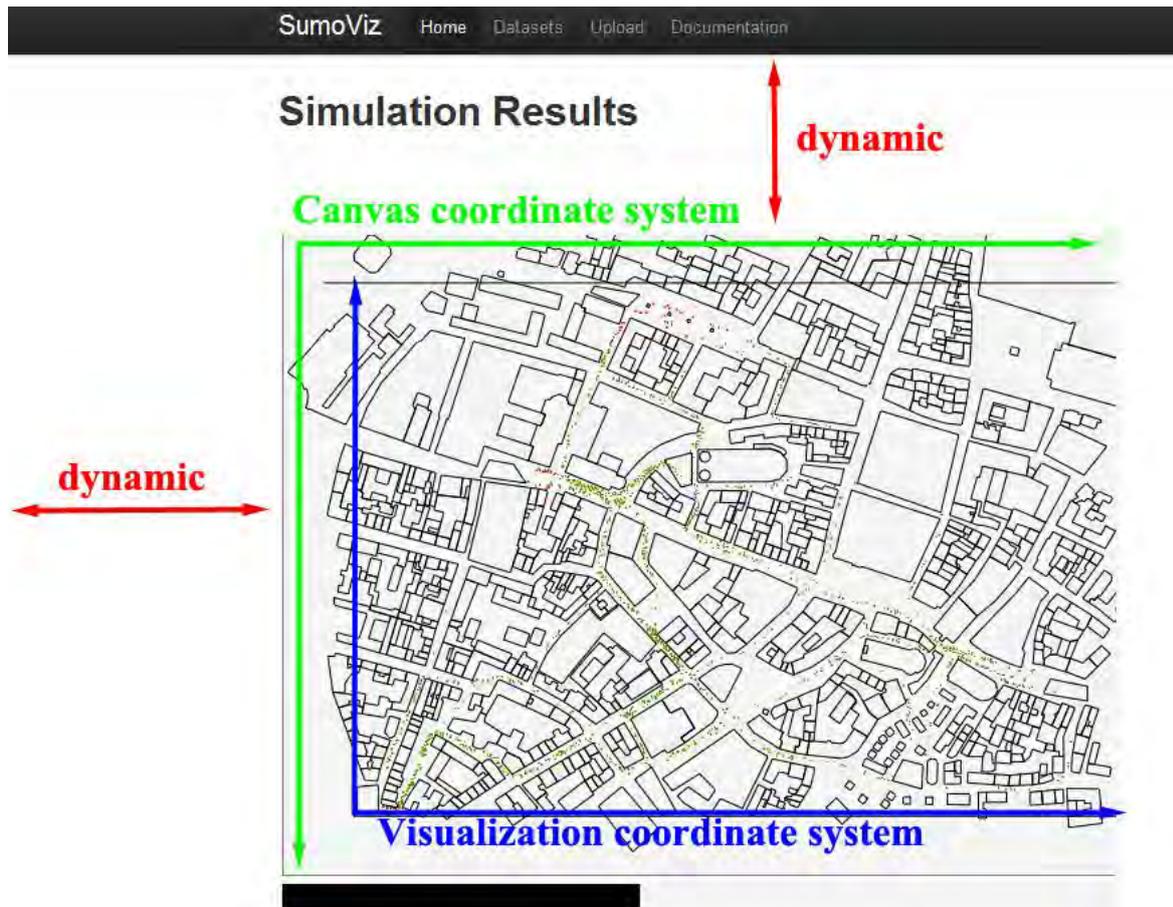


Abbildung 19 Koordinatensysteme

Mit einem Mausklick definiert der Benutzer die zu übergebenden Koordinaten. Das Array mit dem aktuellen Zeitschritt wird durchlaufen, während geprüft wird, ob eine der darin enthaltenen Koordinaten mit dem vom Nutzer definierten Koordinaten übereinstimmen. Dabei ist zu beachten, dass eine Toleranzabweichung implementiert werden muss. Denn es würde sich für den Benutzer als schwierig gestalten, genau auf einen Pixel bzw. auf eine Koordinate zu klicken. Mit der Toleranzabweichung sind die Positionen der Personen nicht nur mit einer Koordinate definiert, sondern bekommen einen Koordinatenbereich zugewiesen. Der Koordinatenbereich ist um den Faktor 0,2 größer als der Bereich, den jede Person zuvor über ihre Koordinaten zugewiesen bekommen hat. Dieser Vergrößerungsfaktor wurde subjektiv als angenehm für die Benutzung dieser Funktion, anhand von Tests, eingeschätzt.

