

# Trajektorien im Kontext:

Einbezug des raumzeitlichen Umfeldes in die Analyse der  
Bewegung von Individuen

Jürg Eggenberger

Masterarbeit

Abteilung Geographische Informationssysteme  
Geographisches Institut der Universität Zürich

Betreut durch Prof. Dr. Robert Weibel und Dr. Patrick Laube

August 2009



**Kontaktinformationen:**

Jürg Eggenberger  
Konrad Hitz-Str. 9  
CH-8816 Hirzel  
ejuerg@gmail.com

**Institut:**

Geographisches Institut  
Universität Zürich  
Winterthurerstr. 190  
8057 Zürich



# Zusammenfassung

Die Analyse von Bewegungsdaten wird für immer mehr Forschungsrichtungen zu einem wichtigen Thema. Neue und verbesserte Ortungstechnologien wie GPS ermöglichen es, die Bewegung von Individuen präzise aufzuzeichnen. Grosse Datensätze erfordern neue Technologien zu deren Speicherung, Abfrage und Auswertung. Bei der Entwicklung dieser neuen Verfahren wurde das Umfeld, in dem die Bewegung stattfindet, weitgehend ignoriert.

In dieser Arbeit wurde der Stand der Forschung zur Integration von räumlichem Kontext in die Analyse von Bewegung zusammengefasst. Um besser zu verstehen, was der Begriff Kontext bedeutet, wurde dieser durch Einteilung in Kontext-Klassen präzisiert. Es konnte gezeigt werden, dass auch bestehende Arbeiten über den Kontext der Bewegung von einer solchen Präzisierung profitieren könnten. Ein Modell für den Kontext der Fortbewegung wurde geschaffen. Dieses zeigt die Relationen zwischen dem sich bewegendem Individuum, den Fortbewegungsarten, deren es sich bedient und der Oberfläche, auf der die Bewegung stattfindet. Dieses Modell diente als Ausgangspunkt für die Verbesserung der RDF von Imfeld (2000). Dabei wurde der euklidische Distanzbegriff zwischen zwei Punkten ersetzt durch die Distanz des kürzesten Weges, der nicht durch das innere von Bewegungshindernissen geht. Der Algorithmus wurde implementiert und ausführlich getestet. Es wurde speziell darauf geachtet, bei welchen Konfigurationen von Hindernissen der Unterschied zwischen den Resultaten der veränderten RDF und der RDF von Imfeld (2000) am grössten ist. So konnte festgestellt werden, wann ein Einsatz des neuen Algorithmus besonders vielversprechend ist.



# Inhaltsverzeichnis

<b>Zusammenfassung</b>	<b>iii</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation.....	1
1.2 Problemstellung und Zielsetzung.....	2
<b>2 Grundlagen</b>	<b>5</b>
2.1 Definitionen.....	5
2.2 Konzepte.....	6
2.2.1 Moving Point Objects.....	6
2.2.2 Trajektorien.....	7
2.2.3 Der Modellbegriff.....	8
2.2.4 Das Random Walk Framework.....	9
2.2.5 Verhalten.....	10
2.3 Forschungsfelder und Forschungsfragen.....	12
2.4 Kontext in der MO-Analyse.....	17
2.4.1 Ist Kontext wichtig?.....	17
2.4.2 Der Places of Interest Ansatz.....	18
2.4.3 Klassifikationen.....	20
2.4.4 Ein Movement Ecology Framework.....	21
2.4.5 Weitere Aspekte.....	22
2.5 Schlussfolgerungen.....	25
2.6 Position der Geographie.....	27
<b>3 Was ist Kontext?</b>	<b>29</b>
3.1 Zwei Typen von Kontext.....	29
3.1.1 Analysekontext.....	29
3.1.2 Bewegungskontext.....	30
3.2 Zwei Herangehensweisen.....	31
3.2.1 Erkundender Ansatz.....	31
3.2.2 Erklärender Ansatz.....	32
3.3 Verknüpfung bestehender Klassifikationen.....	33
<b>4 Modellierung</b>	<b>37</b>
4.1 Anforderungen an das Modell.....	37
4.2 Ein einfaches Modell für Fortbewegungskontext.....	39
4.3 Einbezug der Bewegung.....	40
4.4 Probleme.....	41
4.5 Anwendungsmöglichkeiten.....	42

