

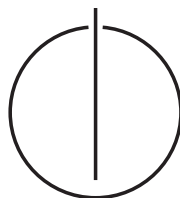
TECHNISCHE UNIVERSITÄT MÜNCHEN

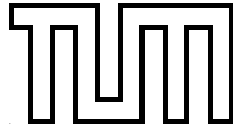
FAKULTÄT FÜR INFORMATIK

Bachelorarbeit in Informatik

Kombinierte Modellierung von Fußgänger- und Kraftfahrzeugverkehr in vereinfachten Situationen

Johannes Peter Zarnitz





TECHNISCHE UNIVERSITÄT MÜNCHEN

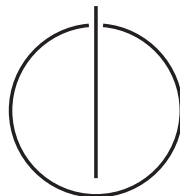
FAKULTÄT FÜR INFORMATIK

Bachelorarbeit in Informatik

Kombinierte Modellierung von Fußgänger- und Kraftfahrzeugverkehr in vereinfachten Situationen

Combined modeling of pedestrian and vehicular traffic in simplified situations

Bearbeiter:	Johannes Peter Zarnitz
Aufgabensteller:	Prof. Dr. Hans-Joachim Bungartz, Prof. Dr. Gerta Köster
Betreuer:	Felix Dietrich
Tag der Abgabe:	13. März 2015



Ehrenwörtliche Erklärung

Ich versichere, dass ich diese Bachelor's Thesis selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

Ort, Datum

Unterschrift

Zusammenfassung

Durch den immensen Verkehr in Großstädten wird es für Stadtplanungen, Versicherungen oder auch Veranstalter von großen Events immer interessanter mögliche Eingriffe in den Straßenverkehr oder den Umgang mit großen Menschenmengen vorab zu simulieren. Solche Simulationen müssen dann auch die verschiedenen Verkehrsteilnehmer entsprechend berücksichtigen. Um komplexen Situationen besser gerecht zu werden, beschäftigt sich diese Arbeit damit, die Personensimulation mit der Kraftfahrzeugsimulation zu kombinieren. So soll beispielsweise bei Großveranstaltungen in Städten die Verkehrslage besser eingeschätzt und bei der Planung berücksichtigt werden. Dafür werden ausgewählte, mikroskopische Modelle der Personen- und Kraftfahrzeugsimulation verglichen und am Beispiel eines Fußgängerüberweges in einer Simulationssoftware integriert.

Abstract

Due to the huge traffic in cities it is very interesting for city planning, insurance companies or the organization of big events to simulate possible interventions in the road or to deal with large crowds in advance. These simulations have to take the different road users into account. To respond better to complex situations, this thesis is dealing with combining pedestrian simulation with vehicle simulation. With a prediction based on this these simulations it should be possible to take the traffic situation at big events better into account and integrate this information in the organization of the event. To achieve that, different microscopic car and pedestrian models are compared and integrated into a simulation software to simulate pedestrians crossing a road populated with cars.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Problemstellung und Motivation	1
1.2	Lösungsansatz	1
1.3	Softwareplattform VADERE	2
2	Grundlegende Modelle	5
2.1	Fußgängersimulation	5
2.1.1	Zellulärer Automat	5
2.1.2	Social Force Model	6
2.1.3	Optimal Steps Model	7
2.1.4	Gradient Navigation Model	8
2.2	Kraftfahrzeugsimulation	8
2.2.1	Optimal Velocity Model	8
2.2.2	Nagel-Schreckenberg-Modell	9
2.2.3	Herman's Car Following Model	10
2.2.4	Driving by Visual Angle Model	11
2.2.5	AIMSUN	12
2.3	Auswahl der Modelle	13
2.4	Einführung eines bimodalen Simulationsmodus	13
3	Implementierung der Fahrzeugsimulation	15
3.1	Fahrzeug als Spezialisierung des Fußgängers	15
3.2	Implementierung des Optimal Velocity Model	16
3.3	Allgemeine Anpassung der Software	20
3.4	Implementierung des bimodalen Simulationsmodus	20
3.4.1	Trennung der Fußgänger- und Fahrzeugklassen	20
3.4.2	Umsetzung des bimodalen Simulationsmodus	21
3.4.3	Integration der Fußgängererkennung in das Optimal Velocity Model	22
4	Modellierung einer Querungsanlage	23
4.1	Fußgängerüberweg	23
4.2	Fußgängerfurt mit Lichtsignal	25
4.3	Fußgängerfurt ohne Lichtsignal	25
5	Fazit	27
5.1	Nutzen und bisheriger Stand	27
5.2	Ausblick	27

1 Einleitung

1.1 Problemstellung und Motivation

Obwohl die Beförderungsleistung des motorisierten Individualverkehrs in den letzten Jahren in Deutschland gesunken ist [Hü13], ist das Verkehrsaufkommen in vielen deutschen Großstädten immens. Sowohl der Straßenverkehr als auch der öffentliche Personennahverkehr sind durch die Straßen- und Schienennetze stark an die vorhandenen Gegebenheiten gebunden. Dazu kommt in Städten eine große Anzahl von Fußgängern und Fahrradfahrern, die ebenfalls am Straßenverkehr teilnehmen. Je nach Land gelten unterschiedliche Regelungen wie diese Teilnahme am Straßenverkehr auszusehen hat.

Im großstädtischen Alltag erlebt man immer wieder interessante Phänomene im Personen- und Kraftfahrzeugfluss. Durch bestimmte Ampel- und Abbiege-Konstellationen, durch Bushaltestellen oder durch viele weitere Faktoren wird jeder Teilnehmer am Straßenverkehr beeinflusst. Diese komplexen Situationen formal zu beschreiben ist keine einfache Aufgabe. Um jedoch besser vorhersagen zu können, wie sich Veränderungen im üblichen Verkehrsfluss auf die gesamte Verkehrssituation auswirken, ist die Forschung in diesem Bereich sehr interessant für die Entwicklung von Städten. Daher wird über modellbasierte Simulationen versucht das reale Geschehen abzubilden. So kann eine Veränderung zuerst in der Simulation getestet werden, bevor sie in der Realität umgesetzt wird. Das Ziel dabei ist es dann, sowohl Kosten zu sparen, als auch mögliche Gefahrensituationen frühzeitig zu erkennen und zu vermeiden.

Um die vollständige Verkehrssituation so realitätsnah wie nötig abbilden zu können, ist es erforderlich unterschiedliche Teilnehmer im Straßenverkehr gemeinsam und mit ihren Interaktionen zu betrachten. Für Fahrzeuge und Fußgänger gibt es jeweils bereits viele Modelle, die zur Simulation herangezogen werden können. In dieser Arbeit sollen die Bereiche der Fahrzeug- und Fußgängersimulation an einer spezifischen Situation, einem Fußgängerüberweg, zusammengeführt werden. Dieses Szenario wurde gewählt, da es im Straßenverkehr Interaktionen erfordert, die nicht durch äußere Einflüsse (wie z.B. Ampeln) vorgegeben sind, sondern individuell durch bestimmte Konstellationen entstehen.

Die gewünschte multimodale Betrachtung beschränkt sich hier auf die zwei Modi Fußgänger und Fahrzeuge. In einem ganzheitlichen, multi-modalen Szenario müssten auch weitere Akteure wie Fahrradfahrer oder der öffentliche Personennahverkehr berücksichtigt werden.

1.2 Lösungsansatz

Die vorhandenen Modelle der Fußgänger- und Fahrzeugsimulation sollen in einem gemeinsamen Szenario interagieren. Dieses soll die Möglichkeit bieten unterschiedliche Fußgänger- und Fahrzeugmodelle auszuwählen und in gemeinsamen Szenarien mögliche Verkehrssituationen zu simulieren. Durch die kombinierte Simulation können komplexere Szenarien aus dem realen Straßenverkehr vor der Umsetzung modelliert werden. Es soll möglich sein un-

terschiedliche Modelle im kombinierten Modus zu verwenden, um die Vor- und Nachteile der jeweiligen Modelle inklusive Interaktion testen zu können.

Die vorhandenen Modelle sind dann soweit zu erweitern, dass eine Interaktion während der Simulation stattfinden kann und so verschiedene Verkehrsteilnehmer aufeinander reagieren. In dieser Arbeit wird im Besonderen auf die Interaktion zwischen Fahrzeugen und Fußgängern an einem definierten Fußgängerüberweg (Zebrastreifen) eingegangen.

1.3 Softwareplattform VADERE

Die Implementierung eines Simulationsmodus mit Fahrzeugen und Fußgängern soll in der bereits vorhandenen Simulationsplattform VADERE der Forschungsgruppe von Prof. Dr. Gerta Köster an der Hochschule München umgesetzt werden. Diese Softwareplattform dient bisher der Computer-Simulation von Personenbewegungen. In interdisziplinärer Zusammenarbeit mit Soziologen, Psychologen, Veranstaltern und der Polizei werden bisher unberücksichtigte psycho-soziale Verhaltensaspekte abgebildet (vgl. [Hoc]). Die Plattform wird derzeit im Rahmen des Verbundprojekts MultikOSi entwickelt. Das Projekt beschäftigt sich mit Sicherheitskonzepten für städtische Großveranstaltungen.

Um die verschiedenen Modelle miteinander interagieren zu lassen, soll die Software VADERE um ein Fahrzeugmodell erweitert werden. Außerdem soll die Implementierung so ausgelegt werden, dass weitere Fahrzeugmodelle implementiert werden können. Das Programm bietet bereits die Möglichkeit unterschiedliche Modelle der Fußgängersimulation zu verwenden. Daher sollen verschiedene Fahrzeugmodelle hinzugefügt werden können und ein multi-modaler Simulationsmodus ergänzt werden. Dieser soll die Möglichkeit bieten, aus den implementierten Fußgänger- und Fahrzeugmodellen jeweils eines für die Simulation wählen zu können.

Über verschiedene Modelleigenschaften, Verkehrsteilnehmer-Quellen und -Ziele sowie die Erstellung einer Topographie kann die Software an die gewünschte Situation angepasst werden (siehe Abbildung 1.1 und Abbildung 1.2). Eine solche Topographie wird dann zur Betrachtung eines Fußgänger-Überweges erstellt.