4.5 Evaluate Rectangle

Mit dieser Funktion kann der Nutzer einen Bereich in Form eines Rechtecks bestimmen, in dem Information, wie Anzahl der Personen, Durchschnittsdichte, Größe des Bereichs und die mittlere Geschwindigkeit, angezeigt wird. Der Nutzer kann dadurch selbst definierte Bereiche genauer analysieren und anhand der zusätzlichen Informationen die Fußgängersimulation besser auswerten.

Zuerst wurde eine Funktion implementiert, die es dem Benutzer möglich macht ein Rechteck zu ziehen, in dem die Daten ausgewertet werden sollen. Dabei wird mit einem Mausklick ein Startpunkt und mit dem Loslassen der Maustaste der diagonale Endpunkt des Rechtecks fest-

gelegt. Während der Nutzer die Maustaste gedrückt hält, kann mit der Maus die Größe des Rechtecks bestimmt werden.



Abbildung 20 Rechteck ziehen

Wurde das Rechteck gezeichnet, werden alle Koordinaten im aktuellen Zeitschritt darauf überprüft, ob sie sich im angegebenen Gebiet befinden. Befinden sich die Personen-Koordinaten im markierten Bereich, werden mittels Index, Geschwindigkeit und Dichte ausgelesen. Geschwindigkeit und Dichte aller Personen im Gebiet werden zur Gesamtgeschwindigkeit und -dichte zusammengefasst, die durch Division mit der Personenanzahl die Durchschnittsgeschwindigkeit und -dichte ergeben. Diese Werte werden dann mit dem Flächeninhalt im Textfeld unter dem Fortschrittsbalken ausgegeben.



Abbildung 21 Evaluate Rectangle Funktion

4.6 Evaluate-Rectangle-Diagram

Bei der Evaluate-Rectangle Funktion werden die Werte zum jeweiligen Zeitschritt angezeigt. Wird der Verlauf der Werte im markierten Bereich über den gesamten Visualisierungszeitraum benötigt, wird mit der Evaluate-Rectangle Diagram Funktion Diagramme generiert, die die Verläufe von Durchschnittsgeschwindigkeit und –dichte zur Zeit anzeigen. Zu diesen zwei Diagrammen wird zusätzlich ein drittes gezeichnet, das den Fußgängerfluss zur Dichte darstellt. Zur Berechnung des Fußgängerflusses, wird die Anzahl der Personen durch die Größe der Fläche dividiert.



Abbildung 22 Evaluate Rectangle Diagram Funktion

Dazu werden die berechneten Funktionen auf jeden Zeitschritt angewendet. Die jeweiligen Werte den Zeitschritten zugeordnet und in einem Array gespeichert. Danach wird ein neues Fenster geöffnet, in dem das Diagramm gezeichnet wird.

4.7 Evaluate Flow

Der Nutzer kann den Fußgängerfluss messen, indem er mit dieser Funktion eine selbst festgelegte Linie zeichnet, die als Schranke oder Schwelle zur Ermittlung des Fußgängerflusses dient. Der Fußgängerfluss gibt Auskunft über Geschwindigkeit und Dichte in einem definierten Gebiet in eine bestimmte Richtung an. Diese Informationen sind zur Einschätzung der Leistungskapazität von Fußgängeranlagen nötig.

Laut [Schadschneider et al., 2008, 8] wird der Fußgängerfluss berechnet, indem die Durchschnittsgeschwindigkeit und -dichte aller Personen mit der Länge einer vom Nutzer definierten Schwelle, die die Personen überqueren, multipliziert wird.

$$J = p v b = J_s b$$

$J = \text{flow}$
 $p = \text{density}$
 $v = \text{velocity}$
 $b = \text{width}$

Der *spezifische Fluss* ist als Fußgängerfluss pro Längeneinheit definiert:

$$J_s = p v.$$

Zu beachten ist, dass es neben dieser Methode noch andere Methoden zur Berechnung des Fußgängerflusses existieren. Diese können aber sehr zeitaufwendig und komplex sein. (Schadschneider et al., 2008, 8)