<b>5 Algorithmus</b>	<b>45</b>
5.1 Bezug zur Literatur.....	45
5.1.1 Die Radial Distance Function.....	45
5.1.2 Shortest Path Probleme.....	47
5.2 Der Algorithmus.....	49
<b>6 Implementierung und Experiment</b>	<b>53</b>
6.1 Implementierung.....	53
6.1.1 Die Java Topology Suite.....	53
6.1.2 Architektur.....	54
6.1.3 Die Klassen und ihre Aufgaben.....	56
6.1.4 Rasterlösung.....	62
6.2 Experiment.....	64
<b>7 Resultate</b>	<b>67</b>
7.1 Zufällige Konfigurationen.....	67
7.2 Spezifische Konfigurationen.....	70
<b>8 Diskussion</b>	<b>75</b>
8.1 Diskussion der Kontext-Begriffe.....	75
8.2 Weitere Integration der Umgehungsdistanz.....	76
8.3 Diskussion des Algorithmus.....	77
8.3.1 Verifikation der Implementierung.....	77
8.3.2 Einfluss der Geometrie der Hindernisse.....	78
8.3.3 Effizienz.....	79
8.3.4 Die Rasterlösung.....	82
8.3.5 Analyseaufwand.....	83
8.3.6 Probleme von Bewegungsdaten.....	83
<b>9 Fazit</b>	<b>85</b>
9.1 Ergebnisse.....	85
9.2 Erkenntnisse.....	87
9.3 Ausblick.....	88
<b>Literatur</b>	<b>89</b>

# Abbildungsverzeichnis

Abbildung 2.1: Prozess der Modellbildung (modifiziert nach Sargent 2005).....	8
Abbildung 2.2: GPS Trajektorien im Space-Time-Cube (Kwan und Lee, 2004).....	14
Abbildung 2.3: Bewegungsökologisches Modell von Nathan et al. (2008).....	20
Abbildung 2.4: Die vier Perspektiven für beschreibende Variablen von Trajektorien (Laube et al. 2007).....	23
Abbildung 3.1: Schematischer Ablauf des erkundenden Ansatzes.....	31
Abbildung 3.2: Schematischer Ablauf des erklärenden Ansatzes.....	32
Abbildung 4.1: Fortbewegungsart-Oberfläche Relationen am Beispiel der Ente.....	37
Abbildung 4.2: Einfaches ER-Modell für Fortbewegungskontext.....	38
Abbildung 4.3: Um die Trajektorie erweitertes ER-Modell für Fortbewegungskontext.....	39
Abbildung 5.1: RDF und Statistik der Ringe (aus Imfeld 2000).....	44
Abbildung 5.2: Temporal RDF-plot (Imfeld 2000).....	44
Abbildung 5.3: Isochronen für die für den Weg benötigte Zeit von einem Punkt aus. (Laurini und Thompson 1992, in Mountain 2005).....	45
Abbildung 5.4: Kürzester Pfad zwischen zwei Punkten mit Hindernissen.....	45
Abbildung 5.5: Visibility Graph für drei Polygone.....	45
Abbildung 5.6: Sichtbarkeit von einem Punkt aus. Blau: nicht sichtbar, gelb: sichtbar (nach De Berg et al. 2000: 311).....	46
Abbildung 5.7: Propagierende Wellenfront (Hershberger und Suri 1993).....	47
Abbildung 5.8: Schattenpunkt (S).....	47
Abbildung 5.9: Ein Umgehungsdistanz-Buffer.....	47
Abbildung 5.10: Umgehungsdistanz-Polygon zerlegt.....	49
Abbildung 6.1: Geometrieklassen der JTS (aus Schmid, 2008).....	51
Abbildung 6.2: Klassendiagramm mit ausgewählten Attributen und Methoden.....	52
Abbildung 6.3: BorderNodes (grün).....	55
Abbildung 6.4: ShadowNodes (pink).....	56
Abbildung 6.5: Konfigurationen beim Hinzufügen eines Dreiecks.....	57
Abbildung 6.6: Resultat eines Runner-Objekts ohne Rekursion (pink).....	58
Abbildung 6.7: Hindernisse als Rasterdaten.....	59
Abbildung 6.8: Kostenberfläche.....	59
Abbildung 6.9: Rasterlösung für die Umgehungsdistanz-RDF.....	60
Abbildung 6.10: Umgehungsdistanz-RDF und Hindernisse.....	62
Abbildung 6.11: Vergleichs-RDF und Hindernisse.....	63
Abbildung 7.1: Vergleich 1 – Zufällige Hindernisse.....	64
Abbildung 7.2: Vergleich 2 – Zufällige Hindernisse, grosse Hindernis-Fläche.....	65