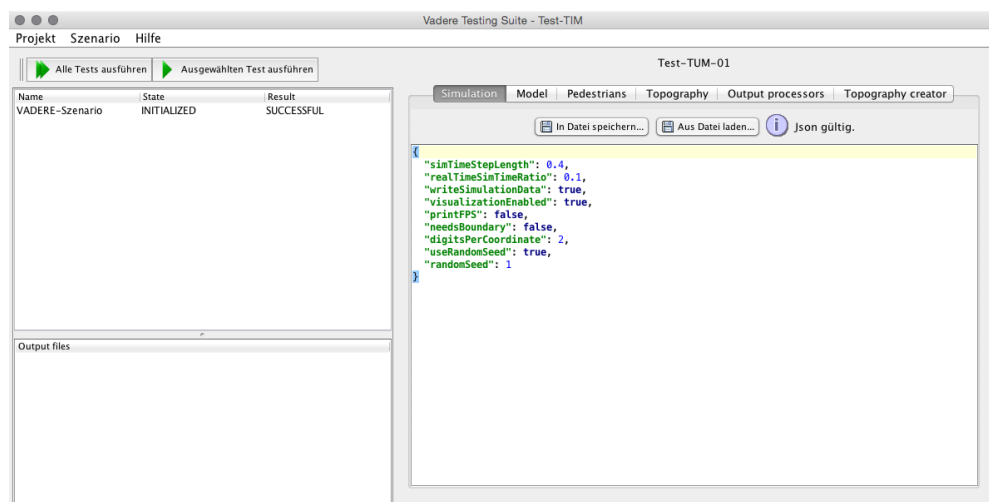


Abbildung 1.1 – VADERE-Simulationsprogramm: Bearbeitung der Simulationsparameter

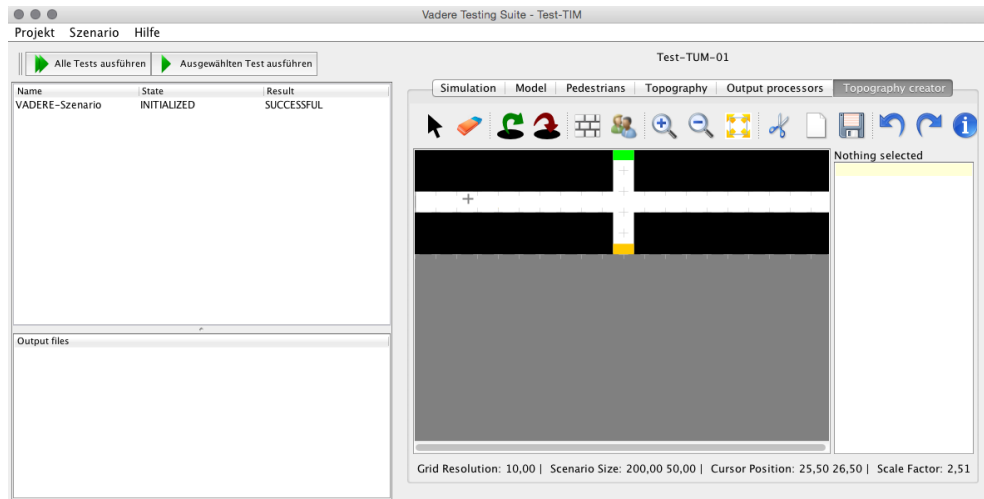


Abbildung 1.2 – VADERE-Simulationsprogramm: Bearbeitung der Topographie

2 Grundlegende Modelle

In den letzten Jahrzehnten wurden bereits einige Modelle zur mikroskopischen Simulation von Verkehrsströmen im Kraftfahrzeugverkehr formuliert und validiert. Im Gegensatz zu makroskopischen Modellen eignen sich diese sehr gut dafür, sie mit den vorhandenen mikroskopischen Modellen der Fußgängersimulation in VADERE zu kombinieren. Um eine verknüpfte Modellierung von Fahrzeug- und Fußgängersimulation zu erstellen, können bereits vorhandene mikroskopische Modelle für die einzelnen Domänen herangezogen werden. In Kapitel 2.3 wird dann basierend auf vorgegebenen Kriterien je ein Modell ausgewählt, um es beispielhaft zu implementieren und im Interaktionsmodus zu testen.

2.1 Fußgängersimulation

Im folgenden werden einige ausgewählte Modelle bzw. Modelltypen der mikroskopischen Fußgängersimulation vorgestellt, basierend auf dem Review in [DKSvS14].

1. Zellulärer Automat
2. Social Force Model
3. Optimal Steps Model
4. Gradient Navigation Model

2.1.1 Zellulärer Automat

Ein einfacher Modellansatz für die Simulation von Verkehrsteilnehmern ist die Verwendung zellulärer Automaten. Ein für die Verkehrssimulation eingesetzter Automat wurde von Kai Nagel und Michael Schreckenberg [SSNI95] entwickelt und basiert auf der Aufteilung der zur Verfügung stehenden Fläche in gleich große Einzelstücke, in denen sich jeweils ein Verkehrsteilnehmer (hier: Fahrzeug) aufhalten kann. Durch Transitionen nach vorgegebenen Regeln innerhalb des zellulären Automaten, der auf diesem Raster aufgebaut ist, wird die Fortbewegung der einzelnen Akteure dargestellt. Während in 2.2.2 auf die Verwendung des Ansatzes zur Fahrzeugsimulation eingegangen wird, soll hier die Weiterentwicklung des Modells in [BKSZ01] zur Simulation von Fußgängern betrachtet werden. Ein Fußgänger kann hier in eine Zelle wechseln, die sich in der Moore-Nachbarschaft der aktuellen Position befindet (siehe Abbildung 2.1).

Durch das vorgegebene Raster, das verschiedene geometrische Formen als Grundelemente verwenden kann, ist die Bewegungsfreiheit der einzelnen Fußgänger stark eingeschränkt. Allerdings kann es durch die räumliche Trennung keine Probleme mit Überlappungen geben.

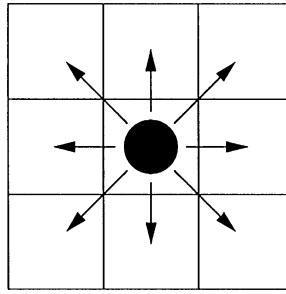


Abbildung 2.1 – Mögliche Bewegungen eines Fußgängers in einem Zeitschritt bei einem Zellulären Automaten mit Moore-Nachbarschaft nach [BKSZ01]

Vorteile ergeben sich durch schnelle Berechnung, Einfachheit und durch die intuitive Einbettung von Fußgänger-Verhalten [DKSvS14]. In Arbeiten wie beispielsweise [ZJWC12] wurde die ursprüngliche Idee weiterentwickelt, sodass die Zellen (bzw. Akteure) mit zusätzlichen Feld- und Personeneigenschaften den zurückzulegenden Weg für den Fußgänger optimieren. Das Modell geht dabei noch von rein rationalen Entscheidungen des Fußgängers aus, was in der Realität allerdings nicht zwangsläufig der Fall ist.

2.1.2 Social Force Model

Die grundlegende Idee, die Dirk Helbing und Péter Molnár 1995 zur Beschreibung des Social Force Model [HM95] verwendet haben, beruht darauf, dass ein Fußgänger von verschiedenen Gegebenheiten seiner Umgebung derart in seiner Bewegung beeinflusst wird, als würde diese von externen Kräften bestimmt. Dabei wird folgender Ablauf für die Entscheidungsfindung zur Bewegungsänderung zu Grunde gelegt:

- Äußere Reize:
 - Aufnahme der Umgebung: Der Fußgänger nimmt die gesamte Situation in seiner Umgebung auf, um aufgrund von Erfahrungswerten handeln zu können.
 - Persönliche Ziele und Interessen: Jeder Fußgänger verfolgt eigene Ziele. Diese sind beispielsweise eine Reihe geplanter Zielorte sowie Abstände zu anderen Fußgängern oder Hindernissen.
- Physiologische und Mentale Prozesse:
 - Verarbeitung der Information: Nach der Aufnahme der äußeren Reize verarbeitet der Fußgänger diese mit dem Ziel seine eigene Bewegung zu optimieren.
 - Entscheidung: Auf dieser Grundlage trifft er dann eine Entscheidung, wie er seine Bewegung zu verändern hat.
 - Motivation zur Bewegungsänderung: Die genaue Bewegungsänderung wird angestrebt.
- Reaktion
 - Physikalische Umsetzung: Die geplante Bewegungsänderung wird umgesetzt.

Über diverse Differentialgleichungen ist dieses Konzept mathematisch formuliert. So werden die gewünschte Richtung sowie die daraus resultierende Beschleunigung mit Hilfe von

Vektoren bestimmt. Hinzu kommen abstoßende Effekte zwischen einzelnen Fußgängern sowie zu vorhandenen Hindernissen. Es ist auch möglich, dass andere Fußgänger - wie beispielsweise Freunde oder Straßenkünstler - eine anziehende Wirkung auf die eigene Bewegung haben. Dabei ist zu beachten, dass Situationen, die sich nicht im Blickwinkel des Fußgängers befinden, keine besondere Rolle bei der Reaktion auf äußere Einflüsse haben.

Diese verschiedenen Einflüsse werden einzeln berechnet und anschließend zu einer gesamten resultierenden Kraft (der „social force“) aufsummiert.

2.1.3 Optimal Steps Model

Das Optimal Steps Model [SK12] verfolgt einen anderen Ansatz als der Zelluläre Automat, um ohne Vorgabe eines Grundrasters zu funktionieren. Das Ziel jedes Fußgängers im Optimal Steps Model ist, seine Position innerhalb des Raumes zu optimieren. Das kann er durch einen Schritt auf einem Kreis [SK12] bzw. einer Kreisscheibe [DKSvS14] um seine Position herum tun. Optimierungskriterien sind dabei die Verringerung der Distanz zum Ziel, Hindernisse zu umgehen und anderen Fußgängern auszuweichen. Der Radius dieses Kreises ergibt sich nach der individuellen Geschwindigkeit des Fußgängers. In besonderen Situationen kann der Fußgänger auch einen Schritt machen, der kleiner ist als der Kreisradius für die normale Geschwindigkeit (siehe Abbildung 2.2). Insbesondere in der Nähe von Hindernissen und anderen Fußgängern muss manchmal so reagiert werden, um Zusammenstöße zu verhindern [DKSvS14]. Dadurch wird gegenüber dem Zellulären Automat der Vorteil erreicht, dass sich die Fußgänger komplett frei im kontinuierlichen Raum bewegen können ohne auf einen Zellenraster beschränkt zu sein.

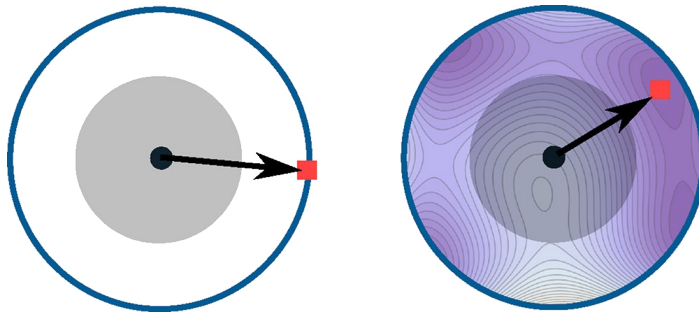


Abbildung 2.2 – Schematische Darstellung der Bewegung eines Fußgängers (grauer Kreis) in einem Zeitschritt nach dem Optimal Steps Model. Links die Bewegung auf dem Kreis nach [SK12], rechts auf der Kreisscheibe nach [DKSvS14] (Abbildung mit freundlicher Genehmigung durch Autorin von Sivers).

Um den optimalen Punkt auf der Kreisbahn oder innerhalb der Kreisfläche zu finden werden Potential-Funktionen angewendet. Das Gesamtpotential für jeden Fußgänger berechnet sich dann über die Summe aus dem Zielpotential $P_t(x)$, den einzelnen Fußgängerpotentialen $P_p(x)$ aller Fußgänger außer dem Betrachteten und allen Hindernispotentialen $P_o(x)$:

$$P_i(x) = P_t(x) + \sum_{j=1, j \neq i}^n P_p^j(x) + \sum_{k=1}^m P_o^k(x).$$

Dieses Gesamtpotential $P_i(x)$ besitzt an der Stelle p^{max} ein Maximum, das als neues Ziel für einen Zeitschritt interpretiert wird.

2.1.4 Gradient Navigation Model

Das Gradient Navigation Model [DK14] beruht darauf den Geschwindigkeitsvektor jedes Fußgängers über eine Differentialgleichung in Abhängigkeit von mehreren skalaren Distanz-Funktionen zu berechnen.