Bei dieser Abschlussarbeit wird aber die im RiMEA-Projekt genannte Formel zur Berechnung des *spezifischen Fußgängerflusses* verwendet:

„Der spezifische Fluss ist die Anzahl Personen, die einen bestimmten Querschnitt pro Meter lichter Breite und pro Sekunde passieren. Die Einheit ist *Personen/ms*. Der spezifische Fluss ist im Wesentlichen abhängig von der Personendichte (Einheit: *Personen/m²*) und lässt sich ähnlich der folgenden Formel berechnen“

$$\Phi_{s,max} = p \cdot 1,34 \cdot \left(1 - e^{-1,913 \left(\frac{1}{p} - \frac{1}{5,4} \right)} \right)$$

(Seyfried, A. et al., 2008, S. 60)

Bei der Implementierung werden zuerst alle Personen aussortiert, die erst im aktuellen Zeitschritt generiert, also im vorherigen noch nicht visualisiert, wurden. Dann werden die Koordinaten des aktuellen Zeitschritts mit denen im vorherigen Zeitschritt verglichen, um die Gehrichtung der Personen zu ermitteln. Der Nutzer gibt über Eingabeaufforderung an, welche Flussrichtung gemessen werden soll. Es kann zwischen den vier Himmelsrichtungen ausgewählt werden.

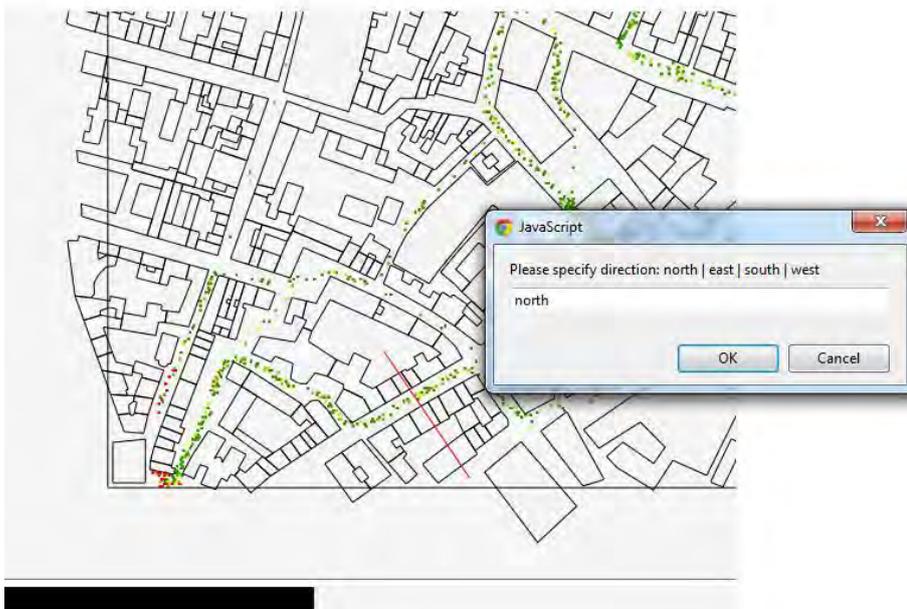


Abbildung 23 Eingabeaufforderung nach Flussrichtung

Alle Personen, deren Gehrichtung nicht der ausgewählten Himmelsrichtung entspricht, werden dann aussortiert. Nun muss geprüft werden, ob die Personen die markierte Schwelle überqueren. Dafür werden die Steigungen der definierten Schwelle und der Strecke zwischen den Koordinaten des aktuellen und vorherigen Zeitschritts einer Person ermittelt. Beide Strecken werden dann als Geraden gesehen, um ihre Linearen Funktionen zu berechnen. Haben beide Linien nicht die gleiche Steigung und sind somit nicht parallel, haben sie einen gemeinsamen Schnittpunkt. Nachdem der Schnittpunkt anhand Gleichstellung der linearen Funktionen berechnet wurde, wird zuletzt überprüft, ob der Schnittpunkt sich im Bereich beider Strecken befindet. Zuletzt wird der Fluss mit den Werten aller Personen berechnet, die die Schwelle überqueren und mit weiteren Werten im Textfenster ausgegeben.



```

Person(s) 2
average Density(p) 0.09
average Velocity(m/s) 1.02
b(m) 12.94
Flow(Jsb) 1.12
specific Flow(Js) 0.09