Abbildung 7.3: Vergleich 3 – Zufällige Hindernisse, hohe Dichte.....	65
Abbildung 7.4: Vergleich 4 – Zufällige Hindernisse, geringer Flächenanteil.....	66
Abbildung 7.5: Vergleich 5 – Zufällige Hindernisse mit geringem Einfluss.....	66
Abbildung 7.6: Vergleich 6 – Ein lineares Objekt als Hindernis.....	67
Abbildung 7.7: Vergleich 7 – In einem Netzwerk.....	68
Abbildung 7.8: Vergleich 8 – Grosse Hindernisse.....	69
Abbildung 7.9: Vergleich 9 – Kleine Hindernisse.....	69

## Tabellenverzeichnis

Tabelle 3.1: Einteilung der Faktoren der Klassifikation von Dodge et al. (2008) nach Motivations-, Navigations- und Fortbewegungskontext.....	33
Tabelle 3.2: Einteilung der Faktoren der Klassifikation von Andrienko und Andrienko (2007) nach Motivations-, Navigations- und Fortbewegungskontext.....	35
Tabelle 4.1: Aus Bewegungsdaten abgeleitete Aktivitätsklassen (nach Dykes und Mountain 2003: 588).....	36



# 1 Einleitung

*„Die Physik [zur Zeit Galileos] zum Beispiel war eine allgemeine Bewegungslehre, die die Natur der Bewegung ohne Rücksicht auf die Umstände untersuchte, unter denen sie eintrat.“ - Feyerabend (1983: 222)*

Einhergehend mit der Verbesserung und Verbreitung von GPS und anderen Ortungstechnologien hat sich ein Forschungsfeld entwickelt, das sich mit der Speicherung, Verwaltung, Bearbeitung und Auswertung grosser Datensätze, von Positionsdaten sich bewegender Objekten, beschäftigt. Moderne Geräte können Positionsdaten in hoher Präzision, kleinen Intervallen und über grössere Zeiträume erfassen. Mit diesen Fortschritten einhergehend wurden Datenbanksysteme für bewegte Objekte (*moving object data bases*, MODB) entwickelt, die diese Daten verwalten können.

Nun, da die Grundlagen für die Speicherung und Verwaltung der Positionsdaten geschaffen sind, besteht eine Chance, sich neu zu orientieren, sich Forschungsinteressen und Forschungsfragen zu widmen, die bisher vernachlässigt wurden. Eine Richtung, die das Forschungsfeld einschlagen könnte, geht weg von der Fokussierung auf Zeitpunkt und Position der GPS-Messpunkte, und hin zur Integration des Kontexts, in dem die Bewegung stattfindet.

## 1.1 Motivation

Geographische Informationssysteme (GIS) wurden in ihrer Entstehung stark von der traditionellen, statischen Kartographie geprägt. Gelegentliche Änderungen der Daten waren lange Zeit beachtliche Herausforderungen, an sich ständig mit der Zeit ändernde Daten war gar nicht zu denken (Gudmundsson et al. 2008). Dennoch fanden GIS bald Eingang in die Analyse von Bewegungsdaten aus ökologischen Studien (Imfeld 2000). Teildisziplinen wie die Bewegungsökologie, die Wildtierbiologie oder die *Time Geography* (Hägerstrand 1970) beschäftigen sich schon seit langem mit der Auswertung von Daten, die über die Bewegung von Individuen gesammelt wurden. Das Erheben dieser Datensätze war jeweils mit grossem Aufwand verbunden, und die resultierenden Datensätze waren in der Regel eher klein.

