Zellulärer Zustandsautomat

3. Projekt zu Modellierung und Simulation

Daniel Graf, Dimitrie Diez, Arne Schöntag, Peter Müller

Inhaltsverzeichnis

1 Einführung

Im Zuge der ersten Studienarbeit wurde das Laufverhalten von Probanden in der Ebene und auf der Treppe untersucht. Basierend auf den Erkenntnissen bezüglich der individuellen Wunschgeschwindigkeiten werden in dieser Studienarbeit Personenbewegungen in der Ebene simuliert. Mit Hilfe solcher Simulationen können, beispielsweise im Zuge von Gebäudeplanungen unterschiedliche Gebäude- und Raumgestaltungen simuliert und hinsichtlich schnellster Räumungen im Krisenfall optimiert werden.

2 Beschreibung des Modells

Für die Simulation der Personenbewegungen wird ein zellulärer Zustandsautomat implementiert. Jede Zelle kann entweder leer oder durch eine Person oder ein Hindernis besetzt sein. Sie kann außerdem ein Ziel oder eine Quelle enthalten. Ein Hindernis kann von keiner Person betreten werden. Die Personen versuchen im Laufe der Simulation von der Quelle (Startposition) zum Ziel zu gelangen. Hierbei können sie sich in der sog. Moore-Umgebung bewegen. Es kann jede Zelle betreten werden, die frei ist und eine Nachbarzelle der aktuellen Zelle ist. Als Nachbarzelle wird jede Zelle bezeichnet, welche mit der aktuellen Zelle eine Kante oder Ecke teilt.
Die Ziele werden als attraktiv für die Personen angesehen. Die Personen steigern ihren persönlichen Nutzen, je näher sie dem Ziel kommen. In der Realität fühlt sich eine Person jedoch unwohl, wenn ihr eine fremde Person zu nahe kommt. Im Modell wird dies dadurch berücksichtigt, dass sich der Nutzen einer Person verringert, wenn sie einer anderen zu nahe kommt. Um die Simulation möglichst realistisch zu gestalten, bewegen sich die Personen mit einer individuellen Wunschgeschwindigkeit. Diese wird aus den Ergebnissen der ersten Studienarbeit ermittelt.

Das erläuterte Modell gibt die einzelnen Personenbewegungen aus und visualisiert diese zusätzlich. Die Bewegung der Personen wird in drei getrennten Simulationen mit jeweils unterschiedlichen Algorithmen berechnet. Zunächst wird für jedes Feld der negative euklidische Abstand als Zielnutzen berechnet. Jede Person versucht durch die Wahl des umliegenden Feldes mit dem maximalen Wert seinen Nutzen zu steigern.
Darüber hinaus besteht die Möglichkeit die Personenbewegung mit Hilfe des Floor-Flooding, basierend auf dem Dijkstra Algorithmus zu ermitteln. Jede Person wählt von den anliegenden Feldern das Feld, welches die geringste Anzahl an notwendigen Bewegungen bis zum Ziel hat.
Als dritte Variante erfolgt die Ermittlung der Personenbewegung ebenfalls mit Hilfe des Floor-Flooding. Als Grundlage dient hierbei jedoch die Lösung der Eikonal-Gleichung,

welche die Ausbreitung einer Welle beschreibt. Hierfür wird der Fast-Marching Algorithmus verwendet.
Im Laufe der Studienarbeit werden die beschriebenen Modelle anhand unterschiedlicher Tests verglichen und ausgewertet. Eine detaillierte Erläuterung der Tests ist im folgenden Kapitel 3.13 beschrieben.

3 Anforderungen/Requirements

3.1 Allgemeines

Es soll ein zellulärer Zustandsautomat implementiert werden. Die Software soll Personenbewegungen in der Ebene simulieren. Die unterschiedlichen Szenarien (Zustand der Zellen zu Beginn der Simulation), wie beispielsweise die Position der Hindernisse werden der Software übergeben. Diese sind im Unterkapitel 3.13 näher erläutert. Es wird davon ausgegangen, dass sich zu Beginn der Simulation alle Personen im Startbereich (es soll möglich sein mehrere Startbereiche anzulegen) befinden.