```

Abbildung 24 Angezeigte Werte bei Evaluate Flow

4.8 Evaluate Flow Diagram

Mit dieser Funktion kann der Verlauf des Fußgängerflusses ermittelt werden. Dazu wird der Fluss in allen Zeitschritten bis zum aktuellen Zeitschritt berechnet und in ein Array gespeichert. Mit den gespeicherten Werten lässt sich ein Zeit-Fluss Diagramm zeichnen

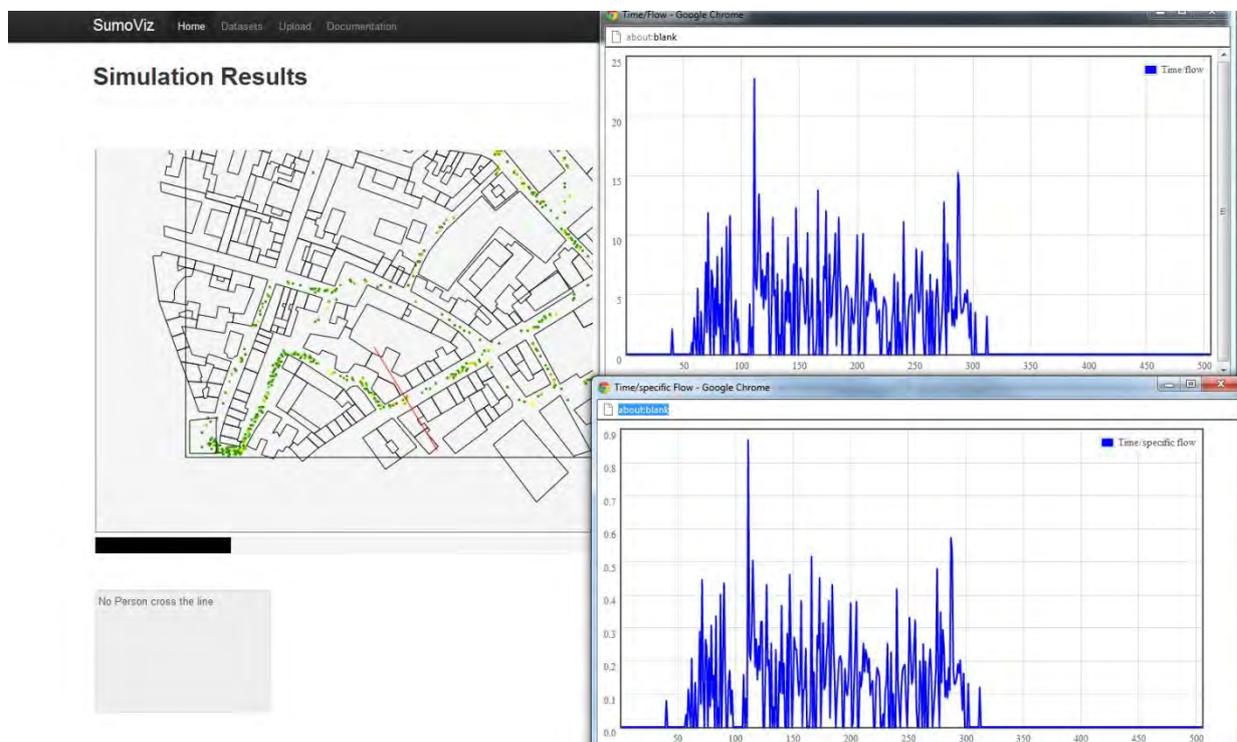


Abbildung 25 Flow Diagramme

4.9 Source / Target

Mit Source/Target werden zum aktuellen Zeitschritt angezeigt, wie viele Personen in welchen Startgebieten bereits visualisiert wurden und wie viele Personen in den Zielgebieten bereits angekommen sind. Der Nutzer kann somit sehen, ob die Fußgängermengen gleichmäßig auf die Startpunkte verteilt werden oder, zum Beispiel aus einer Startquelle überproportional viele Fußgänger generiert werden. Genauso kann bei den Zielgebieten unterschieden werden, ob bestimmte Zielgebiete übermäßig von Fußgängern angesteuert wurden.

