Hochschule für Technik und Wirtschaft Dresden; Fraunhofer-Institut für Verkehrs- und Infrastruktursysteme, Dresden

SIMULATION VON MENSCHENMENGEN IM URBANEN UMFELD MIT OPENSTREETMAP-DATEN

Axel Meinert, Stefan Hahmann, Patrick Brausewetter, Stephan Kopf

Zusammenfassung: Die Simulation von Fußgängern und Menschenmengen im urbanen Umfeld hat große Relevanz bei der Planung von Veranstaltungen, Verkehrs- und baulichen Anlagen und in der Entwicklung von Evakuierungsszenarien. In diesem Beitrag wird zunächst ein Überblick über geeignete physikalisch-mathematische Simulationsmodelle mit einem Fokus auf das Social-Force-Modell gegeben. Ein weiterer Schwerpunkt bildet die Darstellung der Fußgänger in Form von Hüllen innerhalb der Simulation. In einem Softwareüberblick werden nichtkommerzielle Programme zur Fußgängersimulation vergleichend gegenübergestellt. Anschließend wird analysiert, inwiefern öffentliche Geodaten, wie OpenStreetMap-Daten, für entsprechende Simulationen verwendet werden können und welche Objektarten für Fußgängersimulationen nutzbar sind. Darauf aufbauend wird die prototypische Umsetzung der neuen leichtgewichtigen Open-Source-Software jCrowdSimulator vorgestellt, welche Fußgänger und Menschenmengen im urbanen Umfeld simuliert und visualisiert. Abschließend werden die Ergebnisse der Performanztests diskutiert, welche eine lineare Komplexität bezüglich der Anzahl der simulierten Menschen bestätigen.

Schlüsselwörter: Fußgängersimulation, Social-Force-Modell, OpenStreetMap, Java

CROWD SIMULATION IN URBAN AREAS WITH OPENSTREETMAP DATA

Abstract: The simulation of pedestrians and crowds in urban areas is highly relevant for the planning of events, traffic, and buildings, as well as evacuation scenarios. This contribution starts with an overview on physical-mathematical simulation models, with a special focus on the Social Force Model. We also focus on the visualization of pedestrians within the crowd simulation using hulls. A review of non-commercial software in the field is presented. Afterwards, we discuss how Volunteered Geographic Information, in particular OpenStreetMap, can be used for such kind of simulations and which types of data objects are applicable for crowd simulations. Based on this, we present the prototypical implementation of the new lightweight Open-source software jCrowdSimulator, which simulates and visualises pedestrians and crowds in urban areas. Finally, we show performance tests, which prove the linear complexity of the prototype implementation and discuss open research issues.

Keywords: Crowd simulation, Social Force Model, OpenStreetMap, Java

Autoren

M. Eng. Axel Meinert
Hochschule für Technik und Wirtschaft Dresden
Friedrich-List-Platz 1
D-01069 Dresden
und
Fraunhofer-Institut für Verkehrs- und Infrastruktursysteme
Abteilung Strategie und Optimierung
Zeunerstraße 38
D-01069 Dresden
E: axel.meinert@ivi.fraunhofer.de

Dipl.-Inf. Patrick Brausewetter
Dr.-Ing. Stefan Hahmann
Fraunhofer-Institut für Verkehrs- und Infrastruktursysteme
Abteilung Strategie und Optimierung
Zeunerstraße 38
D-01069 Dresden
E: patrick.brausewetter@ivi.fraunhofer.de
stefan.hahmann@ivi.fraunhofer.de

Prof. Dr. rer. nat. Stephan Kopf
Hochschule für Technik und Wirtschaft Dresden
Friedrich-List-Platz 1
D-01069 Dresden
E: stephan.kopf@htw-dresden.de

1 MOTIVATION

Die Simulation von Fußgängern und Menschenmengen hat eine große Relevanz bei der Planung von Veranstaltungen, Verkehrs- und baulichen Anlagen sowie bei der Entwicklung von Evakuierungsszenarien. Sie sind kostensparender als reale Tests und weniger gefährlich (Gatto 1993, Kirchner et al. 2003). Lange Zeit lag der Fokus der entwickelten Anwendungen auf dem Gebiet der Simulation des motorisierten Verkehrs. Die stetig wachsende Bevölkerung, die stärkere Urbanisierung und damit sich verstärkende Verkehrs- und Umweltprobleme führten dazu, dass alternative Fortbewegungsarten an Bedeutung gewannen. Damit einher ging das Ziel, den Anteil des Fußgängerverkehrs zu erhöhen, wobei die Analyse von Fußgängerströmen helfen kann.

Unglücksfälle während Großveranstaltungen, wie bei der Love-Parade in Duisburg 2010 (Helbing & Mukerji 2012) oder mehrere Vorfälle während des Hadsch in Mekka (Johansson 2009) führten in jüngerer Vergangenheit dazu, dass die Planung von Fußgängerströmen für solche Veranstaltungen stärker in das Bewusstsein von Behörden, Veranstaltern und Wissenschaft rückte. Simulationen helfen dabei, bauliche Anlagen zu entwickeln, die dazu beitragen, Fußgängerströme zu optimieren. Ein Beispiel dafür zeigt Abbildung 1. Helbing et al. (2002) nennen eine unvollständige Liste von Unglücksfällen, die große mediale Aufmerksamkeit erlangten.

Die Analyse von Personenströmen in Gebäuden, z. B. während einer Entfluchtung, resultiert ebenfalls aus dem Ziel, Gefahrenstellen zu minimieren. In Deutschland enthalten die Musterbauordnung – MBO (ARGEBAU Fassung 2002, zuletzt geändert in 2012) und die Muster-Versammlungsstättenverordnung – MVStättV (ARGEBAU – Fachkommission Bauaufsicht Fassung 2005, zuletzt geändert 2014) bauliche Vorschriften sowie Festlegungen zu Flucht- und Rettungswegen. Strikte Vorgaben zur Verwendung von Evakuierungssimulationen existieren nicht, werden aber empfohlen (Meyer-König et al. 2007, Orth 2009). Insbesondere der RiMEA e. V. (Richtlinie für Mikroskopische Entfluchtungs-Analysen) verfolgt das Ziel, Simulationen zur Entfluchtung zu standardisieren. Im Sinne dieser Richtlinie sollen Simulationen in folgenden Anlagen durchgeführt werden:

- Versammlungsstätten mit Versammlungsräumen, die mehr als 200 Personen fassen;
- Versammlungsstätten im Freien, die mehr als 1.000 Besucher fassen; Sportstadien, die mehr als 5.000 Besucher fassen;
- Gebäude besonderer Art und Nutzung (= Sonderbauten) wie Krankenhäuser, Flughäfen oder Bahnhöfe (siehe Abbildung 2).

In diesem Beitrag wird die prototypische Entwicklung des ¡CrowdSimulator vorgestellt, der die Bewegung von Personen in einer Menschenmenge simuliert. Im Speziellen soll die Bewegung der Fußgänger auch auf den Spezialfall einer Demonstration anwendbar sein (vgl. Abbildung 3). Damit sollen die Ergebnisse der entwickelten Simulation auch für das Training von Stabs- und Einsatzkräften einsetzbar sein. Ein weiteres Ziel besteht in der Analyse des OpenStreetMap-Datensatzes hinsichtlich der für eine Fußgängersimulation sinnvoll nutzbaren Objektarten. Die Verwendung entsprechender OSM-Daten ist, in Form geospezifischer Datenformate, in das Programm integriert.

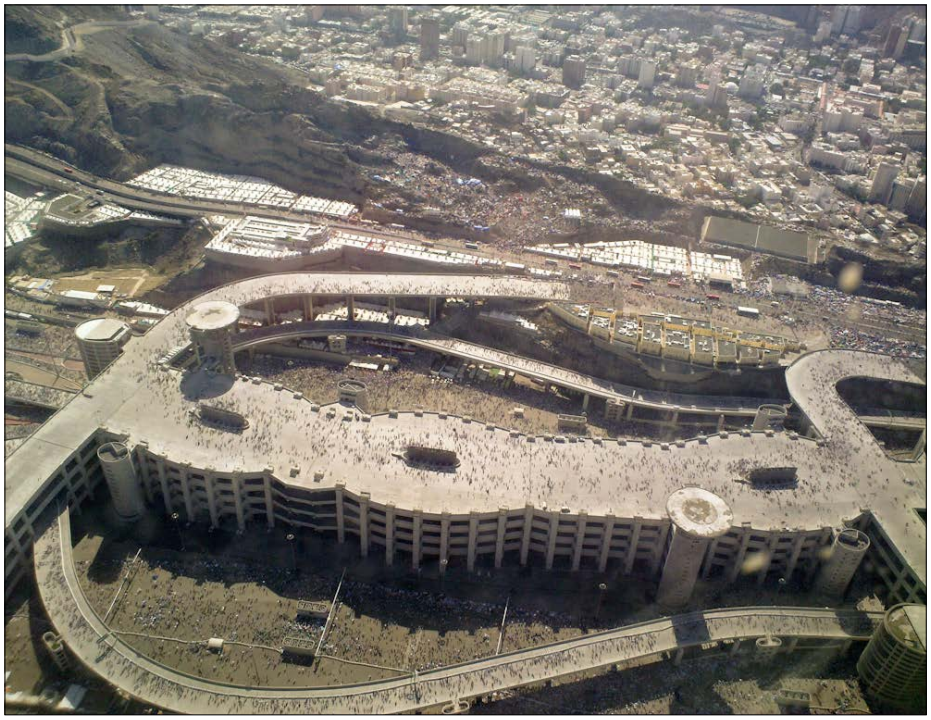

Abbildung 1: Die 5-stöckige Dschamarat-Brücke als Bestandteil der Fußgängerinfrastruktur des Hadsch. Bildquelle: Omar Chatriwala, Creative Commons, Lizenz CC BY-SA 2.0 (https://www.flickr.com/photos/32834977@N03/4142093586).