Mit der Möglichkeit relativ einfach grosse Mengen an Bewegungsdaten zu sammeln, ist ein neues Forschungsfeld entstanden, an dem viele Fachrichtungen beteiligt sind. Grosse Datensätze benötigen spezialisierte Technologien für Speicherung, Abfrage und Auswertung. Die darin enthaltenen Daten müssen unter Umständen laufend abgefragt werden können, auch während sie aktualisiert werden. Für die Analyse dieser Daten sind her-

kömmliche statistische Methoden nur beschränkt geeignet. Um interessante Muster in riesigen Mengen von Bewegungsdaten zu finden, werden neue Verfahren benötigt.

Wenn das Analysieren von Bewegungsdaten, und nicht deren Speicherung und Verwaltung, im Mittelpunkt des Interesses steht, muss man von der datenzentrierten Sichtweise wegkommen. Denn Bewegung, als Resultat von Verhalten, das die räumliche Position eines Individuums verändert, ist immer die Folge von einer Interaktion zwischen einem System und seiner Umgebung (Pfeifer und Bongard 2006). Ein lebendiger Organismus beispielsweise muss seine funktionale Balance aufrecht erhalten. Um dies zu tun, ist Interaktion mit der der Umgebung nötig. Dazu muss er über Möglichkeiten verfügen, relevante Aspekte der Umgebung wahrzunehmen und dann physisch mit seinem Umfeld zu interagieren. Die Bewegung ist eine evolutionär bewährte Strategie in diesem Zusammenhang. Wenn Bewegungsdaten also analysiert, erklärt und interpretiert werden sollen, kommt man nicht darum herum, Informationen über das Umfeld in dem diese Bewegung stattfindet, herbei zu ziehen.

## 1.2 Problemstellung und Zielsetzung

Ein grosser Teil der Literatur über die Analyse von Bewegung geht nicht auf den Kontext der Bewegung ein. Es lassen sich aber dennoch einige Arbeiten finden, die auf den Kontext eingehen, in dem die Bewegung stattfindet, oder auf einen spezifischen Aspekt der beeinflussenden Umgebung eines *moving object* (MO). Auch haben mehrere Autoren den fehlenden Einbezug des Kontexts in der MO-Analyse erkannt und den Forschungsbedarf identifiziert.

Es stellt sich die Frage, wie Informationen über den Kontext benutzt werden können, um die Analyse von Trajektorien zu verbessern. Aus dieser Frage lassen sich die folgenden Ziele für diese Arbeit ableiten:

- (1) Ein Überblick über die Forschung soll geschaffen werden. Im Zusammenhang mit dem Kontext der Bewegung relevante Literatur soll vorgestellt werden. Dabei soll speziell auch auf die Rolle der Geographie in diesem interdisziplinären Forschungsfeld eingegangen werden.
- (2) Konzepte sollen erarbeitet werden, die helfen, den Begriff Kontext im Rahmen der Analyse von Bewegung besser zu verstehen.
- (3) Ein Modell soll entworfen werden, das es erlaubt Informationen über den Kontext in die Analyse von Bewegung zu integrieren.
- (4) Aufgrund des Modells soll eine Anwendung konzipiert und implementiert werden, die Kontext in die Analyse von Bewegung integriert.