Folgende Gleichungen beschreiben die Position x und die Wunschgeschwindigkeit w (relaxed speed) eines Fußgängers i :

$$\begin{aligned}\dot{\vec{x}}_i(t) &= w_i(t)\vec{N}(\vec{x}_i, t) \\ \dot{w}_i(t) &= \frac{1}{\tau}(v_i(\rho(\vec{x}_i))\|\vec{N}(\vec{x}_i, t)\| - w_i(t))\end{aligned}$$

$$\text{Navigationsfunktion: } N(\vec{x}_i, t) = -g(g(\nabla\sigma(\vec{x}_i)) + g(\nabla\delta(\vec{x}_i, t))).$$

Die Funktionen $\nabla\sigma$ und $\nabla\delta$ haben Ähnlichkeiten zu den Potentialfunktionen des Optimal Steps Models, allerdings werden Gradienten der Potentialvektoren eingesetzt, um die Einflüsse von anderen Fußgängern und Hindernissen zu modellieren. Daraus ergibt sich die Navigationsfunktion, die multipliziert mit der Wunschgeschwindigkeit w des Fußgängers den neuen Geschwindigkeitsvektor für den nächsten Zeitschritt bildet [DKSvS14].

2.2 Kraftfahrzeugsimulation

Während sich die Fußgängersimulation immer im zweidimensionalen Raum bewegt, ist die Situation bei grundlegenden Fahrzeug-Folge-Modellen etwas einfacher, da lediglich eine Dimension, gegebenenfalls ein Spurwechsel, betrachtet werden muss. Im folgenden werden verschiedene Modelle zur mikroskopischen Fahrzeugsimulation vorgestellt.

1. Optimal Velocity Model
2. Nagel-Schreckenberg-Modell
3. Herman's Car Following Model
4. Driving by Visual Angle Model
5. AIMSUN

2.2.1 Optimal Velocity Model

Das Optimal Velocity Model ist ein auf einer Differentialgleichung basierendes Fahrzeug-Folge-Modell, das 1995 erstmals auf der Jülich-Konferenz von Yuki Sugiyama vorgestellt wurde [Sug99]. Es nimmt Fahrzeuge und Fahrer als gleich an und modelliert eindimensionale Fahrzeugbewegungen.

Nach dem Modell steuert der Fahrer sein Fahrzeug nach folgender Gleichung:

$$\ddot{x}_n = a(V(\Delta x_n) - \dot{x}_n), \quad \Delta x_n = x_{n-1} - x_n.$$

Die Variable n gibt dabei die Position des Fahrzeugs in der Fahrzeugschlange und x_n die Position des n -ten Fahrzeugs an. \dot{x}_n gibt durch die Ableitung nach der Zeit die Geschwindigkeit des Fahrzeugs an, \ddot{x}_n entsprechend die Beschleunigung. a ist eine positive Konstante, die die Sensibilität der Reaktion des Fahrers angibt.

V ist die sogenannte „Optimal-Velocity-Funktion“ (OV-Funktion), welche die sichere Geschwindigkeit für den vorausliegenden Abstand zum nächsten Fahrzeug bestimmt. Der Fahrer beschleunigt bzw. bremst aufgrund der Differenz seiner eigenen Geschwindigkeit zur so errechneten Optimalgeschwindigkeit. Diese Funktion ist monoton ansteigend und durch einen Maximalwert begrenzt. Ein einfaches Beispiel für eine solche Funktion ist in [Sug99] wie folgt angegeben:

$$V(\Delta x) = \tanh(\Delta x - 2) + \tanh 2$$

Die Kurve der Optimal Velocity Funktion nach [Sug99] ist in Abbildung 2.3 dargestellt.

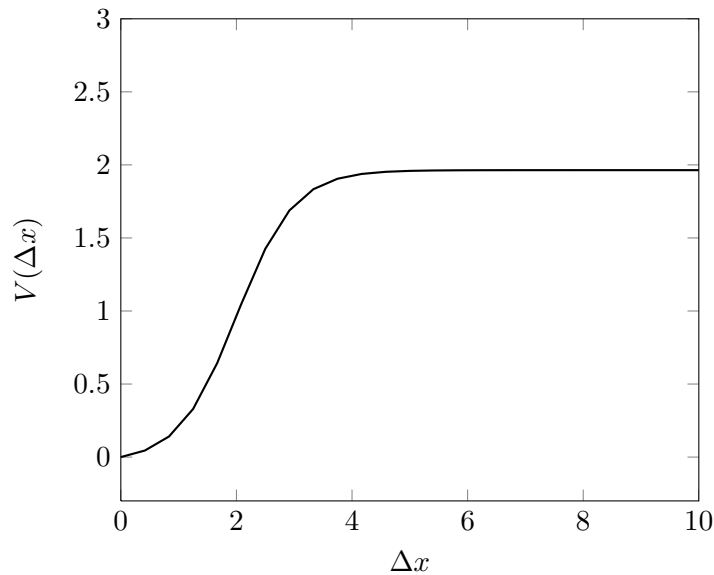


Abbildung 2.3 – Darstellung der Optimal Velocity Funktion nach [Sug99]

2.2.2 Nagel-Schreckenberg-Modell

Das Nagel-Schreckenberg-Modell ist eine Weiterentwicklung des Zellulären Automaten, so dass er für die Fahrzeugsimulation verwendet werden kann. In den ursprünglichen Ansätzen des Zellulären Automaten war nur ein Fortschritt um eine Zelle pro Zeitschritt möglich. Dies wurde dann verallgemeinert, damit Fahrzeuge mehrere Zellen pro Zeitschritt in einem Automaten zurücklegen können. Diese Anzahl wird dann durch ein maximales Limit begrenzt.

Zunächst wurde das Modell als Ring angelegt. So gab es keine Probleme mit dem Durchsatz an Fahrzeugen am Ein- und Ausgang der betrachteten Strecke.

Für die Umsetzung des Modells muss jedes Fahrzeug folgende Attribute besitzen:

- Die eigene Geschwindigkeit v , die in Zellen pro Zeitschritt angegeben wird.
- Die Distanz zum vorausfahrenden Fahrzeug d in Zellen.

Das Fahrzeug an sich beherrscht folgende grundlegende Fahr-Modi, unter denen es in jedem Zeitschritt wählen kann:

- Gleichmäßige Fahrt, bei der die Geschwindigkeit v unverändert bleibt.

- Beschleunigung, bei der die Geschwindigkeit um eine Zelle pro Zeitschritt erhöht wird $v=v+1$.
- Bremsen, bei dem die Geschwindigkeit um eine Zelle pro Zeitschritt verringert wird $v=v-1$.

Diese Fahrt-Modi können dann durch eine Logik mit der Distanz zum vorausfahrenden Fahrzeug d verbunden werden.

Später wurde das Modell auch weiterentwickelt, sodass nicht-zirkuläre Modelle mit gegebenem Fahrzeugdurchsatz am Ein- und Ausgang dargestellt werden konnten. Im Gegensatz zu zirkulären Systemen ist dann aber keine parallele Aktualisierung der Fahrzeugpositionen möglich. Hier muss dann einmalig entschieden werden, ob die Position und Geschwindigkeit der Fahrzeuge in Fahrtrichtung oder dagegen in den Zeitschritten aktualisiert wird.

Konkret wird für die Zellengröße in der Regel die Summe aus der durchschnittlichen Fahrzeuglänge und der Lücke zwischen zwei Fahrzeugen angenommen, das entspricht in etwa 7,5 Meter. Für den Zeitschritt wird typischerweise die Reaktionszeit des Fahrers von etwa einer Sekunde angenommen. So ergibt sich eine Geschwindigkeit von 27 km/h, wenn sich ein Fahrzeug pro Zeitschritt eine Zelle vorwärts bewegt. Mit einer Höchstgeschwindigkeit von fünf Zellen pro Zeitschritt erreicht man dann eine Geschwindigkeit von 135 km/h.

In [BZBP13] wird folgender Ablauf für die Aktualisierung der Position und Geschwindigkeit der Fahrzeuge beschrieben:

1. Beschleunigung, falls die aktuelle Geschwindigkeit kleiner ist als Maximalgeschwindigkeit.
2. Bremsen auf Größe der Lücke in Zellen, falls diese kleiner ist als die Geschwindigkeit.
3. Zufällig bremsen einige Fahrzeuge um eine Zelle pro Zeitschritt ab.
4. Alle Fahrzeuge werden mit ihrer momentanen Geschwindigkeit bewegt.

Mit diesem Modell konnte man das Phänomen der Staus aus dem Nichts als Folge von trödelnden und beim Bremsen überreagierenden Fahrzeugen erklären. Später wurde das Modell durch eine geschwindigkeitsabhängige Wahrscheinlichkeit für trödelnde Fahrzeuge weiter an die Realität angeglichen (Velocity Dependent Randomization Model - VDR-Model [JW06]). Das Modell erlaubt auch eine sehr große Anzahl von Fahrzeugen, da die einzelnen Aktualisierungen auch parallel berechnet werden können.

2.2.3 Herman's Car Following Model

Das 1990 vorgestellte Fahrzeugfolge-Modell von Robert Herman [Her91] basiert ähnlich wie das Optimal Velocity Model auf einer Differentialgleichung. Auch die Grundannahmen sind ähnlich: Alle Fahrzeuge verhalten sich nach den selben Regeln und das Modell ist auf eindimensionale Situationen anwendbar.

Die Veränderung der Beschleunigung zum Zeitpunkt $t+T$ ist proportional zur relativen Geschwindigkeit in Abhängigkeit zum vorausfahrenden Fahrzeug multipliziert mit einem Verstärkungsfaktor λ . Dieser entspricht beispielsweise einem Sensibilitätsfaktor a dividiert durch den Abstand der beiden Fahrzeuge. T ist dabei die Verzögerung der Wirkung der Beschleunigung durch die Reaktionszeit des Fahrers.

Es ergibt sich folgende allgemeine Gleichung:

$$\ddot{x}_n(t+T) = \lambda \left(\frac{d^p x_{n-1}(t)}{dt^p} - \frac{d^p x_n(t)}{dt^p} \right)$$

Für $p = 0$ ist das Modell instabil und alle Werte für $p \geq 2$ sind für den Fahrer schwer zu erfassen (z.B. relative Beschleunigung oder ruckartige Bewegungen). Somit ergibt sich für $p = 1$ folgende Gleichung:

$$\ddot{x}_n(t+T) = \lambda \left(\frac{dx_{n-1}(t)}{dt} - \frac{dx_n(t)}{dt} \right) = \lambda(\dot{x}_{n-1}(t) - \dot{x}_n(t))$$

$$\lambda = \frac{a}{x_{n-1}(t) - x_n(t)}.$$

Als asymptotisches Stabilitätskriterium wird $2\lambda T \leq 1$ angelegt. So sieht man, dass die Stabilität einer Fahrzeuggruppe vom Verhalten der Fahrer dargestellt von der Reaktionszeit T und λ (bzw. dem Sensibilitätsfaktor a) abhängt. Zusätzlich ist darüber zu erkennen, dass über die Funktionen eine maximale Dichte an Fahrzeugen bei einer mittleren Geschwindigkeit möglich ist (vgl. [Her91]).

2.2.4 Driving by Visual Angle Model

2007 publizierten George J. Anderson und Craig W. Sauer ihre Arbeit an einem Modell, dass im Gegensatz zu den vorausgehenden lediglich von Informationen ausgeht, die dem Fahrer durch sein Blickfeld zur Verfügung stehen [AS07]. Als Berechnungsgrundlage für die Distanz zum vorausfahrenden Fahrzeug wird die Breite dieses Fahrzeugs in Abhängigkeit zum Sichtfeld verwendet. Darüber soll eine realistischere Modellierung des Fahrzeugverhaltens ermöglicht werden. Ein weiterer Unterschied zu den zuvor beschriebenen Modellen ist bereits im Ansatz zu erkennen: Anstatt aufgrund der Geschwindigkeit des vorausfahrenden Fahrzeugs zu beschleunigen bzw. zu bremsen, verfolgt der Fahrer ein anderes Ziel wie beispielsweise die Einhaltung des Sicherheitsabstandes und Anpassung der Geschwindigkeit an das vorausfahrende Fahrzeug.