2 FORSCHUNGSSTAND

2.1 MODELLTYPEN

Es existieren unterschiedliche Modellierungsansätze für die Simulation von Fußgängerströmen. In der Literatur werden Modelle nach dem Aggregationsgrad, dem Raumbezug, dem Verhalten der Personen, dem Zweck und der Verfügbarkeit unterschieden (Kretz 2007, Schadschneider et al. 2009). Der Aggregationsgrad gliedert sich in mikroskopische, mesoskopische und makroskopische Simulationen. Mikroskopische Modellierungsansätze beschreiben jeden Fußgänger individuell. Der mesoskopische Ansatz stellt die Fußgänger ebenso individuell dar, jedoch wird die Bewegung der Fußgänger einheitlich berechnet, womit keine Individualisierung mehr besteht. Beim makroskopischen Ansatz erfolgt sowohl die grafische Darstellung als auch die rechnerische Modellierung für alle Fußgänger als Kollektiv. Beim Raumbezug kann zwischen diskreten Räumen durch Gitterstrukturen und kontinuierlichen Räumen unterschieden werden. Der Zweck der Fußgängersimulation unterscheidet, ob die Modellierung einen universellen Ansatz verfolgt oder auf ein spezifisches Szenario bezogen ist, wie beispielsweise im Fall einer Evakuierungssimulation. Implementierungen in quelloffenen oder kommerziellen Produkten werden durch die Verfügbarkeit unterschieden.

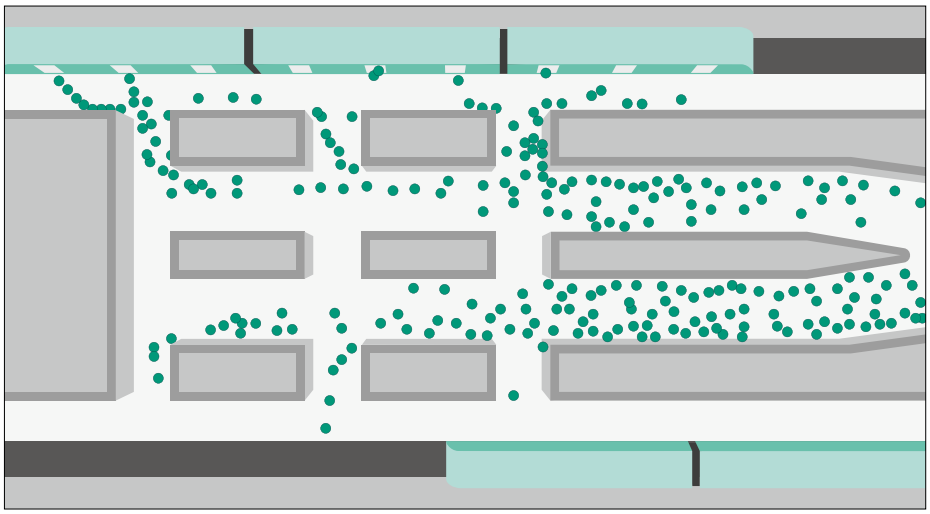

Abbildung 2: Prinzipskizze von Menschen in einem U-Bahnhof (eigene Darstellung)

Abbildung 3: Polizeikessel bei Frankfurter Blockupy-Protesten 2013. Bildquelle: timmy_lichtbild, Creative Commons, Lizenz CC BY 2.0 (https://www.flickr.com/photos/96770691@N03/8918189736).

Drei verschiedene Modelltypen werden im Folgenden genauer vorgestellt: die Modellierung basierend auf zellulären Automaten, die gaskinetische Modellierung und die Kräfte-basierte Modellierung.

Zelluläre Automaten (ZA) sind ein Modellierungsansatz, der in verschiedenen Wissenschaftsdisziplinen für Simulationen verwendet wird (Yuen & Kay 2009). Erste ZA im Kontext von Fußgängersimulationen wurden in den 1990er-Jahren vorgestellt (Fukui & Ishibashi 1999). ZA beschreiben im Allgemeinen einen Raum, welcher durch ein Raster unterteilt wird und dessen Zellen eine Menge definierter Zustände annehmen können. Auf ZA basierende Simulationen nutzen per Definition diskrete Zeitschritte (Toffoli & Margolus 1991). Jeder ZA muss vier Grundbedingungen erfüllen (Burstedde et al. 2001):

- ZA nutzen ein Raster, das in der Regel zweidimensional ist.
- Das Raster hat eine explizite geometrische Form, z. B. rechteckig oder hexagonal.
- Die Anzahl der möglichen Zellzustände muss festgelegt sein.
- Der nächste Zustand einer Zelle wird ausschließlich aus dem Einfluss der direkten Nachbarzellen bestimmt.

Die Fußgängermodellierung mithilfe von ZA ist dem mikroskopischen Aggregationsgrad zuzuordnen. Die Berechnung ist räumlich und zeitlich diskret, da sich die Fußgänger nur von Rasterzelle zu Rasterzelle bewegen können.

Die fluiddynamische bzw. allgemeine gaskinetische Modellierung nimmt an, dass das Bewegungsverhalten, welches sich bei Flüssigkeiten oder Gasen beobachten lässt, denen von Fußgängerströmen ähnelt (Kormanová 2013). Darauf aufbauend werden physikalische Gleichungen, welche die Bewegungen der Partikel beschreiben, auf die Bewegung der Fußgänger adaptiert. Im Falle der fluiddynamischen Herangehensweise handelt es sich um Anpassungen basierend auf den Navier-Stokes-Gleichungen (Wagner 1997). Die allgemeinere gaskinetische Modellierung nutzt den Boltzmann-Ansatz (Henderson 1974). Erste fluiddynamische bzw. gaskinetische Simulationen wurden von Togawa (1955) beschrieben, der die Evakuierung bei Bränden untersucht hat, und Predtechenskii & Milinskii (1978), die den Fluss von Fußgängerströmen in Gebäuden modelliert haben. Weitere Beispiele für die Nutzung dieses Ansatzes finden sich u. a. in Helbing (1990) und Hughes (2000). Die gaskinetische Modellierung berechnet die Menschenmenge immer nach dem Ansatz der makroskopischen Aggregation, was bedeutet, dass das Verhalten einzelner Fußgänger nicht modelliert wird (Schadschneider et al. 2009).

Kräfte-basierte Modelle haben das Alleinstellungsmerkmal, dass sie das zweite Newtonsche Gesetz als Grundlage für die Modellierung von Fußgängerbewegungen nutzen. Die numerische Lösung dieses Modelltyps beruht auf der Berechnung von Differenzialgleichungen, womit sich ein kontinuierliches Simulationsverfahren ergibt. Vom Aggregationsgrad sind sie, wie auch die ZA, den mikroskopischen Modellen zuzuordnen. Es existieren mehrere Modellierungsansätze, z. B. das Magnetic-Field-Modell (Okazaki & Matsushita 1993) und das Zentrifugalkraftmodell (Yu et al. 2005), weiterentwickelt durch Chraibi et al. (2010). Häufig genutzt wird das Social-Force-Modell (SFM) nach Helbing & Molnár (1995). Mithilfe dieses Modells kann die Bewegung einzelner Individuen als Partikel bzw. Agenten beschrieben werden. Diese sind sogenannten sozialen Kräften ausgesetzt. Diese Kräfte beeinflussen die Stärke und Richtung der Bewegung der Agenten. Die Größen dieser Kräfte beschreibt Formel 1 nach Helbing et al. (2005).

Die Formel besteht aus vier Kraftkomponenten, welche auf die Agenten wirken:

- Die erste Kraftkomponente ergibt sich daraus, dass sich der Agent α mit einer gewünschten Geschwindigkeit in Richtung seines Ziels fortbewegen möchte.
- Die zweite Kraftkomponente beschreibt die Abstoßungskräfte zwischen dem Fußgänger α und Hindernissen B in Abhängigkeit von der Entfernung zum Hindernis.
- Die dritte Kraftkomponente beschreibt die abstoßenden Kräfte zwischen zwei Agenten α und β in Abhängigkeit von deren Entfernung und Geschwindigkeiten.
- Die vierte Kraftkomponente beschreibt anziehende Kräfte zwischen den Agenten untereinander (z. B. innerhalb von Gruppen i) und zu bestimmten Objekten (z. B. Schaufenster) mit einer Verweilzeit t.

Dieses Modell hat zwei entscheidende Vorteile: Die Bewegung einzelner Individuen kann modelliert werden und das Modell kann mithilfe von Videodaten zu tatsächlichen Fußgängerströmen relativ leicht empirisch kalibriert werden.

$$\vec{f}_\alpha(t) = \vec{f}_\alpha^{\,0}(\vec{v}_\alpha) + \vec{f}_{\alpha B}(\vec{r}_\alpha) + \sum_{\beta(\neq\alpha)} \vec{f}_{\alpha\beta}(\vec{r}_\alpha, \vec{v}_\alpha, \vec{r}_\beta, \vec{v}_\beta) + \sum_i \vec{f}_{\alpha i}(\vec{r}_\alpha, \vec{r}_i, t)$$

Formel 1: Formel zum Social-Force-Modell nach Helbing et al. (2005)

2.2 MODELLKALIBRIERUNG

Für eine möglichst realitätsnahe Beschreibung der Bewegung von Fußgängern kann das Social-Force-Modell empirisch kalibriert werden. Johansson et al. (2007) und Moussaïd et al. (2010) haben die Bewegung von Fußgängern mittels Video-Tracking in realen Szenarien (z. B. während des Hadsch) analysiert. Weiterhin haben Hoogendoorn & Daamen (2007) sowie Moussaïd et al. (2009) verschiedene Versuchsszenarien im Labor nachgestellt, mit Video aufgenommen und hinsichtlich der Bewegungsparameter analysiert. Modellkalibrierungen quantifizieren die Stärke und die Distanz der Interaktion von Fußgängern untereinander sowie von Fußgängern und Hindernissen. Abbildung 4 zeigt beispielhaft, wie die Stärke der abstoßenden Beschleunigung zwischen den Fußgängern mit zunehmender Distanz geringer wird.

2.3 DARSTELLUNG DER MODELLE

Kräfte-basierte Modelle haben den Vorteil, dass einzelne Individuen modelliert und somit auch in ihrer Bewegung dargestellt werden können. In manchen Anwendungsfällen ist es bei der Darstellung von großen Datenmengen jedoch vorteilhaft, diese zu strukturieren und zu vereinfachen. Bei der Visualisierung einer Fußgängersimulation kann es von praktischem Nutzen sein, diese derart zu abstrahieren, dass anstelle einer individualisierten Darstellung der Fußgänger lediglich der abgeleitete Umriss der gesamten Menschenmenge dargestellt wird. Hierfür eignen sich Clusterverfahren und Methoden zur Ableitung von Hüllen für Punktmengen.