3.2 Zellen

Das gesamte Feld (zweidimensionaler Bereich indem sich die Personen bewegen können) soll in Zellen eingeteilt werden. Die Größe des Feldes und die Anzahl an Zellen sollen der Software als Parameter übergeben werden.
Jede Zelle soll quadratisch sein. Jede Zelle kann verschiedene Zustände haben. Sie kann entweder leer oder besetzt sein. Ist eine Zelle besetzt, wird noch weiter unterschieden, ob sie durch ein Hindernis, das Ziel- (Personensenke) oder Startfeld (Personenquelle) oder durch eine Person besetzt ist.

3.3 Hindernisse

Es soll davon ausgegangen werden, dass jedes Hindernis eine oder mehrere Zellen ganz belegt. Hindernisse können von den Personen nicht betreten werden. Falls sich ein Hindernis zwischen einer Person und ihrem Ziel befindet, ist die Person gezwungen um das Hindernis herum zu gehen.
In der Realität versucht eine Person einem Hindernis möglichst frühzeitig auszuweichen. Sie senkt somit ihren Nutzen, je näher sie einem Hindernis kommt. Dies wird im weiteren Verlauf dieser Studienarbeit nicht berücksichtigt.

3.4 Ziele

Durch das entsprechende Testszenario werden ein oder mehrere Ziele definiert. Ziele werden als attraktiv für die Personen angesehen. Jede Person erhöht ihren Nutzen, wenn sie sich dem Ziel nähert. Befinden sich mehrere Ziele im Feld, wird die Wahl des Ziels für jede Person abhängig vom Betriebsmodus bestimmt.

3.5 Personen

Jede Person soll in ihrem Startfeld starten und sich im Laufe der Simulation von Zelle zu Zelle auf ihr Ziel hin bewegen und dadurch ihren persönlichen Nutzen, je näher sie dem Ziel kommt, steigern. Jede Person kann sich in der Moore-Umgebung (8 mögliche Bewegungsrichtungen) bewegen. Die möglichen Bewegungsrichtungen sind in Abbildung 1 rot markiert.

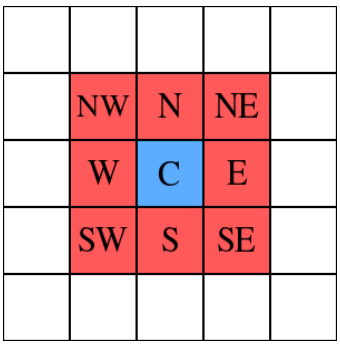

Abbildung 1: Moore-Umgebung

Die Personen können nur freie Zellen betreten. Jede Person wählt genau die Zelle als nächste Zelle, welche ihr den steilsten Nutzenanstieg ermöglicht. Der Nutzen der verschiedenen Zellen wird abhängig vom Betriebsmodus unterschiedlich berechnet. Darüber hinaus soll der Nutzen durch die Position der anderen Personen beeinflusst werden.

Der Nutzen einer Person wird verringern, wenn sie anderen Personen zu nahe kommt. In der Simulation soll folgende Funktion verwendet werden.

$$u_p(d) = \begin{cases} -h \exp(\frac{1}{(\frac{d}{w})^2 - 1}) & falls\, |d| < w \\ 0 & sonst \end{cases}$$

Hierbei wird mit d der Abstand zur nächsten Person bezeichnet. w und h stehen für die Stärke und Reichweite der Funktion.
Der Nutzen einer Nachbarzelle setzt sich somit insgesamt aus der Addition des Zielnutzens

der Zelle (abhängig vom Algorithmus, vgl. Unterkapitel 3.11) mit dem Nutzeneinfluss aller Personen (in der Umgebung) auf die entsprechende Zelle zusammen.

3.6 Geschwindigkeit der Personen

Jede Person soll eine individuelle Wunschgeschwindigkeit besitzen. Sie bewegt sich bei freier Bahn mit dieser Geschwindigkeit fort. Die Wunschgeschwindigkeiten sollen als normalverteilt angesehen werden. Als Mittelwert und Varianz sollen die errechneten Werte aus der ersten Studienarbeit verwendet werden.

3.7 Zeitmodellierung

Die Simulation unterliegt keinen harten Echtzeitanforderungen. Sie soll für die gesamte Simmulationsdauer so schnell wie möglich durchgeführt werden. Es soll keine Globale Uhr modelliert werden.
Die Visualisierung der Simulation soll hingegen die Bewegungen der Personen in Echtzeit darstellen. Sie unterliegt somit Echtzeitanforderungen.