Daneben wurden Ergebnisse von Studien eingearbeitet, inwiefern der Mensch Distanzen wahrnimmt, die weiter entfernt liegen.

Das Modell geht von folgenden Gleichungen und Variablen aus:

- w : Breite des vorausfahrenden Fahrzeugs
- d : Distanz zu vorausfahrendem Fahrzeug
- α : Blickwinkel des vorausfahrenden Fahrzeugs
- α' : Angestrebter Blickwinkel des vorausfahrenden Fahrzeugs
- j, k : Sensibilitätskonstanten

Blickwinkel eines Fahrzeugs: $\theta = \frac{w}{d}$

$$\ddot{x} = j \cdot \left(\frac{1}{\alpha} - \frac{1}{\alpha'} \right) + k \cdot \frac{d}{dt} \alpha$$

Um den angestrebten Blickwinkel α' abhängig von der aktuellen Geschwindigkeit darzustellen, kann folgende Funktion anstatt einem Konstanten Wert verwendet werden:

$$\alpha' = 2 \cdot \operatorname{atan} \left(\frac{w}{\operatorname{timegap} \cdot \dot{x}_{FV}} \right)$$

- \dot{x}_{FV} : Geschwindigkeit des vorausfahrenden Fahrzeugs
- $\operatorname{timegap}$: Angestrebter zeitlicher Fortschritt bzw. Simulationszeitschritt

Sheng Jin et al. haben 2013 dieses Modell verwendet, um eine kombinierte Simulation mit Fahrzeugen und Fußgängern, die die Straße überqueren, zu untersuchen [SQXW13]. Für die Fahrzeugsimulation wurde dazu das Driving by Visual Angle Model weiterentwickelt, um auf Fußgänger-Übergänge zu reagieren. Dazu wurden einige Regeln aufgestellt, nach denen Fußgänger die Straße überqueren und wie die Fahrzeuge darauf reagieren.

Auf der Seite der Fußgänger wird in [SQXW13] folgendermaßen vorgegangen:

- Jeder Fußgänger besitzt eine individuelle Zeitspanne vor dem sich nähernden Fahrzeug, die er für sicher befindet, um die Straßenseite zu wechseln. Aufgrund dieser Zeit kann er, sobald er den Fußgängerüberweg erreicht, feststellen, ob er die Straße vor dem sich nähernden Fahrzeug sicher betreten kann oder nicht. Nach dieser Entscheidung wartet der Fußgänger oder betritt und überquert dann die Straße, worauf das sich nähernde Fahrzeuge entsprechend reagieren muss.
- Zusätzlich ist ein beobachtetes Phänomen, das sogenannte Herden-Verhalten von Fußgängern, miteingearbeitet. Das bedeutet, dass alle am Übergang wartenden Fußgänger die Straßenseite wechseln, sobald ein Fußgänger dies tut.

Für die Fahrzeugsimulation wurde das Driving by Visual Angle Model verändert, so dass bei der Erkennung eines Fußgängers, der die Straßenseite wechselt, an der Position des Überwegs ein virtuelles, stehendes Fahrzeug eingefügt wird, auf das das sich nähernde Fahrzeug reagiert.

Die Erkennung der Fußgänger geschieht ähnlich wie die Erkennung der Geschwindigkeitsänderung des vorausfahrenden Fahrzeugs im Driving by Visual Angle Model über den Sichtwinkel des Fahrers.

Bei der Simulation mit diesem Modell wurde beobachtet, dass die Fahrzeugkapazität des betrachteten Straßenabschnitts durch die Einbringung des Fußgängerüberwegs reduziert wurde.

2.2.5 AIMSUN

AIMSUN ist eine kommerzielle Software zur Simulation von komplexen Straßennetzen. Zur Darstellung von Fahrzeugfolge-Verhalten wurde ein Modell auf Grundlage des Gipps-Modell [Gip81] entwickelt.

Dabei wird als Geschwindigkeit das Minimum der folgenden Geschwindigkeiten gewählt:

- Geschwindigkeit nach Maximaler Beschleunigung bis Maximalwert:

$$v_n^a(t+T) = v_n(t) + 2.5 \cdot a_n^{\max} \cdot T \cdot \left(1 - \frac{v_n(t)}{v_n^{\text{desired}}} \right) \cdot \sqrt{0.025 \cdot \frac{v_n(t)}{v_n^{\text{desired}}}}$$

- Maximale sichere Geschwindigkeit nach Distanz zum vorausfahrenden Fahrzeug:

$$v_n^b(t+T) = d_n^{max} \cdot T + \sqrt{(d_n^{max} \cdot T)^2 - d_n^{max} \cdot \left[2\{x_{n-1}(t) - s_{n-1} - x_n(t)\} - v_n(t) \cdot T - \frac{v_{n-1}(t)^2}{d_{n-1}} \right]}$$

Im Gegensatz zu anderen Modellen bezieht dieses Sicherheitsdistanz-Modell auch eine Wunschgeschwindigkeit sowie Beschleunigungskraft der Fahrzeuge und die Fahrzeuggröße mit ein. Das Modell kommt ohne Differentialgleichungen aus, was den Berechnungsaufwand deutlich reduziert.

2.3 Auswahl der Modelle

Das verwendete Fahrzeugmodell sollte für die Implementierung und Modellierung einer bimodalen Situation folgende Kriterien erfüllen:

- Mikroskopisches Modell
- Einfache Integrierbarkeit
- Einfache Implementierbarkeit
- Einfache Erweiterbarkeit

Alle vorgestellten Modelle sind mikroskopische Fahrzeug-Folge-Modelle, die jedes Fahrzeug einzeln betrachten. Zudem sind alle vorgestellten Modelle eindimensional, was für die Modellierung eines Fußgängerüberweges bereits ausreichend ist. Für die einfache Implementierung bietet sich ein Differentialgleichungsmodell an, da diese durch eine allgemeine Implementierung eines solchen Modells in VADERE einfach zu ergänzen ist. Da einige Modelle wichtige Faktoren nicht berücksichtigen, sollen Parameter wie beispielsweise die Fahrzeuglänge, eine Sichtweite oder eine globale Höchstgeschwindigkeit einfach zu ergänzen sein. Weiterhin soll das Modell einfach um Logiken erweitert werden können, um die Interaktion mit Fußgängern zu simulieren. Aufgrund dieser Kriterien wurde das Optimal Velocity Model gewählt, um Fahrzeuge in VADERE zu simulieren.

Als Fußgänger-Modell soll grundsätzlich jedes Modell, das in der Software VADERE bereits implementiert ist, verwendet werden können, um durch die Betrachtung verschiedener Modelle einer realistischen Simulation näher zu kommen. Die einzige notwendige Anforderung ist, dass Fußgänger stehen bleiben können, z.B. an einer Ampel. Dies erfüllen alle vorgestellten Modelle. Für die Modellierung des Fußgängerüberweges wird daher beispielhaft das Optimal Steps Model herangezogen.

Letztendlich soll es in der Software VADERE möglich sein beliebige Fahrzeug- und Fußgänger-Modelle auf einer gemeinsamen Fläche zu kombinieren. Weitere Fahrzeugmodelle können dann ebenfalls implementiert und in der Interaktion mit Fußgängern verwendet werden.

2.4 Einführung eines bimodalen Simulationsmodus

Es soll ein bimodaler Simulationsmodus entwickelt werden, der die Verbindung zwischen Kraftfahrzeug- und Fußgänger-strömen herstellt. Die Softwareplattform VADERE ist bisher

lediglich auf Fußgänger-Simulation ausgelegt. Daher gilt es zunächst das Fahrzeugmodell und die zugehörigen Komponenten zu implementieren.

Die Situation und damit die Modellierung der Fußgänger verändert sich im kombinierten Modus quasi nicht, da diese lediglich bestimmte Wartepunkte beispielsweise an einer Ampel oder einem Zebrastreifen haben, aber nicht auf Fahrzeuge reagieren müssen. Um diese Situationen noch realistischer abzubilden, könnte aber auch eine umgekehrte Reaktion in einem späteren Schritt ergänzt werden (siehe 5.2).

Die Fahrzeuge hingegen müssen erkennen, ob ein Übergang vor ihnen liegt und ob sie vor diesem halten bzw. abbremsen müssen. Das Fahrzeug erkennt bisher im Optimal Velocity Model lediglich andere Fahrzeuge, auf die es entsprechend der Geschwindigkeit und Distanz reagieren kann. Dies geschieht in der Implementierung in VADERE (siehe 3.2) im Bereich einer vorgegebenen Sichtweite des Fahrers. In dieser Sichtweite ist nun ebenfalls ein Überweg bzw. eine andere Querungsanlage zu berücksichtigen. Bei einer sogenannten Querungsanlage (vgl. [Mü10]) werden drei grundlegende Typen unterschieden:

1. Fußgängerüberweg (Zebrastreifen):
Der Fahrer des Fahrzeugs muss in seiner Sichtweite feststellen, ob ein Übergang existiert, wie die Position von Personen am Übergang ist und daraus eine Entscheidung über die eigene Beschleunigung treffen.
2. Fußgängerfurt mit Ampel:
Diese Form des Übergangs verlangt lediglich eine Reaktion auf vorgegebene Strukturen, die einzelnen Modelle agieren aber weiter getrennt voneinander.
3. Fußgängerfurt ohne Ampel:
Genau umgekehrt zum Fußgängerüberweg ist die Situation an einer Fußgängerfurt ohne Ampelanlage. Hier haben Fahrzeuge Vorfahrt und der Fußgänger muss sich vor dem Überqueren an den Fahrzeugen orientieren.

Für die Modellierung des Fußgängerüberweges ist es lediglich notwendig eine Logik für die Reaktion auf Fußgänger im Fahrzeugmodell zu integrieren. Wie diese Logik aussehen kann, wurde bereits in 2.2.4 dargestellt. Einige dieser Ansätze werden bei der Integration der Fußgängererkennung in das Optimal Velocity Model verwendet.

3 Implementierung der Fahrzeugsimulation

Im Folgenden wird beschrieben, wie die Fahrzeugsimulation, im Speziellen das Optimal Velocity Model, in der Softwareplattform VADERE ergänzt wurde.

3.1 Fahrzeug als Spezialisierung des Fußgängers

Zunächst wurde eine Klasse `Car` als Spezialisierung der Fußgänger-Klasse `Pedestrian` angelegt. Dadurch mussten grundlegende Funktionen in der Darstellung und Berechnung der Simulation nur geringfügig oder gar nicht geändert werden, da diese auf der Klasse `Pedestrian` beruhen.

Die Klasse `Car` ist im Gegensatz zum Fußgänger mit etwas anderen Attributen ausgestattet. So verfügt sie statt über eine kreisförmige Grundfläche über eine rechteckige, die über die Attribute `length` und `width` angegeben werden können. Diese sind auch für verschiedene Modelle wie das Driving by Visual Angle Model wichtig. Zusätzlich verfügt das Fahrzeug über eine Richtung, die als zweidimensionaler Vektor `direction` gespeichert ist. Über diese kann die Reihenfolge der Fahrzeuge in der gleichen Richtung bestimmt werden. Außerdem wird über die Länge des Vektors die Startgeschwindigkeit bei der Erzeugung in der Quelle angegeben.