Clusterverfahren haben im Allgemeinen das Ziel, in einer Punktmenge ähnliche Objekte zu identifizieren. Beim Clustern von räumlichen Objekten besteht die Ähnlichkeit darin, dass die Objekte eines Clusters räumlich nah beieinanderliegen. Clusterverfahren werden in drei Typen differenziert: *partitionierende* Verfahren, *dichtebasierte* Verfahren und *hierarchische* Verfahren (Tan et al. 2013).

Ein Vertreter der *partitionierenden Verfahren* ist das K-means-Clustering (MacQueen 1967). Bei diesem bestimmt der anwenderspezifische Parameter *k* die Zahl der Cluster, die entstehen sollen. In dessen Abhängigkeit berechnet der Algorithmus Zentroide, welche die Mittelpunkte der Cluster bilden. Anschließend werden mittels Distanzberechnung alle Objekte dem jeweils nächstliegenden Zentroid zugeordnet. Die Komplexität des K-means-Algorithmus ist abhängig von der Dimension *d* und der Menge der Cluster *n* und beträgt $O(n^{dk+1})$ (Inaba et al. 1994).

Die Methode des „Density-Based Spatial Clustering of Application with Noise" (DBSCAN) ist ein Beispiel für ein *dichtebasiertes Clustering*-Verfahren (Ester et al. 1996). Die Vorbedingung des Algorithmus besteht in der Spezifizierung der Parameter Epsilon und MinPts. Epsilon beschreibt den maximalen Abstand zwischen zwei Objekten, die zum selben Cluster gehören. MinPts legt fest, wie viele Punkte innerhalb eines Clusters die Bedingung von Epsilon

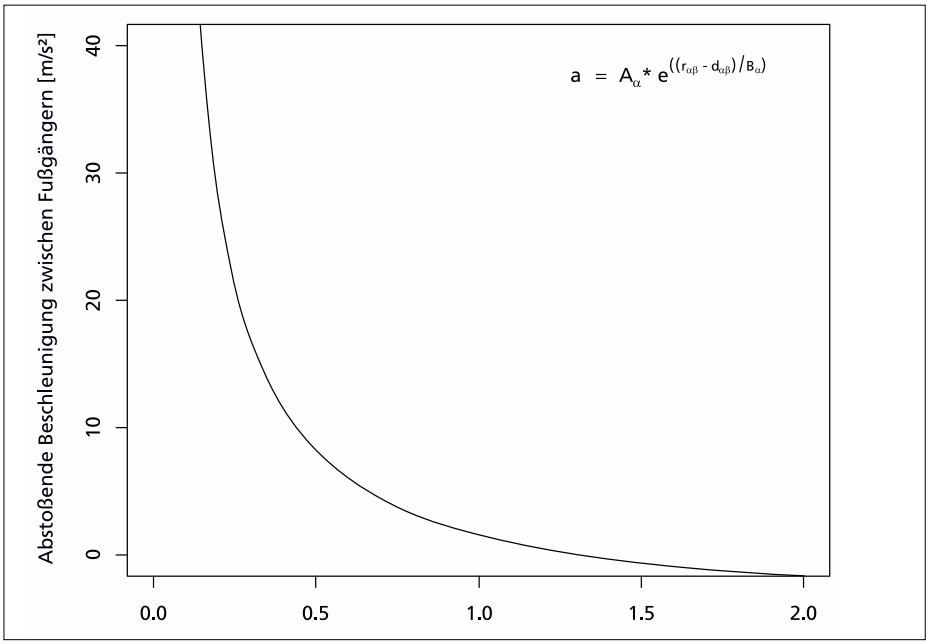

Abbildung 4: Die dritte Kraftkomponente von Formel 1 beschreibt die Interaktionsstärke zwischen Fußgängern. Generalisiert resultiert das in der abstoßenden Beschleunigung a [m/s²] zwischen Fußgängern (mit Radius r) im Verhältnis zur Distanz d [m] zwischen diesen. Die in der Abbildung implementierte Formel beschreibt die zugrunde liegende Berechnungsbasis. Bei dieser wird der Winkel vernachlässigt, in dem sich die Fußgänger aufeinander zu bewegen, wobei A die Interaktionsstärke und B deren Reichweite beschreibt. Diese Parameter können empirisch bestimmt werden.

mindestens erfüllen müssen, damit ein Cluster entsteht. Typische Parametrisierungen dieser Werte unterteilen die zu clusternden Objekte in Kernpunkte, Grenzpunkte und Ausreißer. Dabei erfüllen Kernpunkte die Bedingungen beider Parameter, Grenzpunkte nur die Bedingung des Parameters Epsilon und Ausreißer keine von beiden. Im Resultat des Clusterings bilden die Grenzpunkte den Rand des Clusters und die Kernpunkte die Innenfläche. Ausreißer gehören keinem Cluster an. Ester et al. (1996) spezifizieren die durchschnittliche Komplexität des DBSCAN-Verfahrens mit $O(n \cdot \log n)$. Im schlechtesten Fall hat der Algorithmus eine Komplexität von $O(n^2)$ (Schubert et al. 2017).

Hierarchisches Clustering basiert auf Baumstrukturen. Als Beispiel sei der Single-Linkage-Algorithmus genannt (Jain & Dubes 1988). Das Lösungsverfahren ist iterativ. Im ersten Lösungsschritt werden alle Objekte einem eigenen initialen Cluster zugeordnet. Die jeweils einander nächstgelegenen Cluster werden in einem zweiten Schritt zu einem neuen Cluster aggregiert. Dieser Prozess wird solange wiederholt, bis ein Cluster alle Objekte umfasst. Das dadurch entstandene Dendrogramm repräsentiert damit eine vollentwickelte Baumstruktur. Für ein konsistentes Clustering-Ergebnis muss auf ein spezifisches Level des Dendrogramms zugegriffen werden. Der Single-Linkage-Algorithmus hat eine Komplexität von $O(n^3)$ (Murtagh & Contreras 2012). Mit Weiterentwicklungen, welche sich spezifisch auf die Verbesserung der Performanz des Single-Linkage-Verfahrens konzentrieren, erreicht dieses eine Komplexität von $O(n^2)$ (Sibson 1973).

Tabelle 1 visualisiert die drei Clusterverfahren anhand vier verschieden angeordneter Szenarien. Diese sind so strukturiert, dass die Vor- und Nachteile der jeweiligen Verfahren hervorgehoben werden. Zusätzlich lässt die hinzugefügte Zeitmessung zur Erstellung der jeweiligen Cluster Rückschlüsse auf die Komplexität dieser zu. Für die Modellierung der Fußgängersimulation wurde das DBSCAN-Verfahren eingesetzt. Durch dessen Implementierung können Gruppen von Menschenmengen realitätsnah identifiziert werden, was insbesondere durch das erste und vierte Szenario hervorgehoben wird.

Für die **Ableitung von Hüllen** für Punktwolken im zweidimensionalen Raum kön-

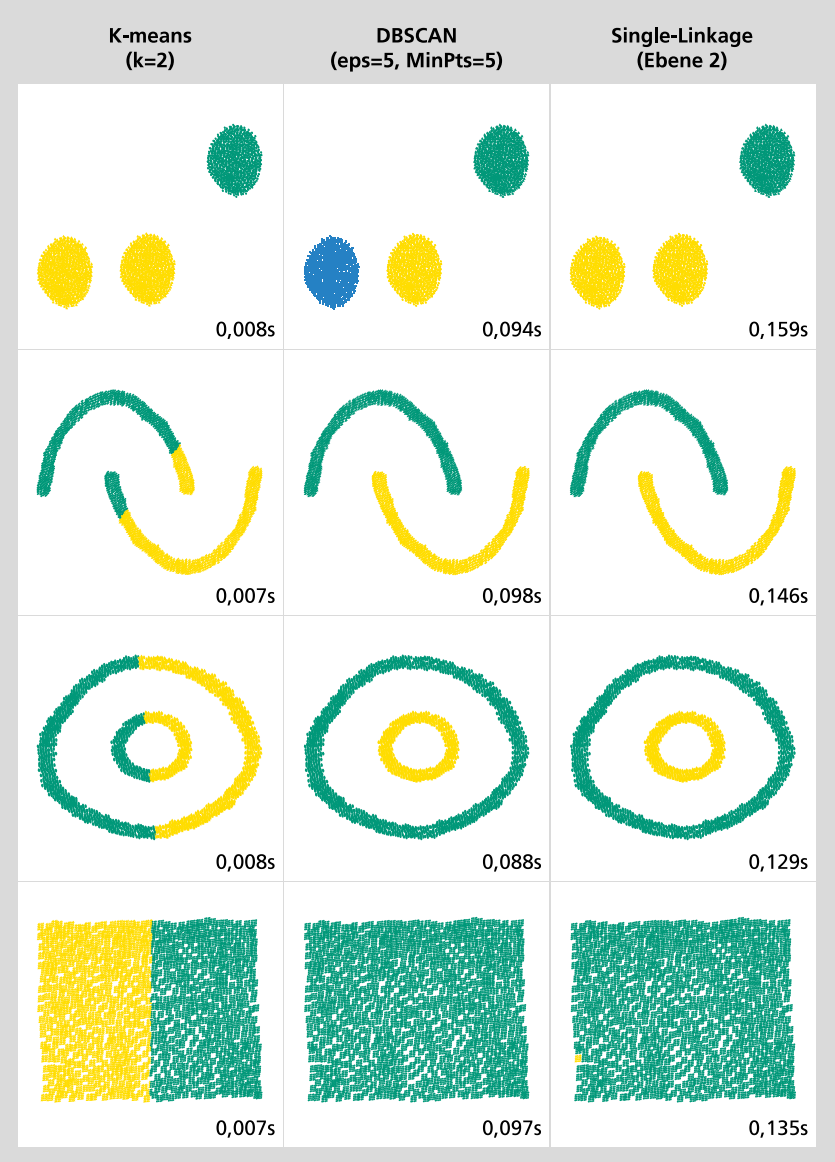

Tabelle 1: Vergleich dreier Clustering-Verfahren: dem partitionierenden Clustering (K-means), dem dichtebasierten Verfahren (DBSCAN) und dem hierarchischen Verfahren (Single-Linkage). Unterschiedliche Farben der Punktmengen symbolisieren die jeweiligen Cluster. Die Zeitmessungen in der Tabelle quantifizieren die Komplexität der Verfahren in Bezug zu dem jeweiligen Szenario.