3.8 Parameter

Die Tabelle 1 zeigt alle möglichen Parameter, welche die Software steuern. Alle Parameter sind optional.

Parameter	Beschreibung	Default
–free-flow-velocity	Mittelwert der Wunschgeschwindigkeit [m/s]	1,0
–free-flow-deviation	Varianz der Wunschgeschwindigkeiten [m/s]	0
–algorithm	Algorithmus (1, 2 oder 3)	2
–output-folder	Ordner in den der Output gespeichert wird	../data/
–cellsize	Größe der einzelnen Zellen [m]	1
–input-map	Auswahl des Testfalls	-

Tabelle 1: Parameter und Defaultwerte

3.9 Berechnungen

Die Aufgabe der Software ist die Simulation von Personenbewegungen. Für möglichst genaue Zeitangaben wird die Java Klasse BigDecimal verwendet. Für die Division werden 32 Nachkommastellen berücksichtigt und als Rundungsmodus wird Half_Even verwendet, da dieser kumulative Fehler bei sich wiederholenden Berechnungen minimiert. Die anschließende Berechnung und Aufbereitung der Daten (insebesondere für den Test 3.13.5) erfolgt mit Mathematica.

3.10 Report

Alle errechneten Ergebnisse sollen über einen Report ausgeleitet werden. Der Report soll im .xml Format ausgeleitet werden um eine gute Weiterverarbeitung der Ergebnisse (Visualisierung) zu gewährleisten. Die Datei *output.xml* soll alle benötigten Informationen, wie beispielsweise die einzelnen Events der Personen, die durch den Algorithmus berechneten Entfernungen der einzelnen Zellen vom Ziel und eine Übersicht über den Zustand der einzelnen Zellen (Fieldmap), für Simulation beinhalten.
Darüber hinaus soll während der Visualisierung der Simulation die Möglichkeit bestehen interessante Aspekte mit Hilfe von Screenshots festzuhalten. Diese sollen im .png Format gespeichert werden.

3.11 Berechung des Zielnutzens

Die Software hat beinhaltet 3 unterschiedliche Algorithmen für die Berechnung des Zielnutzens (Attraktivität) einer Zelle.

Algorithmus 1: Der Nutzen bzw. die Attraktivität jeder Zelle entspricht dem negativen euklidischen Abstand. Hierbei können keine Hindernisse umgangen werden. Jede Person versucht den Nutzen zu maximieren und wählt jeweils die Zelle mit maximalen Wert aus den umliegenden aus.

Algorithmus 2: Der Nutzen bzw. die Attraktivität jeder Zelle wird durch Floor-Flooding, basierend auf dem Dijkstra Algorithmus, berechnet. Für jede Zelle wird die geringste Anzahl an notwendigen Bewegungen bis zum Ziel ermittelt.

Algorithmus 3: Analog zum Algorithmus 2 wird auch in diesem hier der Nutzen bzw. die Attraktivität jeder Zelle mittels Floor-Flooding berechnet. Es wird jedoch der Fast-Marching Alforithmus (Lösung der Eikonal-Gleichung) als Grundlage verwendet.

3.12 Werte für die Auswertung

Alle in der Tabelle 2 aufgeführten Werte sollen auf Basis der Simulation ermittelt und ausgewertet werden. Einige Werte werden nur für bestimmte Tests benötigt.

3.13 Testszenarien für die Simulation

Die implementierten Algorithmen (vgl. Unterkapitel 3.11) sollen anhand unterschiedlicher Testszenarien verglichen und verifiziert werden.

Ermittelte Werte	Beschreibung	Wann benötigt
mittlere Geschwindigkeit	durchschnittliche Geschwindigkeit der Personen	Rimea-Test 4
Fluss (Personen/Meter/s)	Anzahl der Personen mit Geschwindigkeit > 1m/s	Rimea-Test 4
Simulationsdauer	Dauer bis der Test beendet ist	Hühner Test, Evakuierung (2 Türen), Evakuierung (4 Türen)

Tabelle 2: Werte für die Auswertung

3.13.1 Freier Fluss

Im Zuge dieses Tests soll überprüft werden, ob sich Personen bei freier Bahn (keine Hindernisse zwischen Start und Ziel) mit ihrer jeweiligen Wunschgeschwindigkeit auf das Ziel hin bewegen. Des weiteren soll überprüft werden, ob die Person den kürzesten Weg verwendet. Der Test soll mit allen 3 Betriebsmodi durchgeführt werden. Abbildung 2 stellt den Testaufbau schematisch dar.