Die einzelnen Attribute sind in der Klasse `AttributesCar` definiert und können wie bei anderen Modellen in der graphischen Oberfläche der Software VADERE wie in Abbildung 3.1 dargestellt über JSON-Code gesetzt werden.

```
{
  "CAR_ATTRIBUTES": {
    "length": 4.5,
    "width": 1.7,
    "direction": {
      "x": 5.0,
      "y": 0.0
    }
  },
  "OPTIMAL_VELOCITY_MODEL": {
    "attributesODEIntegrator": {
      "solverType": "DORMAND_PRINCE_45",
      "stepSizeMin": 1.0E-4,
      "stepSizeMax": 1.0,
      "toleranceAbsolute": 1.0E-6,
      "toleranceRelative": 1.0E-5
    },
    "sensitivity": 2.0,
    "sightDistance": 100.0,
    "sightDistanceFactor": 2.0,
    "maximumSpeed": 28.0,
    "pedestrianInteraction": false,
    "stopLineX": 0.0
  }
}
```

Abbildung 3.1 – JSON-Code zur Beschreibung der Fahrzeuge und des Optimal Velocity Model

Durch diese Vererbungsstrategie konnte in einem ersten Schritt das Optimal Velocity Model sehr einfach integriert werden, da an der restlichen Software kaum Änderungen nötig waren. Allerdings konnte diese Lösung nicht so für den bimodalen Simulationsmodus verwendet werden, da sich die beiden Typen von Akteuren nur sehr umständlich unterscheiden lassen.

Daher wurde nach der ersten Implementierung des Optimal Velocity Model diese Vererbungshierarchie wieder gelöst, sodass das Fahrzeug `Car` parallel zum Fußgänger `Pedestrian` direkt das Interface `DynamicElement` implementiert.

3.2 Implementierung des Optimal Velocity Model

Die Implementierung des Optimal Velocity Model ist stark parallel zu dem Gradient Navigation Model für Fußgänger-Simulation angelegt, da dieses Modell ebenfalls auf Differentialgleichungen zurückgreift. Durch diese Implementierung war bereits eine allgemeine Klasse für Differentialgleichungsmodelle (`ODEModel1`) vorhanden, von der die neue Klasse für das Modell `OptimalVelocityModel` abgeleitet werden konnte. Zusätzlich wurde eine weitere Klasse `OVMEquations` angelegt, in der die benötigten Gleichungen für das Optimal Velocity Model implementiert sind.

Das Optimal Velocity Model verwendet wie in 2.2.1 dargestellt nachfolgende Gleichung zur Beschreibung der Bewegung der Fahrzeuge mit einer leichten Veränderung. Die OV-Funktion wird mit der über das Modell angegebenen halben Maximalgeschwindigkeit multipliziert. So kann eine vorgegebene Höchstgeschwindigkeit in $\frac{m}{s}$ im Modell berücksichtigt werden. Um das Verhalten der Fahrzeuge nicht zu sehr zu verzerren, wurde die Funktion weiter angepasst. So wird das sonst sehr starke Beschleunigungs- und Bremsverhalten angepasst.

$$\ddot{x}_n = a(V(\Delta x_n) - \dot{x}_n), \quad \Delta x_n = x_{n-1} - (x_n + \text{length})$$

$$V(\Delta x_n) = \frac{v_{max}}{2} \cdot \left(\tanh \left(\frac{4 \cdot \Delta x}{3,6 \cdot v_{max}} - 2 \right) + \tanh 2 \right) = \frac{v_{max}}{2} \cdot \left(\tanh \left(\frac{8 \cdot \Delta x}{3,6 \cdot v_{max}} - 2 \right) + \tanh 2 \right)$$

Die veränderte Kurve der Optimal Velocity Funktion ist in Abbildung 3.2 mit einer Höchstgeschwindigkeit von $28 \frac{m}{s}$ beispielhaft dargestellt.

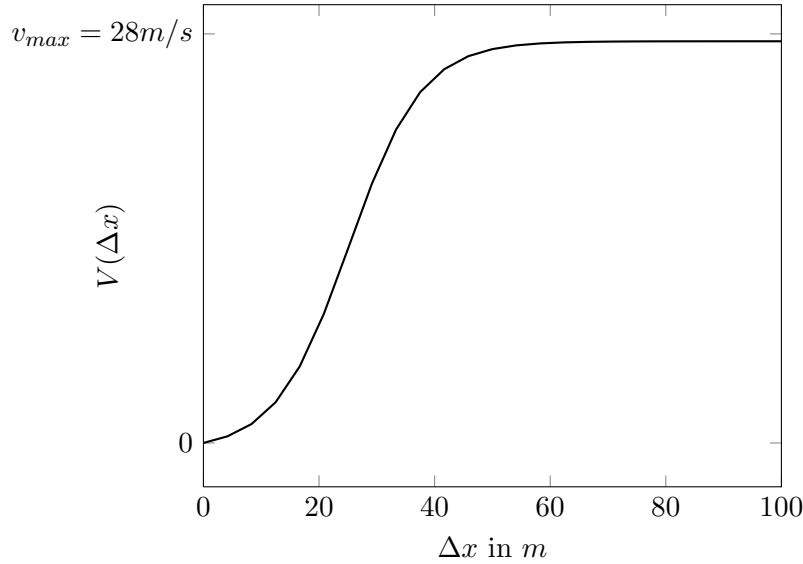


Abbildung 3.2 – Veränderte Optimal Velocity Funktion mit definierter Maximalgeschwindigkeit von $28 \frac{m}{s}$ (etwa $100 \frac{km}{h}$)

Die Beschleunigung des Fahrzeugs wird hier als zweite Ableitung der Position über die gegebene Differentialgleichung beschrieben. Die vorbereitete Klasse `ODEModel` sieht allerdings nur eine einfache Ableitung vor, weswegen diese Gleichung zunächst in eine Gleichung erster Ordnung umgeformt werden musste.

$$\begin{aligned}\dot{x}_n &= v_n \\ \dot{v}_n &= a \cdot V(\Delta x_n) - a \cdot v_n\end{aligned}$$

Durch diese Substitution kann die ursprüngliche Differentialgleichung zweiter Ordnung als System von zwei Differentialgleichungen erster Ordnung dargestellt werden. Durch diese Gleichungen können Position und Geschwindigkeit in jedem Zeitschritt direkt berechnet werden.

Zusätzlich wurde ein weiteres Fahrzeug-Attribut verwendet, die Länge des Fahrzeugs. Beim Vergleich der beiden Positionen muss natürlich die Länge des Fahrzeugs abgezogen werden, damit der echte Abstand zwischen den Fahrzeugen nie kleiner wird als Null.

In jedem Zeitschritt der Simulation wird der berechnenden Methode innerhalb der Klasse `OVMEquations` ein Array übergeben, dass für jedes Fahrzeug die Position und die Geschwindigkeit jeweils in X-Y-Koordinaten enthält. Innerhalb der Methode wird daraus zunächst eine Liste der Fahrzeuge erzeugt. Über diese wird dann iteriert und für jedes Fahrzeug eine neue Position und Geschwindigkeit bzw. deren Ableitungen nach der Zeit berechnet. Dabei werden die oben beschriebenen Gleichung entsprechend auf die übergebenen Parameter angewandt.

Für die Integration nach der Berechnung der Beschleunigung bietet die Software `VADERE` verschiedene numerische Lösungsverfahren (Euler-Verfahren, Dormand-Prince-Methode oder Runge-Kutta-Verfahren) an, die bei der Beschreibung des Modells ausgewählt werden können.

3 Implementierung der Fahrzeugsimulation

Dem Modell wird eine Sichtweite `sightDistance` übergeben, die für alle Fahrzeuge gleichermaßen gilt. Diese legt fest, in welchem Radius um die eigene Position ein Fahrzeug dynamische Elemente in der Umgebung erkennen kann. Über eine bereits implementierte Methode kann zu einer bestimmten Position und einem Radius eine Liste mit Objekten einer bestimmten Klasse erzeugt werden, in diesem Fall der Klasse `Car`. Für jedes Fahrzeug wird eine solche Umgebung erstellt, über die anschließend iteriert wird, um das direkt vorausfahrende Fahrzeug zu ermitteln.

Findet sich in diesem Umkreis kein vorausfahrendes Fahrzeug orientiert sich das Fahrzeug nicht an dieser Geschwindigkeit, sondern an einem imaginären Fahrzeug, dass sich `sightDistance · sightDistanceFactor` entfernt befindet. Der `sightDistanceFactor` wird auch als Attribut des Modells übergeben und ist für alle Fahrzeuge identisch. Durch die Einführung dieses virtuellen Fahrzeugs kann immer dieselbe Differentialgleichung zur Bestimmung der Beschleunigung des Fahrzeugs verwendet werden.

Über die folgenden Diagramme wird die Korrektheit der Implementierung nachgewiesen. In [Sug99] ist das Fundamental-Diagramm, das die Dichte der Fahrzeuge gegen den Fluss darstellt, zum Optimal Velocity Model gegeben (siehe Abbildung 3.3). Im Vergleich zu den Messdaten aus der Simulation (siehe Abbildung 3.4) erkennt man, dass das Verhalten sehr ähnlich ist.

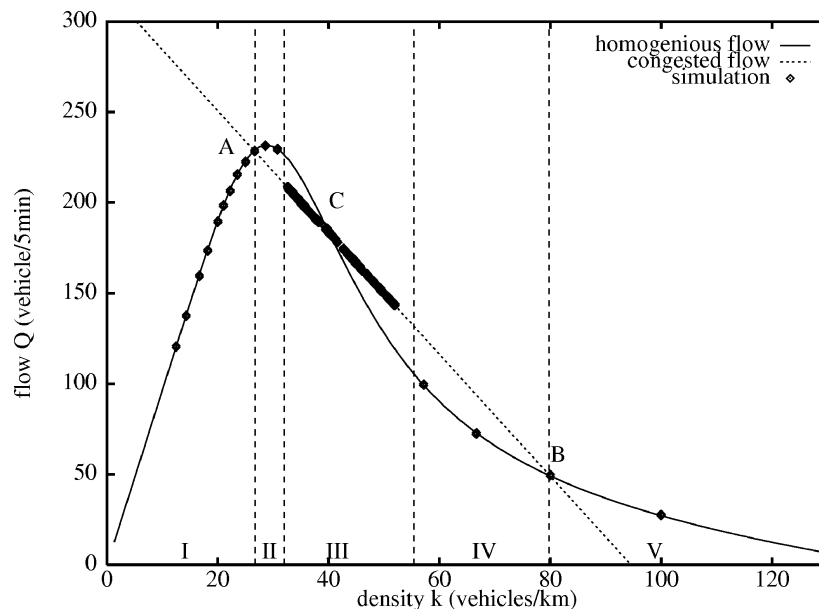


Abbildung 3.3 – Fundamental-Diagramm des Optimal Velocity Model nach [Sug99]