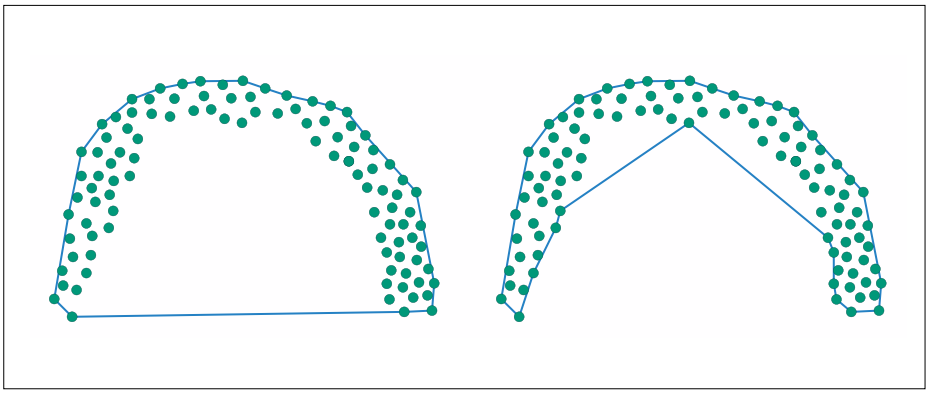

Abbildung 5: Die zwei Methoden der Ableitung von Hüllen. Links: Konvexe Hülle als Ableitung der Menge an Fußgängern (grüne Kreise). Rechts: Beispiel einer konkaven Hülle als Ableitung der Fußgängermenge.

Kriterium	CROWD-SIM	FDS/EVAC	JUPEDSIM	MENGE	MOMENTUM	PEDSIM	VADERE
Art	Simulator	Simulator	Framework	Framework	Framework	Bibliothek	Framework
Entwickler	Privat-person	VTT Finnland	Forschungs-zentrum Jülich	UNC Chapel Hill	TU München	ETH Zürich/ Privatperson	Hochschule München
Programmier-sprache	JavaScript	Fortran	C++, Python	C++	Java	C++	Java
Simulations-modell	Social-Force-Modell	Social-Force-Modell	verschiedene	verschiedene	verschiedene	Social-Force-Modell	verschiedene
Lizenz	MIT	gemeinfrei	LGPL	Apache-Lizenz 2.0	gemeinfrei	GPLv3	LGPL
Tutorials	nein	nein	ja	nein	nein	nein	nein
Dokumentation	nein	vorhanden	vorhanden	teilweise	teilweise	vorhanden	teilweise
GUI	ja	ja	nein	nein	ja	nur in Demo	ja
Beschreibungs-sprache für Szenarien	JSON	TXT	XML	XML	XML	XML	JSON
Aufgaben-gebiet (laut Eigenbeschrei-bung)	Browser-GUI für exempla-rische Fußgängersi-mu-lationen	Simulation von Men-schenmengen (Evac) als Erweiterung zu FDS	Speziell für Evakuierungs-simulationen, aber auch Anderes	Simulation der Bewegung von Men-schenmengen	Framework für agenten-basierte Simulation	Evakuierung in Gebäuden; große Menschen-mengen im Freien	Simulation mikrosko-pischer Fußgänger-dynamiken
Datenquelle	https://github.com/Jacq/crowd-sim	https://github.com/firemodels/fds	https://github.com/JuPedSim/JuPedSim	https://github.com/MengeCrowdSim/Menge	https://github.com/tumcms/MomenTUM	https://github.com/chgloor/pedsim	https://gitlab.lrz.de/vadere/vadere
Literaturquelle	(Crowd-Sim 2016)	(Korhonen & Hostikka 2009)	(Kemloh & Chraibi 2016)	(Curtis et al. 2016)	(Kielar et al. 2016)	(Gloor 2018)	(Seitz 2016)

Tabelle 2: Übersicht über quelloffene, plattformunabhängige Programme zur mikroskopischen Fußgängersimulation

nen zwei Methoden verwendet werden: die *konvexe Hülle* und die *konkave Hülle*. Eine konvexe Hülle definiert sich als kleinste konvexe Menge, in der die zu untersuchenden Punkte enthalten sind. Dabei bedeutet Konvexität, dass alle Verbindungsstrecken zweier beliebiger Punkte ebenfalls innerhalb der konvexen Hülle liegen (Preparata & Shamos 1985, TIOBE Software B. V. 2018)

Im Gegensatz dazu kann die konkave Hülle jede mögliche Form annehmen, wovon die konvexe Hülle ein Sonderfall ist. Aufgrund einer fehlenden strikten Definition ist die Ableitung konkaver Umrisslinien komplexer, sodass für eine Punktwolke üblicherweise eine Vielzahl an konkaven Hüllen existiert. Es gibt verschiedene Ansätze zur Implementierung, wie beispielsweise die Methode der Alpha-Shapes (Edelsbrunner et al. 1983) oder die Methode der Chi-Shapes (Duckham et al. 2008). Beiden Methoden gemein ist, dass eine konkave Hülle entsteht, die die Eigenschaft hat, von der entsprechenden konvexen Hülle eingeschlossen zu werden. Die Methode der Chi-Shapes legt mit dem Parameter Chi fest, wie stark das entstehende Shape in Richtung Konvexität strebt. Ein Beispiel für das Aussehen eines solchen Chi-Shapes im Vergleich zur konvexen Hülle ist in Abbildung 5 zu sehen. Es wird deutlich, dass die abgeleitete Umrisslinie besser der Form der Punktmenge entspricht als die konvexe Hülle.

3 SOFTWAREÜBERBLICK

Dieser Abschnitt gibt einen Überblick über verfügbare Software zur Fußgängersimulation. Eine Arbeit von Kretz (2007) zählt nahezu 60 Modellierungsansätze auf, von denen für ungefähr ein Drittel eine Implementierung verfügbar ist bzw. war. Kuligowski et al. (2010) zählen 26 Simulationsprogramme, von denen ungefähr die Hälfte aktuell existiert. Die überwiegende Mehrheit dieser Programme steht unter kommerziellen Lizenzen. Seitz (2016) gibt einen Überblick über sechs quelloffene Fußgängersimulation-Frameworks. Diese Arbeiten wurden als Grundlage genommen, aktualisiert, um bekannte Software erweitert und den Umfang der Vergleichskategorien erhöht. Tabelle 2 gibt eine Übersicht über quelloffene Fußgängersimulationen. Alle aufgeführten Programme sind plattformunabhängig und simulieren Fußgänger nach dem Prinzip des mikroskopischen Aggregationsgrads. Auf das Hinzufügen von Verkehrssimulationen, wie MATSim (Horni

HINDERNISARTEN	KEY (SCHLÜSSEL)	VALUE (WERT)	Ø-GRÖßE	OBJEKTART	GEOMETRIETYP
Gebäude					
	Buildings	yes		unspez. Gebäudetyp	Area
Künstliche Hindernisse					
	Barrier	bollard	0.1 m · 0.1 m	Poller	Node
		cycle_barrier	1.0 m · 0.1 m	Fahrradbarriere	Node
		fence		Zaun	Way
		guard_rail		Leitplanke	Way
		lift_gate	3.0 m · 0.2 m	Schranke	Node
		wall		Mauer	Way
	Waterway	canal		Kanal	Way
Natürliche Hindernisse					
	Natural	scrub		Sträucher	Area
		tree	1.0 m · 1.0 m	Baum	Node
		water		Wasser	Area
	Waterway	river		Fluss	Way
Stadtmöbel					
	Amenity	bench	2.0 m · 0.5 m	Parkbank	Node
		bicycle_parking	1.0 m · 0.1 m	Fahrradparkplatz	Node
	Highway	street_lamp	0.2 m · 0.2 m	Straßenlaterne	Node
	Leisure	fitness_station	0.5 m · 0.5 m	Fitnessstation	Node
		picnic_table	2.0 m · 2.0 m	Picknicktisch	Node
	Man_Made	drinking_fountain	0.5 m · 0.5 m	Trinkwasserbrunnen	Node
Straßenmobiliar					
	Highway	bus_stop	0.1 m · 1.0 m – 3.0 m · 10.0 m	Bushaltestelle	Node
		speed_camera	0.2 m · 0.2 m	Blitzer	Node
		traffic_sign	0.1 m · 0.1 m	z. B. Stoppschild	Node
		traffic_signal	0.1 m · 0.1 m	Ampel	Node
Technische Einrichtungen					
	Amenity	charging_station	0.3 m · 0.3 m	Ladesäule	Node
		post_box	0.5 m · 0.5 m	Briefkasten	Node
		vending_machine	1.0 m · 1.0 m	Verkaufsautomat	Node
		telephone	1.0 m · 1.0 m	Telefonzelle	Node
		toilets	2.0 m · 2.0 m	Toilette	Node
		waste_basket	0.5 m · 0.5 m	Mülleimer	Node
	Emergency	fire_hydrant	0.2 m · 0.2 m	Hydrant	Node
		phone	0.2 m · 0.2 m	Rettungstelefon	Node
	Man_Made	mast	0.2 m · 0.2 m	Mast	Node
Sonstiges					
	Amenity	fountain	5.0 m · 5.0 m	Springbrunnen	Node
	Historic	memorial	5.0 m · 5.0 m	Denkmal	Node
		milestone	0.5 m · 0.5 m	Meilenstein	Node
	Man_Made	water_well	1.0 m · 1.0 m	Brunnen	Node

Tabelle 3: *Auswahl an Hindernissen im Sinne einer Fußgängersimulation nach den MapFeatures des OSM-Wiki (https://wiki.openstreetmap.org/wiki/Map_Features)*

et al. 2016) und SUMO (Krajzewicz et al. 2012), oder agentenbasierter Simulationsframeworks, wie Repast Simphony (North et al. 2013), wird verzichtet. Diese bieten ähnliche Möglichkeiten zur Fußgängersimulation, sind jedoch hinsichtlich ihrer thematischen Zielstellung generischer positioniert.