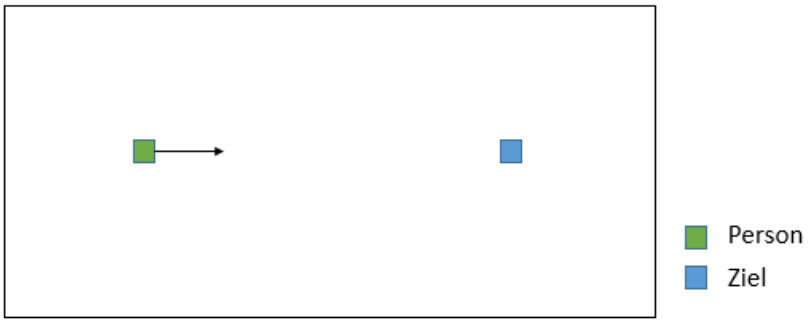

Abbildung 2: Testfall Freier Fluss (schematisch)

3.13.2 Hühnertest

Zwischen Start und Ziel soll ein U-förmiges Hindernis (Öffnung in Richtung des Startes) eingefügt werden. Abhängig vom Betriebsmodus wird ein unterschiedliches Ergebnis erwartet:
Betriebsmodus 1: Die Personen sollen sich auf dem kürzesten Weg auf das Ziel zubewegen. Sobald sie das Hindernis erreicht haben, sollen sie stehen bleiben. Ein erfolgreiches absolvieren des Tests wird nicht erwartet.
Betriebsmodi 2 und 3: Die Personen sollen das Hindernis umgehen und das Ziel erreichen. Bei einem „Feststecken im Hindernis“(wie im Betriebsmodus 1 gefordert) wird der Test als fehlgeschlagen gewertet. Abbildung 3 stellt den Testaufbau schematisch dar.

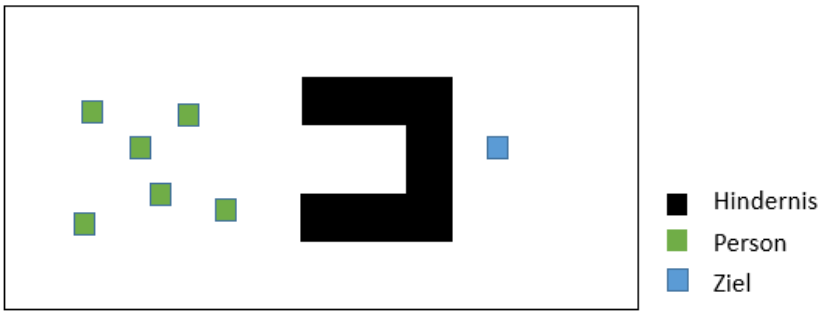

Abbildung 3: Testfall Hühnertest (schematisch)

3.13.3 Evakuierung eines Raumes mit 2 Türen

Die Personen befinden sich zu Beginn in einen von Hindernissen umgebenen, quadratischen Raum. An 2 Seiten sind Öffnungen platziert (freie Zellen). Hinter diesen Öffnungen befindet sich jeweils ein Ziel. Im Laufe des Tests sollen die Personen durch die beiden Engstellen zum Ziel gelangen.
Dieser Test soll mit den Betriebsmodi 2 und 3 in unterschiedlichen Versionen durchgeführt werden. Zunächst sollen sich die Türen an benachbarten Seiten befinden. Hierbei sollen sie einmal mit geringem und einmal mit sehr großem Abstand zueinander plaziert werden. Anschließend sollen die Türen an gegenüberliegenden Seiten plaziert mittig werden. Unterschiede hinsichtlich der Evakuierungsdauer sind festzuhalten. Die Abbildungen 4, 5 und 6 zeigen die unterschiedlichen Testversionen schematisch.