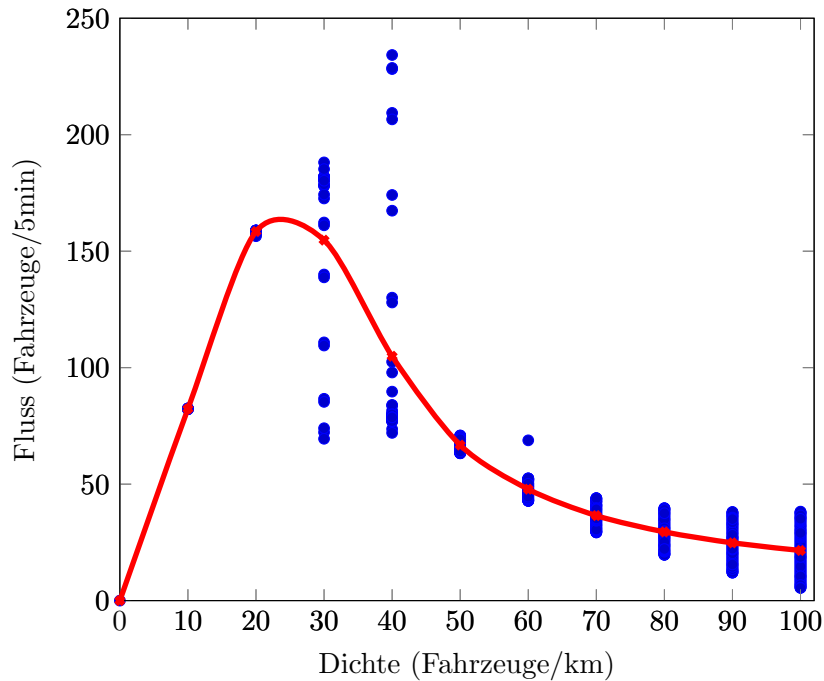


Abbildung 3.4 – Fundamental-Diagramm aus den Simulationsdaten von VADERE, Blaue Punkte: Berechnung des Flusses der einzelnen Fahrzeuge im letzten Zeitschritt, rote Linie: Interpolation der Mittelwerte dieses Flusses

Das Verfahren kann auch aus Sicht der Regelungstechnik betrachtet werden. Die zu regelnde Größe ist die Geschwindigkeit jedes Fahrzeugs. Dazu steht ein Regelsystem auf Grundlage der gegebenen Differentialgleichung zur Verfügung. In Abbildung 3.5 ist das System des Optimal Velocity Model als Blockschaltbild dargestellt. In der implementierten Version ist der rot eingezeichnete Regler nicht vorhanden.

Eine Beobachtung in der Simulation war, dass das Optimal Velocity Model nicht gut geeignet ist, um ein vollständiges Stehenbleiben zu simulieren, da die Geschwindigkeit über die Beschleunigung nur durch ein leichtes Oszillieren um 0 auf einen Wert sehr nah an 0 gebracht werden kann. Daher „wackeln“ die Fahrzeuge leicht, wenn sie hinter einem stehenden Fahrzeug ebenfalls stehenbleiben müssen. Im Blockschaltbild in Abbildung 3.5 ist zu erkennen, dass der offene Regelkreis instabil ist. Durch eine zusätzliche Regelung in der Zustandsrückführung der Geschwindigkeit kann das Problem des Oszillierens behoben werden. Dieser Regler (rot in Abbildung 3.5) ist so zu wählen, dass der geschlossene Kreis stabil ist.

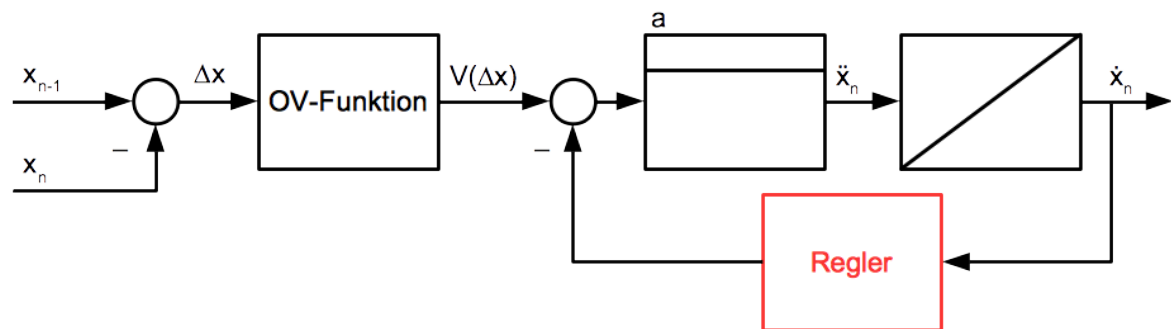


Abbildung 3.5 – Blockschaltbild des Optimal Velocity Model

In der Abbildung 3.6 ist ein Ausschnitt der Simulationsplattform VADERE während der Simulation einer einfachen Fahrzeugkette dargestellt.

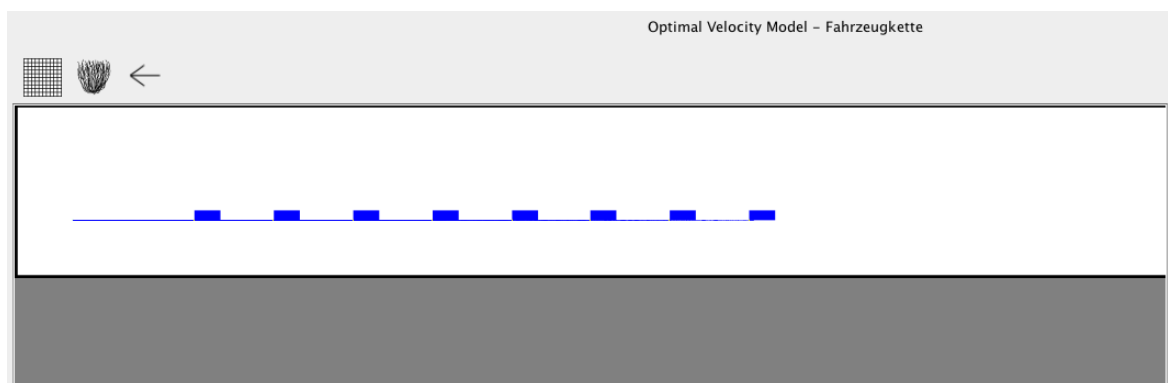


Abbildung 3.6 – Simulation einer Fahrzeugkette in VADERE

3.3 Allgemeine Anpassung der Software

Um den neu geschaffenen Model-Typ in der Softwareplattform VADERE verwenden zu können, waren zusätzliche Anpassungen in der Programmierung der graphischen Oberfläche, der Verwaltung von fixen Datenwerten und der Erzeugung der Modelle notwendig.

3.4 Implementierung des bimodalen Simulationsmodus

Im folgenden wird beschrieben, wie der bimodale Simulationsmodus in der Software-Plattform VADERE realisiert wurde.

3.4.1 Trennung der Fußgänger- und Fahrzeugklassen

Zunächst wurde wie am Ende von 3.1 erwähnt die Ableitung des Fahrzeugs `Car` vom Fußgänger `Pedestrian` wieder aufgelöst und beide implementieren nur noch das allgemeine Interface `DynamicElement`. Hierzu wurden ausgehend von der Methode `getPedestrian()` im `OnlineVisualizationModel` viele Teile des Programms beim Refactoring verändert. So werden jetzt über die veränderte Methode `getDynamicElements()` alle Instanzen, die ein

`DynamicElement` implementieren in der Simulation berücksichtigt. Durch die Trennung der beiden Domänen können die Modelle der zwei Gruppen unabhängig voneinander agieren.

3.4.2 Umsetzung des bimodalen Simulationsmodus

Im Gegensatz zu den Modellen für einen Akteur-Typ muss der bimodale Simulationsmodus mit zwei verschiedenen Akteuren bzw. Modellen arbeiten, die unabhängig voneinander agieren können, im späteren Verlauf aber durch eine gegenseitige Beachtung interagieren. Zunächst wird die Struktur des Simulationsmodus in VADERE betrachtet.

Um die beiden Modelle in einem Szenario anzulegen, müssen nach der Eingabemaske zur Auswahl der Modelle in einem weiteren Dialog (siehe Abbildung 3.7) die beiden gewünschten mikroskopischen Modelle für die Fußgänger- und Fahrzeugsimulation gewählt werden. Für diese wird in der graphischen Oberfläche der entsprechende JSON-Code erzeugt. Im Bereich „Model“ werden nacheinander die beiden Modelle mit ihren jeweiligen Attributen eingebettet. Im Gegensatz zu den Fußgängern, die in einem getrennten Reiter bearbeitet werden können, werden die Attribute der Fahrzeuge auch im Bereich „Model“ beschrieben.

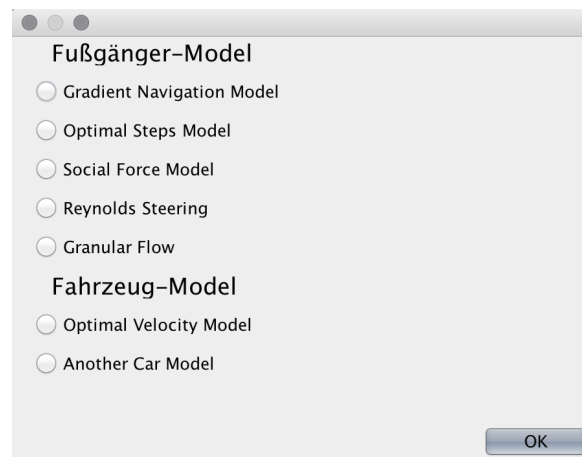


Abbildung 3.7 – Auswahl der gewünschten Modelle für die einzelnen Akteure

Zusätzlich musste auch die Klasse `Source` bzw. `AttributesSource` angepasst werden. Um verschiedene Quellenarten unterscheiden zu können, wurde in den Attributen der Quelle ein zusätzliches Attribut angelegt, das die Art der Quelle speichert. Dafür wurden die statischen Werte `CAR` und `PEDESTRIAN` im enum `DynamicElementType` angelegt. Über diesen Wert kann sowohl im JSON-Code die Quellenart definiert werden, als auch in anderen Teilen des Programms bei der Erzeugung von neuen Quellen und anderen Operationen zwischen den verschiedenen Quellentypen bzw. Akteuren unterschieden werden.

Der zugehörige `SourceController` wurde darauf ausgelegt mit unterschiedlichen Quellenarten zu arbeiten. Bei der Erzeugung wird über das Attribut `type` der übergebenen Attribute der Akteure unterschieden, welcher Akteur erzeugt werden muss.

Beim Start der Simulation werden über den `ModelCreator` zwei `DynamicElementFactory`s mit dem Typ `Pedestrian` und `Car` definiert. Dies geschieht bei der Auswertung der einzelnen mikroskopischen Modelle für die jeweilige Akteur-Gruppe. Diese versorgen dann die Quellen in der Topographie mit den jeweiligen Akteuren.

3.4.3 Integration der Fußgängererkennung in das Optimal Velocity Model

Damit Fahrzeuge an einem Fußgängerüberweg entsprechend auf Fußgänger reagieren können, muss in den Berechnungen der einzelnen Fahrzeuge ein entsprechendes Regelwerk ergänzt werden.

In der Klasse `AttributesOVM` des Optimal Velocity Model kann über den `boolean`-Parameter `pedestrianInteraction` eingestellt werden, ob die Fahrzeuge in der Simulation auf Fußgänger reagieren oder nicht. Nach der Berechnung des vorausfahrenden Fahrzeugs wird dann über eine Liste aller Fußgänger im Sichtbereich iteriert, die analog zur Umkreissuche bei der Bestimmung des vorausfahrenden Fahrzeugs funktioniert. Sofern sich in dieser Liste Fußgänger mit einer Position vor dem Fahrzeug befinden, wird an der X-Position der Haltelinie des Fußgängerüberweges, die über die Modell-Attribute angegeben ist, und der Y-Position des Fahrzeugs ein virtuelles Fahrzeug erzeugt, das dem aktuell betrachteten Fahrzeug als Vorgänger zugeordnet wird. Dabei löst es sich in der Abhängigkeit möglicherweise von dem vor ihm fahrenden Fahrzeug und bremst ab. Sollte jedoch zwischen dem erkannten Fußgängerüberweg und dem Fahrzeug noch ein weiteres Fahrzeug existieren, bleibt dieses weiterhin der Vorgänger.