Ziel (1) bildet die Grundlage der Arbeit. Schlussfolgerungen aus dem Studium der Literatur liefern das Wissen, mit dem Ziel (2) erreicht werden kann. Dabei soll auch darauf geachtet werden, dass bestehende Forschung mit den erarbeiteten Konzepten kompatibel ist. Ziel (3) erfordert eine Einschränkung. Kontext, soviel ist jetzt schon klar, ist ein breiter Begriff. Es wird nicht möglich sein, auf das ganze Spektrum der Faktoren des Kontexts der Bewegung einzugehen. Ziel (2) sollte die Grundlagen für eine solche Einschränkung liefern. Die Anwendung (4) soll auf dem Modell (3) aufbauen. Sie soll zeigen, wie Informationen über den Kontext in der Analyse von Bewegungsdaten verwendet werden können.



## 2 Grundlagen

Die Analyse von Bewegung ist für viele Richtungen der Wissenschaft von zunehmender Bedeutung (Laube 2005: 11). Dieses Kapitel gibt eine Übersicht der grundlegenden Konzepte und Begriffe, der involvierten Forschungsfelder und der wichtigen Forschungsschwerpunkte. Danach wird auf die Literatur eingegangen, die sich konkret mit dem Kontext, in dem Bewegung stattfindet, auseinandersetzt.

### 2.1 Definitionen

Bewegung und Kontext sind die zwei zentralen Begriffe dieser Arbeit. Da beide Begriffe schwer fassbar sind, werden sie in diesem Abschnitt definiert, und gegenüber anderen Begriffen abgegrenzt. Zusätzlich müssen gewisse Einschränkungen gemacht werden.

Der Duden (1996) setzt **Kontext** mit „*umgebender Text, Zusammenhang, Inhalt [eines Schriftstücks]*“ gleich. Merriam-Webster's Online Dictionary<sup>1</sup> definiert *context* folgendermassen:

- „1. the parts of a discourse that surround a word or passage and can throw light on its meaning*
- 2. the interrelated conditions in which something exists or occurs: environment, setting <the historical context of the war>“*

Die erste Definition ist ausschliesslich auf Texte bezogen. Auch im Duden werden Texte erwähnt. Wenn wir über den Kontext der Bewegung diskutieren, kommt daher die zweite Definition zum Zug. In einem Text gibt der Kontext Aufschluss über die Bedeutung der Wörter. Auch die Beschäftigung mit dem Kontext der Bewegung kann einen solchen Mehrwert generieren. Kenntnis des Kontexts erlaubt es uns, Eigenschaften und Muster der Bewegung zu interpretieren und das Verhalten, das zu diesen Mustern führt, zu verstehen. Im Rahmen dieser Arbeit wird die folgende Definition verwendet:

**Kontext** ist die Menge aller umgebenden Faktoren, die ein Ereignis beeinflussen oder zum Verständnis dieses Ereignisses beitragen.

Im Rahmen der Bewegung sind dies sicher alle physischen Interaktionen, die während der Bewegung stattfinden. Aber auch die Motivation, die zur Bewegung führt, gehört zu deren Kontext. Was hier nicht zum Kontext der Bewegung gezählt wird sind Eigenschaften dieser Bewegung. Dies sind z.B. die Geschwindigkeit oder die zurückgelegte Distanz.

Um Bewegung zu definieren wird der Begriff der Veränderung benötigt. Veränderung, im Bezug auf Geodaten, kann nach Blok (2000) drei Formen annehmen. Die erste Form ist das Auftauchen oder Verschwinden eines Objekts. Der Gültigkeitsbereich des Objekts ist

---

<sup>1</sup> <http://www.merriam-webster.com/dictionary/context>, Zugriff: 26.02.09

zeitlich beschränkt. Die zweite Art der Veränderung ist die Mutation. Die Attribute eines Objekts verändern sich. Die dritte Form der Veränderung ist die Bewegung. Die Bewegung ist entweder eine Veränderung der Form, eine Änderung der Grösse oder eine Änderung der Position entlang eines Pfades (Frank 2001: 22). Definitionen von Bewegung, die in der MO-Literatur zu finden sind, beinhalten denn auch die Begriffe Änderung und Standort oder Position. Nathan et al (2008: 19052) definieren Bewegung als eine Veränderung des räumlichen Standorts eines ganzen Individuums mit der Zeit. Bei Andrienko (2008: 18) ist die Bewegung eine Änderung in der Position eines Objekts relativ zu einem Referenzsystem, das die Bestimmung der Position des Objekts zulässt.