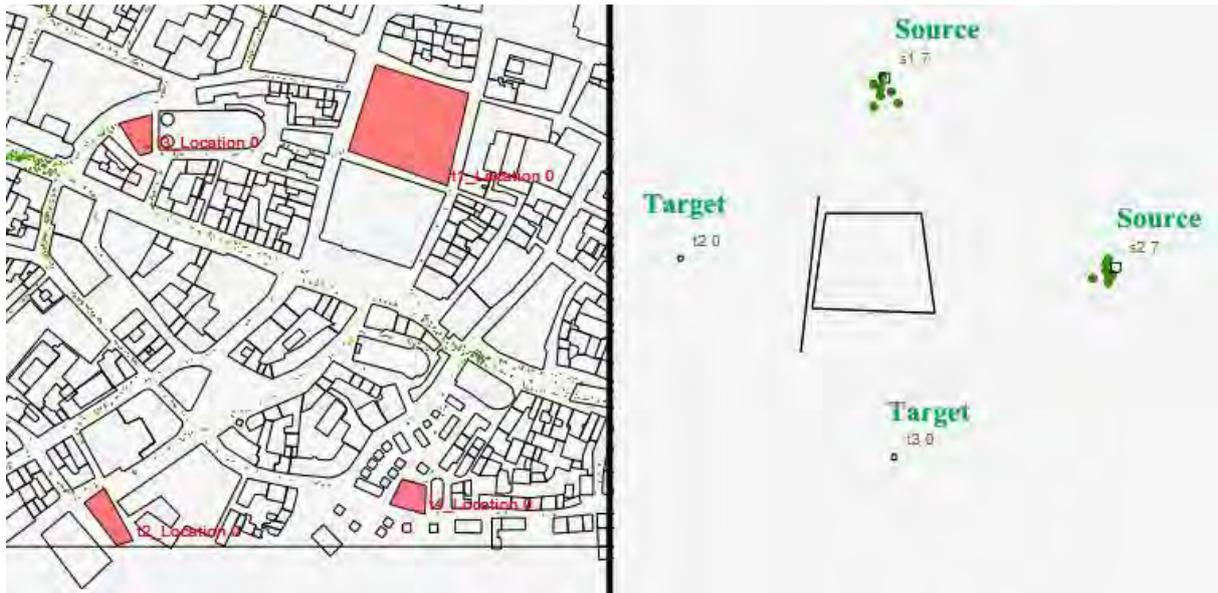


Abbildung 26 Source / Target Funktion

Zur Realisierung dieser Funktion müssen zuerst alle Startkoordinaten der Personen identifiziert, ausgelesen und gespeichert werden. Dazu wird für jede Person je ein Objekt angelegt, welches die ID als Schlüssel und die Startkoordinaten als Wert hat. Beim Anlegen wird geprüft, ob die ID bereits einen Wert zugeordnet bekommen hat. Damit wird abgefragt, ob die Person bereits vorher generiert wurde. Die Koordinaten, die ein Fußgänger beim ersten Auftauchen im Array hat, sind gleichzeitig die Startkoordinaten. Das Array wird bis zum aktuellen Zeitschritt als Index durchlaufen. Wurden alle Startkoordinaten gespeichert, müssen alle Startgebiete identifiziert werden. Dafür wird das Array mit den Geometriedaten durchlaufen. Über String-Einträge nach den Geometrieinträgen ist es möglich die verschiedenen Geometrieobjekte zu unterscheiden. Nun werden alle Startkoordinaten durchlaufen und jeweils geprüft, ob diese sich in einem der Gebiete befinden. Anschließend wird gezählt, welchen Gebieten wie viele Koordinaten zugewiesen wurden. Die Ermittlung der angekommenen Fußgänger in den Zielgebieten funktioniert auf die gleiche Weise. Anstatt die Koordinaten beim ersten Erscheinen einer ID zu speichern, werden nun die letzten Koordinaten gespeichert, die eine ID hat.