4 Modellierung einer Querungsanlage

4.1 Fußgängerüberweg

Beim Fußgängerüberweg (Zebrastreifen) muss ein sich näherndes Fahrzeug im Gegensatz zur Ampelanlage überprüfen, ob sich ein Fußgänger in der Nähe des Übergangs aufhält. Im realen Leben kann man als Fahrer nicht zwangsläufig davon ausgehen, dass eine Person, die sich nur in der Nähe des Übergangs aufhält, auch den Wunsch hat die Straßenseite zu wechseln. Dennoch sollten Fahrzeuge vorsichtig abbremsten. In der vereinfachten Betrachtung wird allerdings davon ausgegangen, dass jeder Fußgänger, der im Blickfeld des Fahrzeugs erscheint, die Straße überqueren wird.

In der Abbildung 4.1 ist die betrachtete Situation dargestellt.

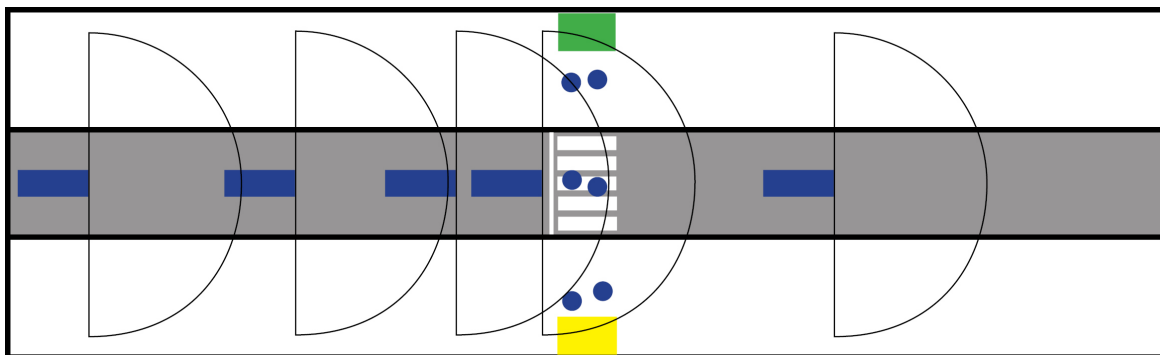


Abbildung 4.1 – Darstellung der Interaktion an einem Fußgängerüberweg:
Fahrzeuge (blaue Rechtecke) fahren von links nach rechts und müssen vor der Haltelinie (weiße Linie) vor dem Zebrastreifen anhalten, um Fußgänger (blaue Kreise), die von der Quelle (grün) zum Ziel (gelb) über den Zebrastreifen laufen. Die Sichtweite der Fahrzeuge ist durch den schwarzen Halbkreis dargestellt.

Wie beschrieben regelt sich das Betreten des kritischen Bereiches „Zebrastreifen“ durch die Interaktion zwischen den Fahrzeugen und Fußgängern. Durch den Vorrang des Fußgängers ist das Fahrzeug im Zweifelsfall gezwungen diesen einzuräumen. Dazu beobachtet der Fahrer über den nach vorne gerichteten Blick die Umgebung. In VADERE werden tatsächlich sämtliche Objekte im Umkreis des Sichtpunktes erkannt, im weiteren Verlauf werden jedoch die hinter dieser Position liegenden Objekte ignoriert. Die erkannten Objekte werden im Falle der Fahrzeuge zur Bestimmung der optimalen Geschwindigkeit verwendet, im Falle eines Fußgängers zum Erkennen eines Fußgängerüberweges vor dem abzubremsten ist. Die nachfolgenden Fahrzeuge bremsen aufgrund des Abbremsen ihres Vordermannes ebenfalls ab.

In VADERE wird zunächst das vorausfahrende Fahrzeug bestimmt. Bei aktivierter Personenerkennung, wird anschließend nach Bedarf ein virtuelles Fahrzeug am definierten Fuß-

gängerüberweg erzeugt.

Durch Setzen des virtuellen Fahrzeugs als „vorausfahrendes“ Fahrzeug löst sich das Fahrzeug, das auf den Zebrastreifen zufährt, vom real vorausfahrenden Fahrzeug. Weitere an den Zebrastreifen heranfahrende Fahrzeuge, die Fußgänger im Blickfeld erkennen, richten sich jedoch auch nach diesem virtuellen Fahrzeug. Dies geschieht auch dann, wenn bereits ein Fahrzeug am Übergang wartet. Dies führt dann zu Überlappungen der Fahrzeuge.

In VADERE ist dieses Problem dadurch gelöst, dass bevor das virtuelle Fahrzeug als vorausfahrendes Fahrzeug gesetzt wird, noch einmal überprüft wird, ob das reale vorausfahrende Fahrzeug sich noch vor der Haltelinie befindet. Sollte dies der Fall sein, ist dieses reale Fahrzeug der korrekte Vordermann, andernfalls das virtuelle Fahrzeug.

Bei der Simulation hat sich ein weiteres Problem gezeigt, das durch die Umkreissuche begründet ist. Fußgänger, die die Straße bereits überquert haben, liegen immer noch im Sichtbereich des Fahrzeugs, sodass dieses weiterhin an der Haltelinie wartet, obwohl der kritische Bereich schon wieder frei wäre. Durch eine Analyse der Bewegungsrichtung der Fußgänger und einem Vergleich mit der Y-Position des Fahrzeugs könnte dieses Problem auch noch behoben werden.

In der Abbildung 4.2 ist ein Ausschnitt einer Simulation eines solchen Fußgängerüberweges in VADERE dargestellt. Die Haltelinie ist hier nicht sichtbar, befindet sich aber auf Höhe der rechten Kante der linken Hindernisse (schwarze Rechtecke).

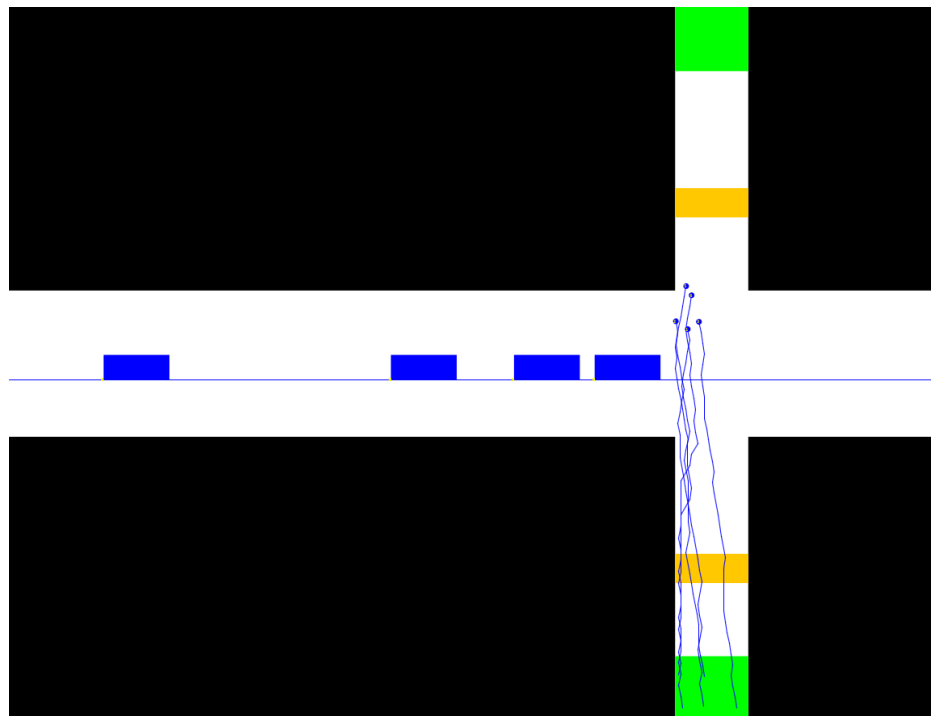


Abbildung 4.2 – Simulation eines Fußgängerüberweges in VADERE mit Haltelinie auf Höhe der rechten Kante der linken Hindernisse analog zur Abbildung 4.1 mit Fußgängerströmen von zwei Seiten, die jeweils zum gegenüberliegenden Ziel gehen.

4.2 Fußgängerfurt mit Lichtsignal

Für die Modellierung eines Fußgängerübergangs mit Lichtzeichenanlage ist eine kombinierte Betrachtung der beiden Modelle ohne eine Interaktion möglich. Durch eine externe Steuerung wird der kritische Bereich des Fußgängerüberwegs jeweils nur einer Akteurgruppe zugeteilt. So muss kein Akteur-Typ auf den anderen während der Laufzeit der Simulation reagieren und es können unveränderte mikroskopische Modelle verwendet werden.

In der Abbildung 4.3 sind die zwei möglichen Situationen dargestellt.

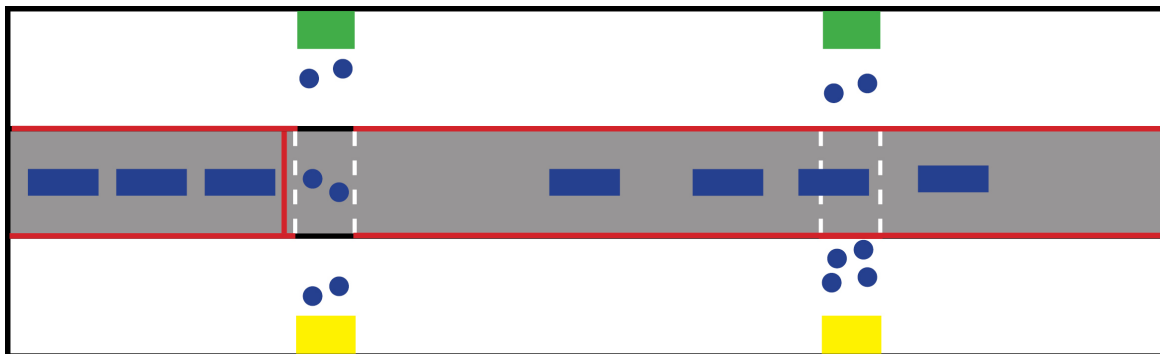


Abbildung 4.3 – Mögliche Situationen an einem durch Lichtzeichenanlage gesteuerten Fußgänger-Übergang analog zu Abbildung 4.1: Ampel auf grün für Fußgänger (links) und Ampel grün für Fahrzeuge (rechts)

Auf der linken Seite der Grafik ist der kritische Bereich auf der Straße zwischen den gestrichelten Linien nur für Fußgänger zu betreten, die von einer Quelle (grün) zu einem Ziel (gelb) streben. Auf der rechten Seite ist dann entsprechend der kritische Bereich nur für Fahrzeuge zugänglich. Fußgänger müssen am Rand der Straße wie vor einem unüberwindlichen Hindernis warten.

4.3 Fußgängerfurt ohne Lichtsignal

Für die Betrachtung der Verkehrssituation an einer Fußgängerfurt ohne Lichtsignal muss umgekehrt zum Fußgängerüberweg die Logik zum Betreten des kritischen Bereichs bei den Fußgängern integriert werden. Die Abbildung 4.4 stellt diese Situation dar.