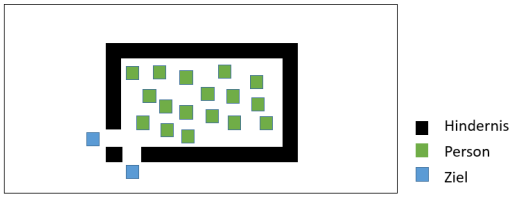

Abbildung 4: Testfall Evakuierung eines Raumes mit 2 Türen (Nebeneinander, geringer Abstand)

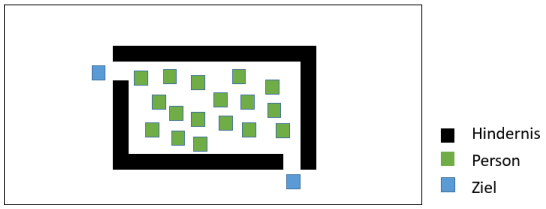

Abbildung 5: Testfall Evakuierung eines Raumes mit 2 Türen (Nebeneinander, großer Abstand)

3.13.4 Evakuierung eines Raumes mit 4 Türen

Dieser Testfall entspricht der in Kapitel 3.13.3 beschriebenen Evakuierung mit 2 Türen. In diesem Fall befindet sich jedoch an jeder Raumseite eine Tür. Die Türen sollen sich

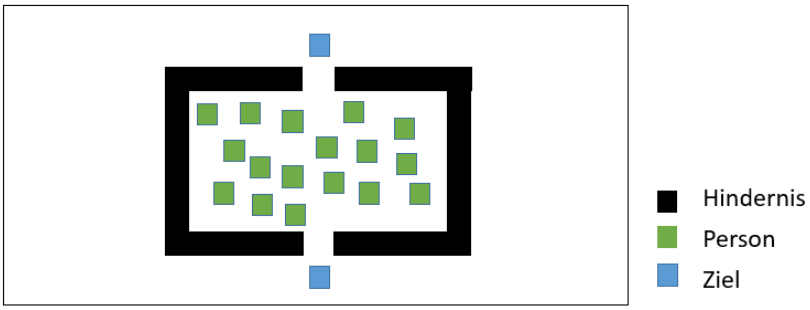

Abbildung 6: Testfall Evakuierung eines Raumes mit 2 Türen (Gegenüber)

in einer ersten Version mittig bezogen auf die jeweilige Raumseite befinden. Darüber hinaus sollen in einer weiteren Version die Auswirkungen ermittelt werden, wenn jeweils die 2 Türen an benachbarten Seiten mit minimalen Abstand zueinander plaziert werden. Unterschiede hinsichtlich der Evakuierungsdauer sind festzuhalten. Aufgrund der Analogie zur Evakuierung mit 2 Türen wird auf eine graphische Veranschaulichung des Testaufbaus verzichtet.

3.13.5 Fundamentaldiagramm - Rimea-Test 4

In diesem Testfall soll ein $65m$ langer und $12m$ breiter Gang angelegt werden. Auf der einen Seite des Ganges befindet sich die Personenquelle, auf der anderen Seite die Personensenke. Im Laufe des Testes sollen die Personen den Gang entlangschreiben. Die Anzahl der Personen ist zu variieren. In einer weiteren Version des Tests sollen Personen, welche die Personensenke erreicht haben, erneut bei der Personenquelle starten (Endlosschleife). Abbildung 7 zeigt den Testaufbau schematisch.

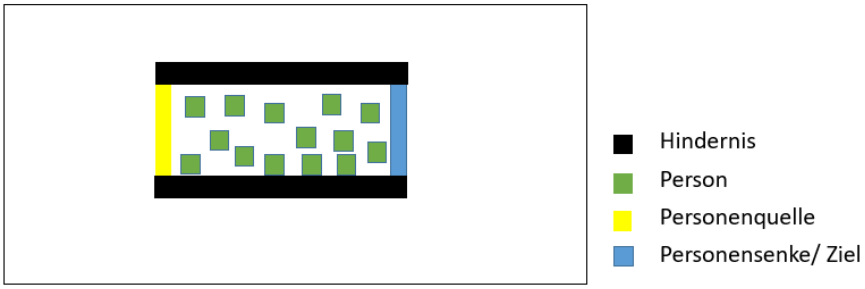

Abbildung 7: Testfall Rimea 4 (schematisch)

3.13.6 Engstelle (unidirektional)

In diesem Testfall soll, analog zum Rimea-Test 4, ein $65m$ langer und $12m$ breiter Gang angelegt werden. Am Ende des Ganges befindet sich eine Engstelle mit $6m$ Breite und dahinter die Personensenke. Es soll mit variierender Anzahl an Personen getestet werden, wie lange es dauert bis die Engstelle durchschritten wurde. Der schematische Testaufbau ist in Abbildung 8 aufgeführt.