4 OPENSTREETMAP-DATEN

Der frei verfügbare Datensatz von OpenStreetMap (OSM) eignet sich als Datengrundlage für die Simulation von Menschenmengen in urbanen Räumen, da existierende Geoobjekte insbesondere in Großstädten detailliert kartiert sind. Semantische Informationen der Geodaten in Form von Attributen (Tags) können genutzt werden, um für die Fußgängersimulation relevante von nichtrelevanten Daten zu trennen. Neben der Modellierung aller Arten von Hindernissen (vgl. Tabelle 3) können beispielsweise vorhandene Informationen zur Breite von Straßen und Wegen verwendet werden. Bei einer weltweiten durchschnittlichen Erhebungsquote von 1,1 % des Attributs *width* als Eigenschaft von Straßen bzw. Wegen ist die Verfügbarkeit dieser Information jedoch noch relativ niedrig (Quelle: taginfo.openstreetmap.org). Allerdings haben Stichproben in deutschen Großstädten gezeigt, dass dort ein höherer Anteil an Wegen dieses Attribut hat.

Für die Umsetzung einer Fußgängersimulation können die Primärdaten von OSM genutzt werden. Dessen Datenstruktur besteht aus den drei grundlegenden Elementen: Punkte (Nodes), Linien (Ways) und Relationen (Relations). Eine Sonderform der Linien bilden die Flächen (Areas) (Ramm & Topf 2010). Eine Auswahl von für die Fußgängersimulation im urbanen Umfeld besonders relevanten Hindernisarten wurde bereits in einem früheren Beitrag (Meinert et al. 2018) vorgestellt. Tabelle 3 erweitert diese mittels den zugehörigen OSM-Tags, bestehend aus Schlüssel und Wert, den geschätzten durchschnittlichen Größen der Punktelemente, der deutschen Bezeichnung der Objektart sowie dem überwiegend genutzten Geometrietyp.

Bei der Implementierung nach Tabelle 3 müssen einige Aspekte beachtet werden:

- Als Nodes gespeicherte Objekte, die in der Simulation als Hindernis betrachtet werden, müssen mit einer Art Standardgröße modelliert werden (vgl. Spalte durchschnittliche Größe).
- Einige Objekte im OSM-Datensatz sind nicht mit dem für eine Fußgängersimulation geeigneten Geometrieelement aufgenommen. Ein Beispiel hierfür sind Schranken, die im Wesentlichen als Nodes registriert werden, für eine Fußgängersimulation allerdings Linien sein müssten.
- Im OSM-Datensatz gibt es teilweise Redundanzen. Beispielsweise werden Objekte gleichen Typs mit unterschiedlichen Geometrietypen aufgenommen. Für eine optimale Fußgängersimulation gilt es, solche redundanten Daten zu vermeiden.

5 PROTOTYPISCHE IMPLEMENTIERUNG

Als Modellgrundlage für den jCrowdSimulator wurde das Kräfte-basierte Social-Force-Modell ausgewählt. Die Vorteile dieses Modells liegen in der leichten Implementierung mit gleichzeitig realitätsnaher, mikroskopischer Berechnung der Bewegung der Personen einer Menschenmenge. In der Anwendung wurden die ersten drei Terme des Modells umgesetzt (vgl. Gleichung 1), für den vierten Term (anziehende Kräfte wie bspw. Schaufenster) fehlen teilweise geeignete empirische Untersuchungen. Für die generalisierte Darstellung der Menschenmenge in Form eines Umrisses ist sowohl das DBSCAN-Clusterverfahren als auch die Möglichkeit der Berechnung von konkaven und konvexen Hüllen implementiert.

Von den in Tabelle 2 referenzierten Implementierungen von Fußgängersimulationen erlaubt lediglich MomenTuM den Import von Geodaten (Kielar et al. 2016), indem XML-Rohdaten von OpenStreetMap gelesen werden können. Gängige Datenformate Geographischer Informationssysteme, wie etwa Esri-Shape (Esri 1998) oder dem vom Open Geospatial Consortium (OGC) definierten Well-known Text (WKT) (Open Geospatial Consortium 2015), können nicht importiert werden. Da Geodaten zum urbanen Umfeld jedoch häufig in diesen Formaten vorliegen, ist es für Fußgängersimulationen nützlich, die entsprechende Formatunterstützung durch die Simulati-

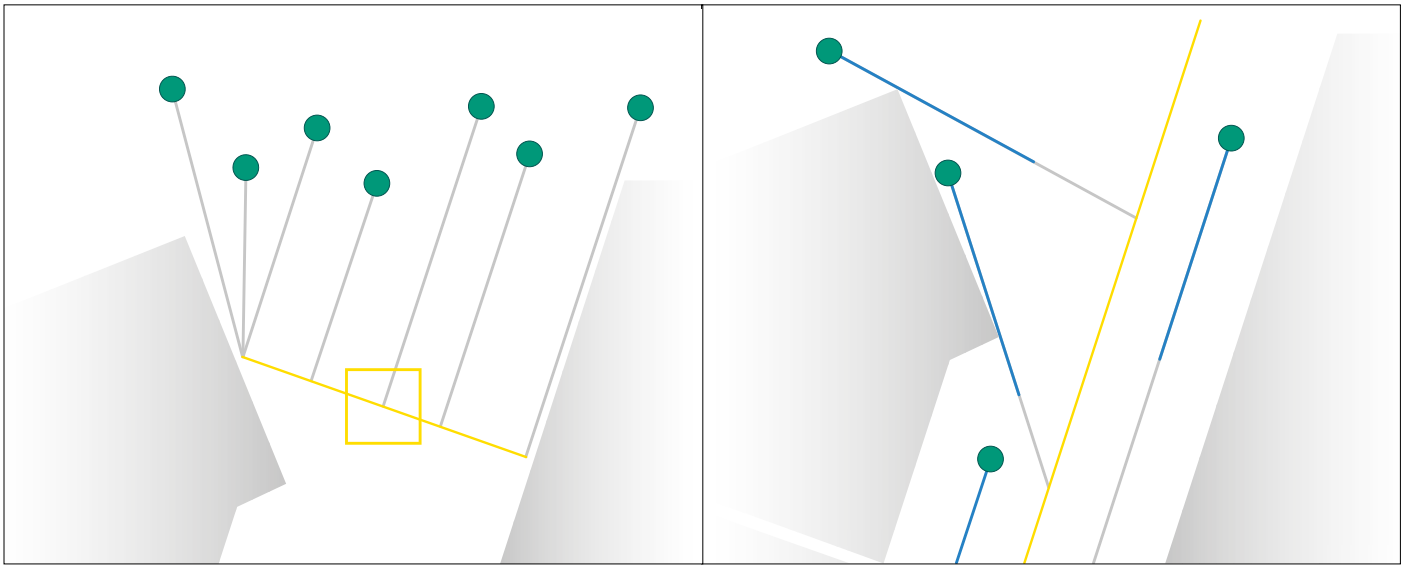

Abbildung 6: Darstellung der Wegefindung mithilfe von Wegpunkt-Vertikalen und Wegpunkt-Achsen. Links: Fußgänger laufen nächstgelegenen Punkt, visualisiert mittels Zielvektor, der Vertikalen an. Rechts: Zwei Fußgänger nutzen die Wegpunkt-Achse zur Re-Orientierung, bis der nächste Wegpunkt wieder sichtbar ist.

onsumgebung bereitzustellen. Im Folgenden wird daher eine leichtgewichtige, prototypische Implementierung vorgestellt, die diese Anforderung erfüllt und ausschließlich als Fußgängersimulation fungiert. Die entstandene Programmbibliothek ist in der Programmiersprache Java – seit langer Zeit auf Platz 1 im Tiobe-Index für Programmiersprachen (TIOBE Software B. V. 2018) – implementiert und steht unter der LGPL-Lizenz (https://github.com/FraunhoferIVI/jCrowdSimulator) zur Verfügung. Sie gliedert sich dabei in einen Programmteil, der die Simulationsalgorithmen implementiert, und einen Teil, der zur Visualisierung genutzt wird. Ersterer kann dabei unabhängig vom zweiten verwendet werden.

5.1 MERKMALE

Der Simulator unterstützt den Import von Daten, die das urbane Umfeld modellieren, z. B. Gebäude und Hindernisse. Hierfür können OSM-Daten, nach Umwandlung in das Shape- oder WKT-Format, genutzt werden. Weiterhin können Wegpunkte, die die einzelnen Agenten der Menschenmenge anlaufen, geladen werden. Schließlich ist es möglich, die Startpositionen der Agenten als Punktmenge zu importieren. Dabei werden alle Fußgänger einer Gruppe zugeordnet, die gemeinsame Wegpunkte hat. Derzeit ist nur die Verarbeitung einer Gruppe möglich, eine Erweiterung auf mehrere Gruppen von Fußgängern, die unterschiedliche Wegpunkte anlaufen, ist jedoch vorgesehen. Der Datenimport unterliegt zwei Validierungskriterien, welche den Algorithmus der Wegefindung unterstützen. Einerseits müssen alle Agenten eine freie Sichtachse auf den ersten Wegpunkt ihrer Gruppe haben. Weiterhin dürfen zwischen zwei aufeinander folgenden Wegpunkten keine Hindernisse die Sichtachse blockieren. Beide Kriterien werden direkt nach dem Import der Daten überprüft und garantieren, dass die Fußgänger ihre Wegpunktliste ablaufen können.