Die zweite Art der Bewegung, bei der ein Objekt seine Form verändert, wird in dieser Arbeit nicht behandelt. Bewegungen von Körperteilen, die z.B. in den Sportwissenschaften oder der Computergraphik eine wichtige Rolle spielen, interessieren uns im Rahmen dieser Arbeit also nicht. Bewegung ist hier immer eine Veränderung der Position, und die Position wird durch einen Punkt repräsentiert. Für diese Arbeit gilt die folgende Definition:

**Bewegung** ist die Veränderung der Position eines Objekts relativ zu einem Bezugssystem.

## 2.2 Konzepte

Mit einer Arbeitsdefinition von Kontext und Bewegung ist es nun möglich, einige zentrale Begriffe der MO-Forschung zu diskutieren. *Moving Point Objects* und Trajektorien sind die Untersuchungsobjekte der MO-Analyse. Das *Random Walk Framework* ist ein prägendes Forschungsparadigma. Verhalten ist untrennbar mit Bewegung verknüpft und darf deshalb hier nicht fehlen. Da Modelle eine wichtige Rolle in dieser Arbeit spielen, wird auch auf den Modellbegriff eingegangen.

### 2.2.1 Moving Point Objects

*Moving objects* sind Entitäten, deren Position oder geometrische Form sich mit der Zeit ändert. Weil in vielen Applikationen nur die Position der Objekte relevant ist, reicht es oft aus, das Objekt durch einen Punkt zu repräsentieren (Dodge et al. 2008: 240). Daher wird oft der Begriff *moving point object* (MPO) verwendet. In dieser Arbeit wird trotzdem vorrangig der Begriff *moving object* (MO) verwendet.

Bewegung findet im geographischen Raum statt. Es können aber auch metaphorische Bewegungen in abstrakten Räumen untersucht werden. Laube und Imfeld (2002) analysieren die Bewegung von Schweizer Gemeinden in einem abstrakten ideologischen Raum, dessen Dimensionen links/rechts und ökologisch/technokratisch sind. Auch die Bewegung von *Gaze Points* bei der Analyse von Augenbewegungen sind abstrakte Bewegungen (Dodge et al. 2008: 240). Solche Bewegungen, die in abstrakten Räumen stattfinden, wer-

den in dieser Arbeit ausgeklammert. Der Kontext abstrakter Bewegung, falls es so etwas gibt, ist nicht Gegenstand dieser Arbeit.

Das Verhalten von physischen, nicht biologischen Objekten ist so einfach, dass es möglich ist, es mit derzeit existierenden mathematischen Mitteln zu beschreiben (Dawkins 1987: 5). Die Bewegung eines fallenden Steines ist komplett durch die, den Gesetzen der Physik gehorchenden, Interaktionen mit seiner Umwelt determiniert. Wenn also der Kontext des fallenden Steines bekannt ist, lässt sich die Bewegung berechnen und vorhersagen. Viel interessanter ist die Bewegung von Objekten, die Möglichkeiten haben, ihre Bewegung zu beeinflussen. Diese Bewegungen sind nicht rein physikalisch determiniert. Kenntnis des Kontexts hilft hier, die Bewegung zu verstehen, sie zu interpretieren und sie in ein Verhältnis zu Verhalten und Aktivitäten zu setzen.

### 2.2.2 Trajektorien

Eine Trajektorie ist der Pfad, den ein MO im Raum, in dem es sich bewegt, zurücklegt (Andrienko et al. 2008: 18). Trajektorien können nicht in ihrer Gesamtheit gemessen werden. Trajektorien werden deshalb durch eine Menge von Positionen  $(x, y)$ , die jeweils zu einem Zeitpunkt  $(t)$  gemessen wurden, repräsentiert. Diese Messpunkte werden auch *fixes* genannt. Trajektorien haben einen zeitlichen Anfang und ein (zumindest vorläufiges) Ende. Dazwischen liegen beliebig viele Zeitpunkte. Theoretisch gibt es für jeden Zeitpunkt zwischen dem Anfang und dem Ende der Trajektorie eine Position, die das Objekt zu diesem Zeitpunkt hatte. Natürlich können nicht alle diese Positionen gemessen und gespeichert werden. Die Trajektorie muss diskretisiert werden. Positionen werden als Sequenzen gemessen, mit einem variablen oder festen zeitlichen Abstand zwischen den einzelnen Messungen. Je mehr Messpunkte in einem Zeitintervall, desto genauer wird die Bewegung abgebildet.