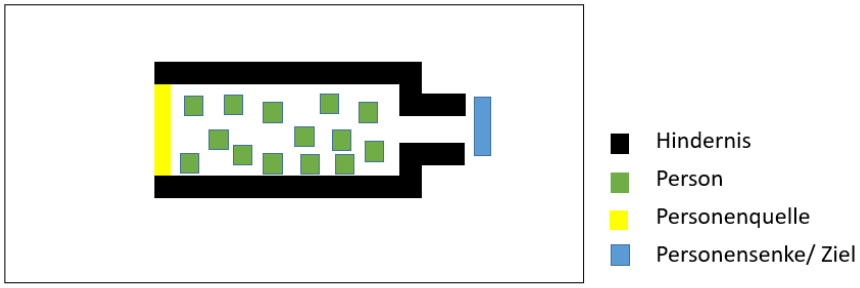

Abbildung 8: Testfall Engstelle (schematisch)

3.14 Visualisierung

Die Visualisierung soll die errechneten Personenbewegungen graphisch darstellen. Als Grundlage soll der ausgegebene Report (vgl. Abschnitt 3.10) dienen. Konkrete Anforderungen bezüglich der Visualisierung sind in Abbildung 9 abgebildet.

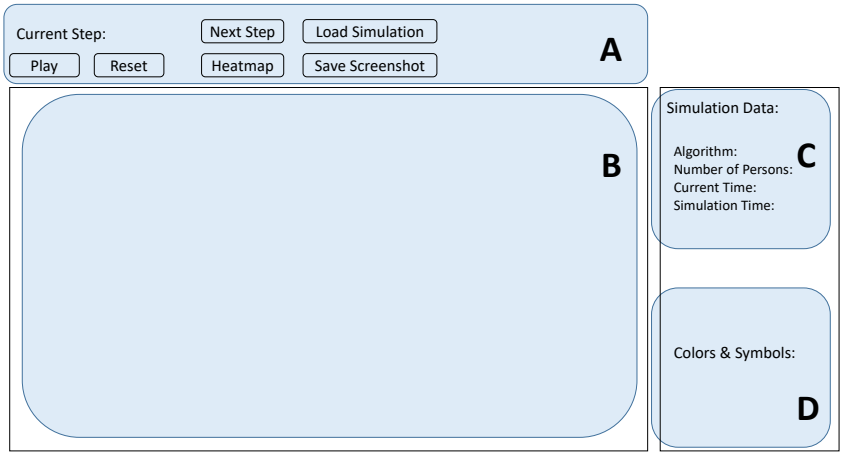

Abbildung 9: Visualisierung (schematisch)

Die Oberfläche der Visualisierung soll aus 4 verschiedenen Teilen bestehen, welche in der Abbildung durch die blauen Kästen A-D hervorgehoben wurden. Im Teil A sollen Konfigurationsmöglichkeiten für die Visualisierung aufgeführt sein. Beispielsweise soll zwischen einer normalen Darstellung der Simulation und einer Heatmap gewechselt werden können. Darüber hinaus soll die Möglichkeit bestehen, die Visualisierung zu stoppen und interessante Aspekte mit Hilfe von Screenshots festzuhalten.
Teil B enthält die eigentliche Visualisierung der Personen, Hindernisse und Ziele. In Teil C sind alle Informationen bezüglich der Visualisierung aufgeführt und Teil D enthält eine Legende der verwendeten Symbole und Farben.

4 Softwaredesign

Der Aufbau der Anwendung wurde im Team diskutiert und anschließend mittels UML spezifiziert.

5 Softwaretest

5.1 Verifikation

5.2 Validation

6 Ausblick und Fazit