Die Orientierung der Fußgänger basiert auf der importierten Wegpunktliste. Die enthaltenen Punkte werden in der Reihenfolge der Liste abgelaufen. Wenn die Fußgänger den Zielpunkt erreichen und somit alle Wegpunkte abgelaufen haben, verharren sie an diesem. Um den Fluss der Menschenmenge realitätsnaher zu gestalten, wurde die sogenannte Wegpunkt-Vertikale konzipiert (vgl. Abbildung 6). Diese

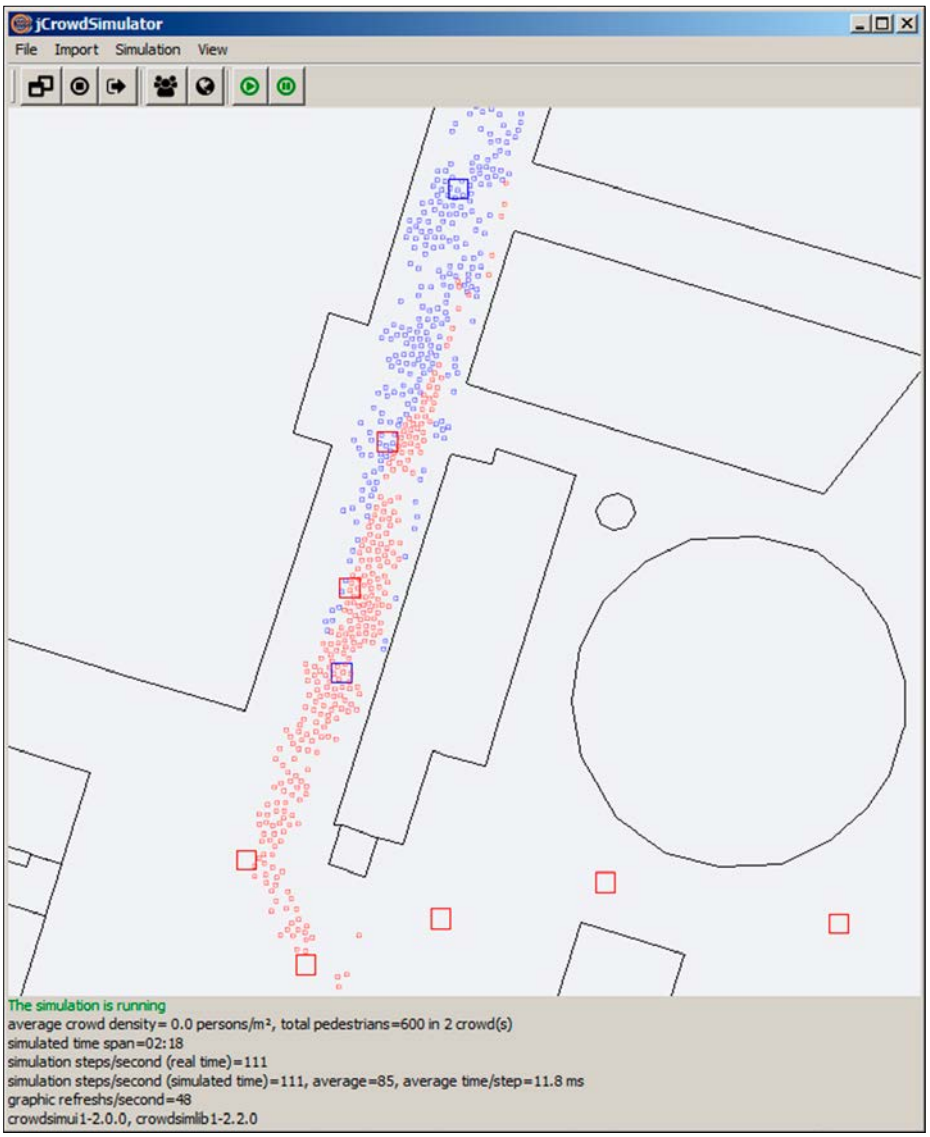

Abbildung 7: Ausschnitt aus der entwickelten Fußgängersimulation. Zwei Fußgängergruppen (rot und blau) laufen in entgegengesetzter Richtung durch einen Abschnitt der Prager Straße in Dresden, welcher durch OSM-Daten visualisiert wird. Screenshot von jCrowdSimulator, Programm und Quellcode verfügbar unter: https://github.com/FraunhoferIVI/jCrowdSimulator.

Vertikalen schneiden die Wegpunkte und liegen senkrecht zur Achse zwischen den Wegpunkten. Ihre Länge definiert sich aus einer benutzerdefinierten Maximallänge, die beispielsweise aus dem OSM-Attribut *width* abgeleitet werden kann und aus der Begrenzung durch umliegende Hindernisse. Das linke Bild in Abbildung 6 visualisiert eine von Gebäuden begrenzte Vertikale und die Zielvektoren der Fußgänger, die auf sie zeigen. Durch die Nutzung des zum einzelnen Fußgänger nächstgelegenen Punkts auf der Vertikalen als Zielpunkt wird eine gleichmäßige Verteilung der Personen innerhalb einer Menschenmenge im Straßenquerschnitt erreicht.

Eine mögliche Störanfälligkeit der vorgestellten Wegefindung besteht darin, dass Fußgänger ihren nächsten Wegpunkt nicht sehen, z. B. da die Sichtachse zum nächsten Wegpunkt durch ein Gebäude verdeckt wird. Dann wird versucht, den Fußgänger auf die Achse zwischen zwei benachbarten Wegpunkten zurückzuführen. Das rechte Bild in Abbildung 6 zeigt die Funktionsweise. Die Abbildung zeigt, dass ein Fußgänger aufgrund eines Gebäudevorsprungs keine direkte Sichtachse zum nächsten Wegpunkt hat. Übergangsweise nutzt dieser Agent einen Punkt auf der Wegpunkt-Achse als Zielpunkt für seine Bewegung. Wenn die Sichtachse zum eigentlichen Wegpunkt wieder frei ist, wird dieser als Ziel reaktiviert. Falls auch die Sichtachse zur Wegpunkt-Achse verdeckt ist, hat der Agent seine letzte Position, an

der der Wegpunkt sichtbar war, gespeichert und versucht zu dieser zurückzukehren.

Jeder Fußgänger hat während der Fortbewegung entlang der für ihn definierten Wegpunkte eine eigene Normal- und Maximalgeschwindigkeit. Die Fußgänger weichen im Fall von Verzögerungen bis zu ihrer Maximalgeschwindigkeit von ihrer Normalgeschwindigkeit ab. Um eine Annäherung an ein realistisches Verhalten von Fußgängern innerhalb einer Menschenmenge zu erreichen, ist eine zufällige Verteilung dieser Geschwindigkeiten implementiert. Die konkreten Werte für die Geschwindigkeiten bzw. weitere Parameter, z. B. zur Fußgänger-Fußgänger-Interaktion, resultieren aus den empirischen Kalibrierungen, welche in Kapitel 2.2 vorgestellt wurden. Zwei leicht voneinander abweichende Kalibrierungsvarianten sind implementiert – das Modell von Helbing et al. (2005) und das von Johansson et al. (2007).

Ein zentraler Bestandteil der Simulationsentwicklung lag in der Erstellung einer grafischen Benutzeroberfläche. Abbildung 7 zeigt einen Ausschnitt, bei welchem zwei jeweils 600 Personen umfassende Menschenmengen simuliert werden. Um die Hindernisse zu modellieren, wurden OSM-Daten genutzt.

Die Programmoberfläche besteht aus der Simulationssteuerung und der Simulationsansicht. Die Sichtbarkeiten von Objekten und deren Eigenschaften, z. B. Kraftvektoren, die auf die Agenten wirken, können feingranular eingestellt werden. Weiterhin ist es möglich die Geschwindigkeitsparameter der Fußgänger sowie die Geschwindigkeit der Simulation zu manipulieren. In der Simulationsansicht des Programms wird die Simulation in Echtzeit mit den ausgewählten Elementen visualisiert.

Die im Abschnitt 2.3 beschriebene vereinfachte Darstellung einer Menschenmenge mithilfe eines charakterisierenden Umrisses ist implementiert. Es kann zwischen den Optionen konvexe und konkave Hülle sowie der Nutzung von Clustering gewählt werden. Abbildung 8 zeigt beispielhaft die Ableitung einer konkaven Hülle mit und ohne Clustering.

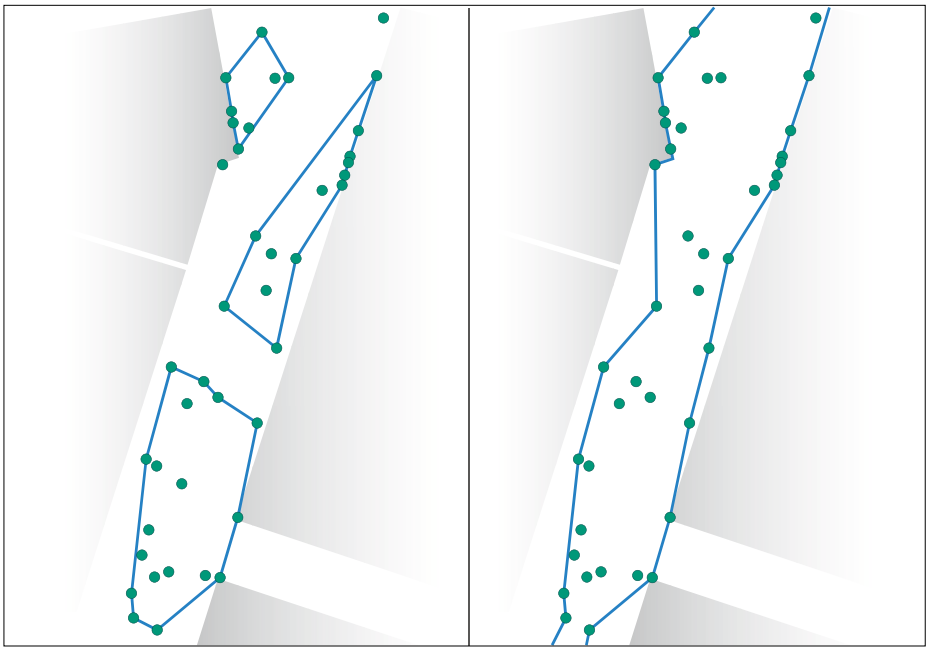

Abbildung 8: Vergleich der Ableitung konkaver Hüllen mit und ohne Clustering: Links: Clusterbedingte Gruppenbildung aufbauend auf der Ableitung der Fußgänger als konkave Hülle. Rechts: Originäre konkave Hülle ohne die Anwendung eines Clustering-Verfahrens.