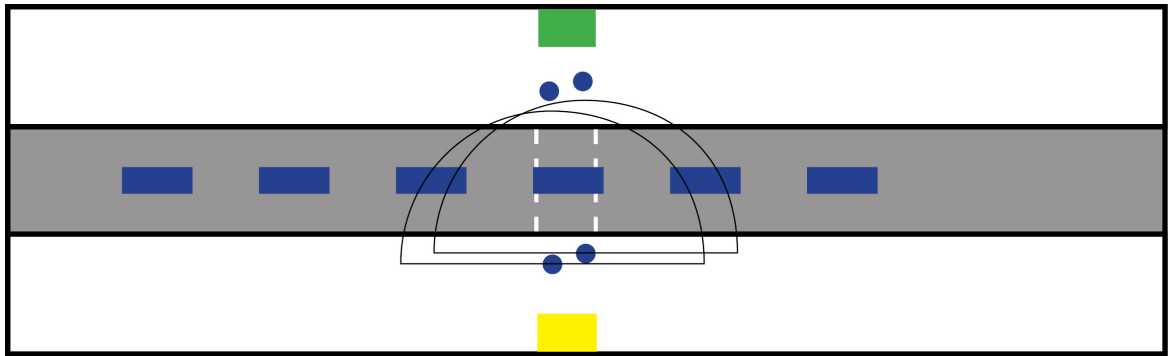


Abbildung 4.4 – Schematische Darstellung der Situation an einer Fußgängerfurt ohne Lichtzeichenanlage analog zu Abbildung 4.1

Die von links kommenden Fahrzeuge haben Vorfahrt. Die Fußgänger von der Quelle (grün) müssen vor Betreten der Straße feststellen, ob das sicher ist. Hier können Ansätze von [SQXW13] übernommen werden. Ähnlich wie bei den Fahrzeugen geschieht das innerhalb eines Sichtbereiches.

5 Fazit

Durch die Implementierung des Optimal Velocity Model und die Integration der Fußgängererkennung ist es nun möglich eine einfache Situation mit den beiden Akteuren, Fußgänger und Fahrzeuge, zu modellieren. Weiterhin können verschiedene Modelle kombiniert werden, allerdings steht derzeit nur ein Fahrzeugmodell zur Verfügung. Das Beispiel des Fußgängerüberweges konnte in VADERE simuliert werden.

5.1 Nutzen und bisheriger Stand

Mit dem Ziel reale Verkehrssituationen simulieren zu können sind viele Schwierigkeiten verbunden. Eine davon ist die in dieser Arbeit behandelte Interaktion zwischen unterschiedlichen Akteuren im Straßenverkehr, in diesem Fall Fußgänger und Kraftfahrzeuge.

Mit der Betrachtung und Implementierung dieser Schnittstelle ist die Softwareplattform VADERE in der Lage in sehr einfachen Situationen, wie einem definierten Fußgängerüberweg, eine Interaktion zwischen Fußgängern und Fahrzeugen zu simulieren.

Anwendungsfälle für diese Simulation und möglichen Erweiterungen (siehe 5.2) sind vielfältig. Im Umfeld des Projekts VADERE ist es sicher auch bei der Simulation von Großveranstaltungen im städtischen Raum interessant die kritischen Bereiche im Umfeld von Straßen nahe des Veranstaltungsortes in die Simulation zu integrieren. Dabei sind zweierlei unter anderem Ergebnisse von Interesse:

1. Die Entwicklung der Fußgängerverteilung vor bzw. nach einem Event oder in einer Entfluchtungssituation
2. Die resultierenden Veränderungen im Straßenverkehr der Umgebung (z.B. Staus und Behinderungen)

Aktuell sind das Modell und die damit darstellbaren Situationen stark vereinfacht. In der Simulation existieren nur identische Fahrzeuge, auf unterschiedliches Fahrverhalten oder Fahrzeuggrößen wird nicht eingegangen. Auch bewegen sich die Fahrzeuge derzeit nur in einer Dimension, in einer vorgegebenen Richtung. Eine weitere Einschränkung ist, dass durch die vorgegebene Position der Haltelinie an einem Fußgängerüberweg nur ein solcher Überweg darstellbar ist.

5.2 Ausblick

Um die in 5.1 beschriebenen Phänomene an komplexeren Situationen untersuchen zu können, ist noch Entwicklungsarbeit nötig. Durch die Entwicklung eines allgemeinen bimodalen Simulationsmodus ist es jedoch möglich verschiedene Modelle leicht zu integrieren. Mit der Betrachtung von Simulationen auf Grundlage von unterschiedlichen Modellen kann man sich dann der realen Situation annähern.

Dabei ist die Interaktion im implementierten Modell aktuell ausschließlich auf Fahrzeuge begrenzt. Wie in [SQXW13] beschrieben, reagieren Fußgänger ebenso auf Fahrzeuge. Daher kann in einer weiteren Entwicklung der Fußgängermodelle die Interaktion mit Fahrzeugen integriert werden.

Folgende Erweiterungen wären in weiteren Entwicklungsschritten denkbar:

- Frei wählbare Fahrtrichtung der Fahrzeuge: Um komplexere Situationen des Straßenverkehrs darstellen zu können, sollten sich die Fahrzeuge entlang gegebener Strecken frei im Raum bewegen können. Dabei ist zu beachten, dass der Algorithmus zur Feststellung des vorausfahrenden Fahrzeugs anzupassen ist.
- Einführung eines Topographie-Elements für Querungsanlagen: Durch ein solches Element könnten in einer Topographie mehrere Querungsanlagen verwendet werden und so reale Situationen besser nachgebildet werden.
- Diversität von Fahrzeugtypen: Um realistischen Verkehr besser modellieren zu können, müssen auch unterschiedliche Fahrzeuge betrachtet werden. Dabei spielt unter anderem die Leistung und Größe des Fahrzeugs, das Fahrverhalten (aggressiv, defensiv, ...) sowie weitere individuelle Eigenschaften eine Rolle.
- Simulation einer T-Kreuzung: Als erstes komplexeres Straßenelement könnte eine T-Kreuzung betrachtet werden, die mit drei Fußgängerüberwegen und diversen Abbiegemöglichkeiten für Fahrzeuge einige Herausforderungen bereitstellt.
- Implementierung weiterer Fahrzeugmodelle: Mit der Betrachtungsmöglichkeit mehrerer Modelle an gleichen Situation können verschiedene Resultate erzielt und wiederum verglichen werden.
- Genauere Analyse der Fußgängerbewegung durch das Fahrzeug: Um zu erkennen, ob ein Fußgänger im Blickfeld die Straße bereits überquert hat, muss das Bewegungsverhalten durch das Fahrzeug erkannt und bewertet werden.

Abbildungsverzeichnis

1.1	VADERE-Simulationsprogramm: Bearbeitung der Simulationsparameter . . .	2
1.2	VADERE-Simulationsprogramm: Bearbeitung der Topographie	3
2.1	Schematische Darstellung eines Zellulären Automaten nach [BKSZ01]	6
2.2	Schematische Abbildung des Optimal Steps Model nach [SK12] und [DKSvS14]	7
2.3	Darstellung der Optimal Velocity Funktion nach [Sug99]	9
3.1	JSON-Code zur Beschreibung der Fahrzeuge und des Optimal Velocity Model	15
3.2	Veränderte Optimal Velocity Funktion mit definierter Maximalgeschwindigkeit	17
3.3	Fundamental-Diagramm des Optimal Velocity Model nach [Sug99]	18
3.4	Fundamantal-Diagramm aus den Simulationsdaten von VADERE	19
3.5	Blockschaltbild des Optimal Velocity Model	20
3.6	Simulation einer Fahrzeugkette in VADERE	20
3.7	Auswahl der gewünschten Modelle für die einzelnen Akteure	21
4.1	Darstellung der Interaktion an einem Fußgängerüberweg	23
4.2	Simulation eines Fußgängerüberwegs in VADERE	24
4.3	Mögliche Situationen an einem durch Lichtzeichenanlage gesteuerten Fußgänger- Übergang	25
4.4	Schematische Darstellung der Situation an einer Fußgängerfurt ohne Lichtzei- chenanlage	26

Literaturverzeichnis

- [AS07] ANDERSEN, GEORGE J. und CRAIG W. SAUER: *Optical Information for Car Following: The Driving by Visual Angle (DVA) Model*. Human Factors, 49(5):878–896, 2007.
- [BKSZ01] BURSTEDDE, C., K. KLAUCK, A. SCHADSCHNEIDER und J. ZITTARTZ: *Simulation of pedestrian dynamics using a two-dimensional cellular automaton*. Physica A: Statistical Mechanics and its Applications, 295:507–525, 2001.
- [BZBP13] BUNGARTZ, HANS-JOACHIM, STEFAN ZIMMER, MARTIN BUCHHOLZ und DIRK PFLÜGER: *Modellbildung und Simulation: Eine anwendungsorientierte Einführung*. Springer-Verlag, Berlin, Oktober 2013. 2. überarb. Aufl.
- [DK14] DIETRICH, FELIX und GERTA KÖSTER: *Gradient navigation model for pedestrian dynamics*. Phys. Rev. E, 89:062801, Jun 2014.
- [DKSvS14] DIETRICH, FELIX, GERTA KÖSTER, MICHAEL SEITZ und ISABELLA VON SIVERS: *Bridging the gap: From cellular automata to differential equation models for pedestrian dynamics*. Journal of Computational Science, 5(5):841–846, 2014.
- [Gip81] GIPPS, P.G.: *A behavioural car-following model for computer simulation*. Transportation Research Part B: Methodological, 15(2):105–111, 1981.
- [Her91] HERMAN, ROBERT: *Traffic dynamics through human interaction: Reflections on some complex problems*. Journal of Economic Behavior and Organization, 15(2):303 – 311, 1991.
- [HM95] HELBING, DIRK und PÉTER MOLNÁR: *Social Force Model for pedestrian dynamics*. Physical Review E, 51(5):4282–4286, 1995.
- [Hoc] HOCHSCHULE MÜNCHEN: *Teilvorhaben der HM - MultikOSi*. <http://www.multikosi.de/teilvorhaben-der-hm>. aufgerufen 11.03.2015.
- [Hü13] HÜTTER, ANDREA: *Verkehr auf einen Blick*. Statistisches Bundesamt, 2013.
- [JW06] JIANG, RUI und QING-SONG WU: *A stopped time dependent randomization cellular automata model for traffic flow controlled by traffic light*. Physica A: Statistical Mechanics and its Applications, 364(0):493 – 496, 2006.
- [Mü10] MÜHR, WENDELIN: *Handbuch barrierefrei im Verkehrsraum : Leitdetails für Planung und Bauausführung/Ingenieurbüro für Barrierefreie Mobilität*. Ingenieurbüro für Barrierefreie Mobilität, Fulda, 2010. 1. Aufl.
- [SK12] SEITZ, MICHAEL und GERTA KÖSTER: *Natural discretization of pedestrian movement in continuous space*. Phys. Rev. E, 86:046108, Oct 2012.

- [SQXW13] SHENG, JIN, XIAOBO QU, CHENG XU und DIAN-HAI WANG: *Dynamic characteristics of traffic flow with consideration of pedestrian road-crossing behaviour*. Physica A, 392:3881–3890, 2013.
- [SSNI95] SCHRECKENBERG, M., A. SCHADSCHNEIDER, K. NAGEL und N. ITO: *Discrete stochastic models for traffic flow*. Phys. Rev. E, 51:2939–2949, Apr 1995.
- [Sug99] SUGIYAMA, YUKI: *Optimal velocity model for traffic flow*. Computer Physics Communications, 121–122(0):399 – 401, 1999. Proceedings of the Europhysics Conference on Computational Physics CCP.
- [ZJWC12] ZHANG, PENG, XIAO-XIA JIAN, S. C. WONG und KEECHOO CHOI: *Potential field cellular automata model for pedestrian flow*. Physical Review E, 85(2-1):021119, 2012.