5 Zusatzfunktionen

5.1 Path Direction

Die Path Direction ist eine Funktion, die die Gehrichtung aller generierten Personen anzeigt. Zuerst werden alle Personen aussortiert, die im nächsten Zeitschritt nicht mehr vorkommen, also ihr Ziel erreicht haben. Dann werden alle Koordinaten des nächsten Zeitschritts ausgelesen und mit den aktuellen Koordinaten verbunden. Path Direction zeichnet also eine Linie von jeder Person zu ihrer nächsten Position.



Abbildung 27 Path Direction Funktion

5.2 Zoom Funktion

Die Layouts der Simulationsdaten können sich in der Größe, basierend auf das darzustellende Gebiet und Flächen, stark unterscheiden. Dafür wurde eine Zoom Funktion implementiert, die dem Benutzer ermöglicht die Visualisierung zu Vergrößern oder zu Verkleinern. Bei der Darstellung der Fußgänger und der Geometrie, werden deren Koordinaten mit einem Zoomfaktor verrechnet. Dieser Zoomfaktor kann mit den zu Verfügung stehenden Schaltflächen verändert werden. Zusätzlich kann der Nutzer den Zoomfaktor wieder auf den ursprünglichen Wert zurücksetzen um die Visualisierung auf die anfängliche Größe zu skalieren. Die Besonderheit bei dieser Funktion ist, dass sie während der Animation betätigt werden kann. Zusätzlich sind alle weiteren Funktionen auch im Zoom Modus noch anwendbar, da auch die Mauskoordinaten mit dem Zoomfaktor verrechnet werden.



Abbildung 28 Zoom Funktion

6 Ausblick

Obwohl die jetzigen Fußgängersimulationen schon wertvolle Informationen liefern, wird ein großer Teil der Mechanismen in der Fußgängerdynamik noch nicht vollends verstanden. Das Problem liegt in den menschlichen Parametern, die schwer zu verstehen, beziehungsweise vorherzusehen, sind. Doch werden die Fortschritte in der Psychologie und der Physik helfen, Modelansätze zu optimieren. Computer mit mehr Rechenleistung erlauben Simulationen mit komplexeren Strukturen. Ziel ist es, im Zuge des Fortschritts, realistischere Simulationen zu generieren, um das menschliche Verhalten genauer und korrekter vorherzusehen und somit Fußgängeranlagen effizienter und effektiver zu gestalten. (Schadschneider et al., 2012, 46)

Literaturverzeichnis

- Isik, M. K. (2012):** SumoViz – HTML5-based Visualization of Pedestrian Simulation Data. Documentation, München 2012.
- Mortchek et al. (2012):** JSON. In: <https://developer.mozilla.org/en-US/docs/JSON>.
zugegriffen am 16.09.2012
- Nodejs.org:** node.js - Node's goal is to provide an easy way to build scalable network programs. In: <http://nodejs.org/about/>, zugegriffen am: 24.09.2012
- RandallLeeds (2012):** Introduction. In: <http://wiki.apache.org/couchdb/Introduction>,
zugegriffen am 16.09.2012
- Menéndez R., Lowe D. (2001):** Murach's CICS for the COBOL Programmer. Mike Murach & Association, 01.06.2001
- Schadschneider, A. et al. (2008):** Evacuation Dynamics: Empirical Results, Modeling and Applications. Springer, Berlin 2008.
- Seyfried, A. et al. (2008):** Richtlinie für Mikroskopische Entfluchtungsanalysen. In: www.rimea.de, 2004-2007.
- Simpson K. :** JSON-P (JSONP). In: <http://www.json-p.org/>. zugegriffen am 16.09.2012
- Spiegel (2010):** Tödliches Love-Parade-Desaster - Die Menschenfalle. In: <http://www.spiegel.de/panorama/toedliches-love-parade-desaster-die-menschenfalle-a-708340.html>, zugegriffen am 11.09.2012.
- Transportation Research Board (1985):** Highway Capacity Manual. Special Report 209, Washington DC 1985.
- Weidmann, U. (1993):** Transporttechnik der Fussgänger - Transporttechnische Eigenschaften des Fussgängerverkehrs. Schriftenreihe des IVT Nr. 90, Zürich 1993.
- Zyunfei et al.(2012):** window.requestAnimationFrame. In: <https://developer.mozilla.org/en-US/docs/DOM/window.requestAnimationFrame>, zugegriffen am 16.09 2012