5.2 LEISTUNGSFÄHIGKEIT

Die entwickelte Fußgängersimulation wurde einem Performancetest unterzogen. Die Basis bildeten achtzehn Datenpaare mit Shape-Dateien zwischen 300 und 100.000 Fußgängern und den korrespondierenden Berechnungszeiten. Die Zeiten sind definiert als die durchschnittliche Berechnungszeit pro Simulationsschritt in Millisekunden. Um eine belastbare Zeit zu erhalten, wurde die Simulation eine vorab definierte Zeit lang durchgeführt. Die daraus resultierende lineare Laufzeitentwicklung ist in Abbildung 9 dargestellt. Diese Laufzeit kann durch die Nutzung einer Quadtree-Datenstruktur erzielt werden (Finkel & Bentley 1974). Der Quadtree ermöglicht einen performanten räumlichen Zugriff auf die Position der Fußgänger, was insbesondere für die Fußgänger-Fußgänger-Interaktion relevant ist, da bei dieser ein einzelner Fußgänger nur mit den in seiner unmittelbaren Umgebung befindlichen anderen Fußgängern interagiert. Durch die $O(n)$-Komplexität des Quadtrees (Samet 2006), entspricht die Simulation derselben Komplexität. Wenn parallel zur Laufzeit die Visualisierung der Menschenmenge mit konvexen bzw. konkaven Hüllen oder Clustering dargestellt werden soll, sinkt die Leistungsfähigkeit entsprechend. Die konvexe Hülle nach dem Graham-Algorithmus hat eine Komplexität von $O(n \cdot \log n)$ (Graham 1972). Das gleiche gilt für konkave Hüllen berechnet mit dem Chi-Algorithmus (Duckham et al. 2008). Für Clustering-Verfahren liegt die Berechnungszeit im Allgemeinen deutlich darüber. Das implementierte DBSCAN-Verfahren hat eine Komplexität von $O(n^2)$ (vgl. Kapitel 2.3), durch welche die Performance bei sehr großen Menschenmengen schnell abnimmt.

6 FAZIT UND AUSBLICK

Mit dem vorgestellten ¡CrowdSimulator ist es gelungen, eine Open-Source-Software zu entwickeln, die eine leistungsfähige, mikroskopische Modellierung großer Menschenmengen bietet. Im Vergleich zu weiteren, frei verfügbaren Programmen (vgl. Tabelle 2) ist der Import von Geodaten, speziell OSM-Daten, möglich. Unter den Simulatoren besitzt die Anwendung das Alleinstellungsmerkmal, in Java entwickelt zu sein. Dieser Aspekt ist besonders hervorzuheben, da Java gerade im Open-Source-Spektrum eine hohe Verbreitung in der Entwickler-Community hat. Durch die vorhandene grafische Benutzeroberfläche mit diversen Steuerungsparametern und den intuitiv nutzbaren Importoptionen zum Aufbau eines individuellen Szenarios ist der Simulator für die Anwender leicht verständlich und verwendbar.

Es ist auf mehreren Ebenen geplant, die vorgestellte Modellierung weiterzuentwickeln. Hier gilt es insbesondere, den implementierten Wegefindungsalgorithmus zu erweitern, der neben den Kräftemodellen maßgeblich den Realismus der Simulation

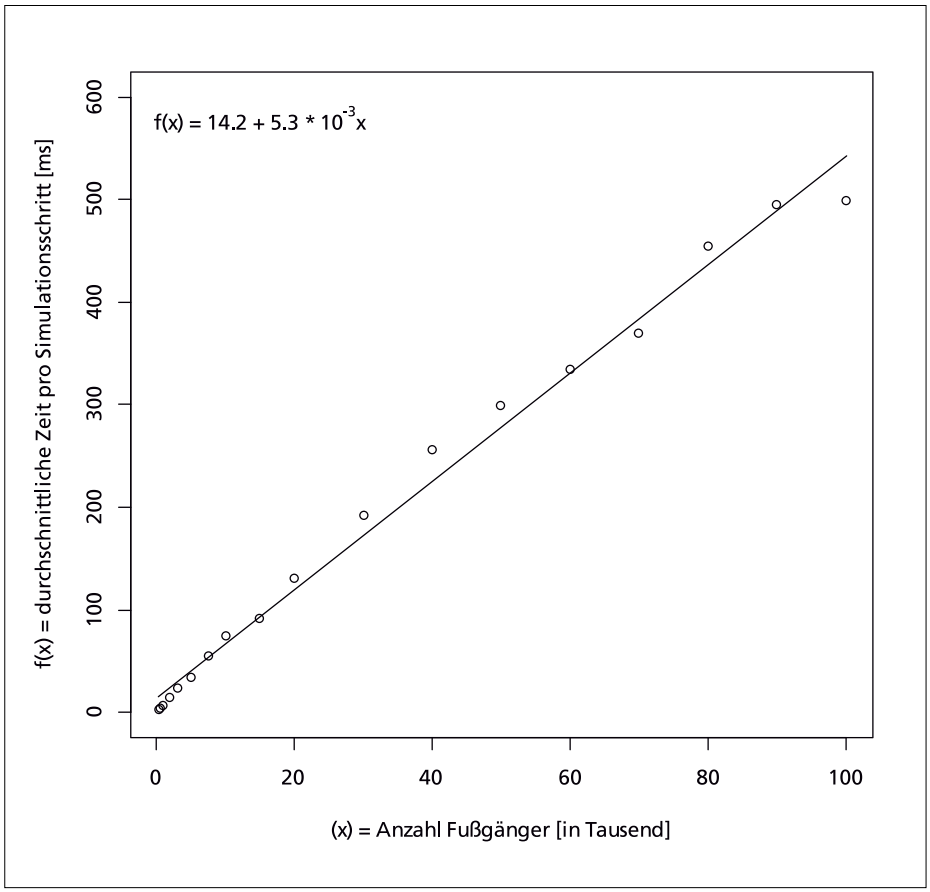

Abbildung 9: Ergebnis des Performancetests. Dieser zeigt die Abhängigkeit der durchschnittlichen Berechnungsdauer pro Simulationsschritt in Millisekunden zur Anzahl der Fußgänger.

bestimmt. Dieser kann weiter verbessert werden, indem Fußgänger vernetzt werden, um sich untereinander bei der Orientierung zu unterstützen. Eine weitere Verbesserung betrifft eine intelligente Raumausnutzung, bei der die Agenten analysieren, in welcher Richtung dichtere Menschenmengen existieren, um diese zu vermeiden.

Eine bisher nicht empirisch untersuchte Fragestellung findet sich im vierten Term des Social-Force-Modells (vgl. Kräfte-basierte Modelle in Kapitel 2.1). Offen ist hier, inwieweit Points of Interest anziehende Kräfte auf Fußgänger hervorrufen. Praktische Beispiele hierfür sind Schaufenster von Geschäften, Straßenmusikanten oder in Bezug auf Demonstrationen stationäre Kundgebungen. Hier fehlt derzeit ein mathematisches Modell, das die wirkenden Kräfte beschreibt und empirische Untersuchungen mit geeignetem Videomaterial, um die Parameter eines solchen Modells zu bestimmen. Neuartige drohnenbasierte Videoüberwachungssysteme (Almer et al. 2016) könnten zudem einen Beitrag leisten, die Parameter des Social-Force-Models noch genauer als bisher möglich zu kalibrieren.

Durch die gleichzeitige Simulation mehrerer Menschenmengen können zudem komplexere Szenarien, wie beispielsweise im Spannungsfeld von Demonstrationen und Gegendemonstrationen, entworfen werden. Beispielsweise können die Geschwindigkeitsparameter der Menschenmenge genutzt werden, um aggressive (= schneller) oder defensive (= langsamer) Verhaltensweisen zu modellieren. Weiteren Untersuchungsbedarf ist hinsichtlich der Verwendung semantischer Informationen des OSM-Geodatenbestands zu sehen. So ist aktuell unklar, ob die Verfügbarkeit des vorgestellten Attributs der Straßenbreite bezogen auf deutsche Großstädte für den Verwendungszweck ausreichend ist.

Weitere Ziele bestehen darin, das Modell so anzupassen, dass im zu entwickelnden Trainingssystem Vorgehensweisen für die Absicherung von Demonstrationen modelliert werden können, z. B. das Verhindern des Weiterlaufens einer Demonstration durch eine Blockade mit anschließendem versuchten Zugriff auf spezifische Personen in der Menge durch Polizeikräfte. Die Reaktion auf dieses Szenario könnte in unterschiedlichen Verhalten resultieren:

- Die Demonstration wird gestoppt und einzelne Personen werden durch Einsatzkräfte in Gewahrsam genommen.
- Die Demonstration wird gestoppt, blockiert aber das Eindringen von Einsatzkräften in ihre Reihen. Nach einiger Zeit löst sich dieser Zustand auf und die Einsatzkräfte ziehen sich zurück und
- die Einsatzkräfte versuchen in die Menschenmenge einzudringen, woraufhin diese sich aufteilt und in diverse Richtungen flieht. Die Einsatzkräfte nehmen daraufhin die Verfolgung von einer oder mehreren flüchtenden Gruppen auf.

Die letztgenannten Aspekte zur Forschung und Entwicklung im Kontext von Trainingssystemen resultieren aus den Anforderungen des Forschungsprojekts („TARGET" EU H2020), in das die Entwicklung, die zu diesem Beitrag führte, eingebettet war.

DANKSAGUNG

Die vorgestellten Forschungsarbeiten wurden unterstützt durch das EU-Förderprogramm für Forschung und Innovation der europäischen Kommission Horizont 2020 unter der Fördernummer: n° 653350 (TARGET).

Literatur

Almer, A.; Perko, R.; Schrom-Feiertag, H.; Schnabel, T.; Paletta, L. (2016): Critical Situation Monitoring at Large Scale Events from Airborne Video Based Crowd Dynamics Analysis. In: Sarjakoski, T.; Santos, M. Y.; Sarjakoski, L. T. (Eds.): Lecture Notes in Geoinformation and Cartography. Springer, Cham, S. 351-368.

ARGEBAU (Fassung 2002, zuletzt geändert im 2012): Musterbauordnung – MBO.

ARGEBAU – Fachkommission Bauaufsicht (Fassung 2005, zuletzt geändert 2014): Musterverordnung über den Bau und Betrieb von Versammlungsstätten (Muster-Versammlungsstättenverordnung – MVStättV).

Burstedde, C.; Klauck, K.; Schadschneider, A.; Zittartz, J. (2001): Simulation of pedestrian dynamics using a two-dimensional cellular automaton. In: Physica A: Statistical Mechanics and its Applications, 295 (3-4), S. 507-525.

Chraibi, M.; Seyfried, A.; Schadschneider A. (2010): Generalized centrifugal-force model for pedestrian dynamics. In: Physical Review, E 82 (4 PT 2), S. 1-9.

Crowd-Sim (2016): Crowd-Sim. https://rawgit.com/Jacq/crowd-sim/master/demo/index.html, Zugriff 06/2018.

Curtis, S.; Best, A.; Manocha, D. (2016): Menge, A Modular Framework for Simulating Crowd Movement. In: Collective Dynamics, 1, S. 1-40.

Duckham, M.; Kulik, L.; Worboys, M.; Galton, A. (2008): Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. In: Pattern Recognition, 41 (10), S. 3224-3236.

Edelsbrunner, H.; Kirkpatrick, D.; Seidel, R. (1983): On the shape of a set of points in the plane. In: IEEE Transactions on Information Theory, 29 (4), S. 551-559.

Esri (1998): Esri Shapefile Technical Description. An Esri White Paper, July 1998. https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf, Zugriff 07/2018.

Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. (1996): A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: Proceedings 2nd International Conference Knowledge Discovery and Data Mining, S. 226-231.

Finkel, R.; Bentley, J. (1974): Quad trees a data structure for retrieval on composite keys. In: Acta Informatica, 4 (1), S. 1-9.

Fukui, M.; Ishibashi, Y. (1999): Jamming Transition in Cellular Automaton Models for Pedestrians on Passageway. In: Journal of the Physical Society of Japan, 68 (11), S. 3738-3739.

Gatto, D. (1993): The use of interactive computer simulations in training. In: Australasian Journal of Educational Technology, 9 (2), S. 144-154.

Gloor, C. (2018): PEDSIM Pedestrian Crowd Simulation. http://pedsim.silmaril.org/, Zugriff 06/2018.

Graham, R. (1972): An efficient algorithm for determining the convex hull of a finite planar set. In: Information Processing Letters, 1 (4), S. 132-133.

Helbing, D. (1990): Physikalische Modellierung des dynamischen Verhaltens von Fußgängern (Physical Modeling of the Dynamic Behavior of Pedestrians). In: SSRN Journal.

Helbing, D.; Buzna, L.; Johansson, A.; Werner, T. (2005): Self-Organized Pedestrian Crowd Dynamics, Experiments, Simulations, and Design Solutions. In: Transportation Science, 39 (1), S. 1-24.

Helbing, D.; Farkas, I.; Molnàr, P.; Vicsek, T. (2002): Simulation of pedestrian crowds in normal and evacuation situations. In: Schreckenberg, M.; Sharma, S. D. (Eds.): Pedestrian and Evacuation Dynamics. Springer, Berlin, S. 21-58.

Helbing, D.; Molnár, P. (1995): Social force model for pedestrian dynamics. In: Physical Review E, 51 (5), S. 4282-4286.

Helbing, D.; Mukerji, P. (2012): Crowd disasters as systemic failures, Analysis of the Love Parade disaster. In: EPJ Data Science, 1 (1), S. 1-40.

Henderson, L. (1974): On the fluid mechanics of human crowd motion. In: Transportation Research, 8 (6), S. 509-515.

Hoogendoorn, S.; Daamen, W. (2007): Microscopic Calibration and Validation of Pedestrian Models: Cross-Comparison of Models Using Experimental Data. In: Schadschneider, A.; Pöschel, T.; Kühne, R.; Schreckenberg, M.; Wolf, D. (Eds.): Traffic and Granular Flow'05. Springer, Berlin/Heidelberg, S. 329-340.

Horni, A.; Nagel, K.; Axhausen K. (2016): The Multi-Agent Transport Simulation MATSim. Ubiquity Press, London.

Hughes, R. (2000): The flow of large crowds of pedestrians. In: Mathematics and Computers in Simulation, 53 (4-6), S. 367-370.

Inaba, M.; Katoh, N.; Imai, H. (1994): Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering. In: Proceedings of the Tenth Annual ACM Symposium on Computational Geometry, S. 332-339.

Jain, A.; Dubes, R. (1988): Algorithms for clustering data, Prentice Hall advanced reference series. Prentice-Hall, Englewood Cliffs.

Johansson, A. (2009): Data-driven modeling of pedestrian crowds. Dissertation, Fakultät Verkehrswissenschaften, Technische Universität Dresden.

Johansson, A.; Helbing, D.; Shukla, P. (2007): Specification of the Social Force Pedestrian Model by evolutionary adjustment to Video Tracking Data. In: Advances in Complex Systems, 10 (supp02), S. 271-288.

Kemloh, U.; Chraibi, M. (2016): Jupedsim/Jupedsim: Jupedsim V0.8.1. Zenodo, Forschungszentrum Jülich GmbH.

Kielar, P.; Biedermann, D.; Borrmann, A. (2016): MomenTUMv2: A Modular, Extensible, and Generic Agent-Based Pedestrian Behavior Simulation Framework TUM-I1643, Technische Universität München.

Kirchner, A.; Klüpfel, H.; Nishinari, K.; Schadschneider, A.; Schreckenberg, M. (2003): Simulation of competitive egress behavior, Comparison with aircraft evacuation data. In: Physica A: Statistical Mechanics and its Applications, 324 (3-4), S. 689-697.

Korhonen, T.; Hostikka, S. (2009): Fire Dynamics Simulator with Evacuation: FDS+Evac. Technical Reference and User's Guide.

Kormanová, A. (2013): A Review on Macroscopic Pedestrian Flow Modelling. In: Acta Infor-

matica Pragensia, 2 (2), S. 39-50.

Krajzewicz, D.; Erdmann, J.; Behrisch, M.; Bieker, L. (2012): Recent Development and Applications of SUMO – Simulation of Urban MObility. In: International Journal On Advances in Systems and Measurements, 5 (3&4), S. 128-138.

Kretz, T. (2007): Pedestrian Traffic. Simulation and Experiments. Dissertation, Fakultät für Physik, Universität Duisburg-Essen.

Kuligowski, E.; Peacock, R.; Hoskins, B. (2010): A Review of Building Evacuation Models. 2nd Edition. US Deptartment of Commerce, National Insitute of Standards and Technology.

MacQueen, J. (1967): Some Methods for Classification and Analysis of Multivariate Observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical, Statistics and Probability, Vol. 1, S. 281-296.

Meinert, A.; Hahmann, S.; Kopf, S.; Brausewetter, P. (2018): Simulation von Menschenmengen im urbanen Umfeld. In: 38. Wissenschaftlich-Technische Jahrestagung der DGPF und PFGK18 Tagung in München, Publikationen der DGPF, Band 27.

Meyer-König, T.; Waldau, N.; Klüfel, H. (2007): The RiMEA Project – Development of a new Regulation. In: Waldau, N.; Gattermann, P.; Knoflacher, H.; Schreckenberg, M. (Eds.): Pedestrian and Evacuation Dynamics 2005. Springer, Berlin/Heidelberg, S. 309-313.

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. In: Proceedings of the Royal Society, 276 (1668), S. 2755-2762.

Moussaïd, M.; Perozo, N.; Garnier, S.; Helbing, D.; Theraulaz, G. (2010): The walking behaviour of pedestrian social groups and its impact on crowd dynamics. In: PloS one, 5 (4), S. e10047.

Murtagh, F.; Contreras, P. (2012): Algorithms for hierarchical clustering, An overview. In: WIREs Data Mining Knowl Discovery (Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery), 2 (1), S. 86-97.

North, M.; Collier, N.; Ozik, J.; Tatara, E.; Macal, C.; Bragen, M.; Sydelko, P. (2013): Complex adaptive systems modeling with Repast Simphony. In: Complex Adaptive Systems Modeling, 1 (1), S. 3.

Okazaki, S.; Matsushita, S. (1993): A study of simulation model for pedestrian movement with evacuation and queuing. In: Proceedings International Conference Engineering for Crowd Safety, S. 271-280.

Open Geospatial Consortium (2015): Geographic information – Well-known text representation of coordinate reference systems. http://docs.opengeospatial.org/is/12-063r5/12-063r5.html, Zugriff 07/2018.

Orth, M. (2009): Simulation von Personenströmen, Virtuelle Evakuierung. In: PROTECTOR, 12, S. 18-19.

Predtechenskii, W.; Milinskii, A. (1978): Planning for Foot Traffic Flow in Buildings. Amerind, New Delhi.

Preparata, F.; Shamos, M. (1985): Convex Hulls: Basic Algorithms. In: Preparata, F.; Shamos, M. (Eds.): Computational Geometry. Springer, New York, S. 95-149.

Ramm, F.; Topf, J. (2010): OpenStreetMap. Die freie Weltkarte nutzen und mitgestalten. Lehmanns Media, Berlin.

Samet, H. (2006): Foundations of multidimensional and metric data structures. The Morgan Kaufmann series in data management systems. Elsevier/Morgan Kaufmann, Amsterdam.

Schadschneider, A.; Klingsch, W.; Klüpfel, H.; Kretz, T.; Rogsch, C.; Seyfried, A. (2009): Evacuation Dynamics: Empirical Results, Modeling and Applications. In: Meyers, R. A. (Hrsg.): Encyclopedia of complexity and systems science. Springer, New York, S. 3142-3176.

Schubert, E.; Sander, J.; Ester, M.; Kriegel, H.; Xu, X. (2017): DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. In: ACM Transactions on Database Systems, 42 (3), S. 1-21.

Seitz, M. (2016): Simulating pedestrian dynamics: Towards natural locomotion and psychological decision making. Dissertation, Fakultät für Informatik, Technische Universität München.

Sibson, R. (1973): SLINK, An optimally efficient algorithm for the single-link cluster method. In: The Computer Journal, 16 (1), S. 30-34.

Tan, P.-N.; Steinbach, M.; Karpatne, A.; Kumar, V. (2013): Introduction to data mining. Addison-Wesley, Boston.

TIOBE Software B. V. (2018): TIOBE Index for July 2018. https://www.tiobe.com/tiobe-index/, Zugriff 07/2018.

Toffoli, T.; Margolus, N. (1991): Cellular automata machines. A new environment for modeling. MIT Press, Cambridge.

Togawa, K. (1955): Study on fire escapes basing on the observation of multitude currents. In: Report of the Building Research Institute, 14.

Wagner, C. (1997): A Navier-Stokes-like traffic model. In: Physica A: Statistical Mechanics and its Applications, 245 (1-2), S. 124-138.

Yu, W.; Chen, R.; Dong, L.; Dai, S. (2005): Centrifugal force model for pedestrian dynamics. In: Physical Review E, 72 (2 Pt 2), S. 26112.

Yuen, A.; Kay, R. (2009): Applications of cellular automata. http://www.cs.bham.ac.uk/~rjh/courses/NatureInspiredDesign/2009-10/StudentWork/Group2/design-report.pdf, Zugriff 08/2018.