Eine Messung der Position besteht aus folgenden Daten:

- (1) Eine Identifikation des Objekts oder der Trajektorie, damit die Messung der entsprechenden Trajektorie zugewiesen werden kann.
- (2) Ein Zeitpunkt  $t$ . Dieser kommt pro Trajektorie nur einmal vor. Ein Objekt kann zu einem Zeitpunkt nur an einem Ort sein.
- (3) Eine Position, im Normalfall bestehend aus einer  $x$ - und einer  $y$ -Koordinate. Es kann aber zusätzlich auch eine Höhe  $z$  gemessen werden. Dabei werden als Referenzsystem der Bewegung rechtwinklige, geographische Koordinaten verwendet. Es wäre aber auch möglich Polarkoordinaten oder andere Systeme zu verwenden.

Eine Trajektorie ist also eine Sequenz von Positionen, geordnet in der Reihenfolge, in welcher sie besucht wurden (Andrienko et al. 2008b: 29).

Aus aufeinander folgenden Positionsmessungen lassen sich Eigenschaften der Bewegung ableiten. Diese Eigenschaften werden von Dodge et al. (2008) diskutiert und klassifiziert. Aus  $(x, y)$  lassen sich die Distanz und die Richtung der Bewegung bestimmen. Daraus wiederum können die räumliche Verteilung, die Änderung der Richtung oder die Sinuosität der Trajektorie bestimmt werden. Aus den Zeitpunkten der Messungen lassen sich Dauer, temporale Verteilung und die Veränderung der Dauer bestimmen. Werden Zeit und Position zu  $(t, x, y)$  kombiniert, können Geschwindigkeit und Beschleunigung bestimmt werden.

Anfang und Ende einer Trajektorie sind gewöhnlich Anfang oder Ende einer Messung. Trajektorien können aber auch als, im Kontext der Anwendung, bedeutungsvolle Reisen definiert werden (Spaccapietra et al. 2008). Ein Beispiel dafür ist die jährliche Migration von Zugvögeln. Anfang und Ende einer Trajektorie ist dann Start und Ende einer solchen Reise.

### 2.2.3 Der Modellbegriff

In dieser Arbeit wird der Modellbegriff von Stachowiak (1973) verwendet. Stachowiak definiert das Modell als dreistellige Relation zwischen modelliertem Objekt, dem Modell und dem modellierenden Subjekt. Ein Modell hat drei Hauptmerkmale:

- (1) Das Abbildungsmerkmal. Ein Modell ist eine Abbildung, es repräsentiert ein natürliches oder ein künstliches Objekt.
- (2) Das Verkürzungsmerkmal. Nicht alle Attribute des Originals werden modelliert. Es wird nur abgebildet, was dem modellierenden Subjekt relevant erscheint.
- (3) Das pragmatische Merkmal. Modelle sind zweckorientiert. Sie werden für Subjekte geschaffen und ersetzen für diese das Original für eine Zeitspanne in Bezug auf bestimmte Operationen.

Von einem Problemgegenstand wird abstrahiert und es werden Theorien entwickelt, die dann durch Modellierung zu einem konzeptuellen Modell führen. Der Modellbildungsprozess führt über mehrere Stufen, bis hin zur Implementierung, Anwendung und Auswertung des Modells. Abbildung 2.1 zeigt die Stufen der Modellbildung nach Sargent (2005). Zusätzlich ist aufgeführt, welches Kapitel dieser Arbeit sich mit welcher Stufe auseinandersetzt.

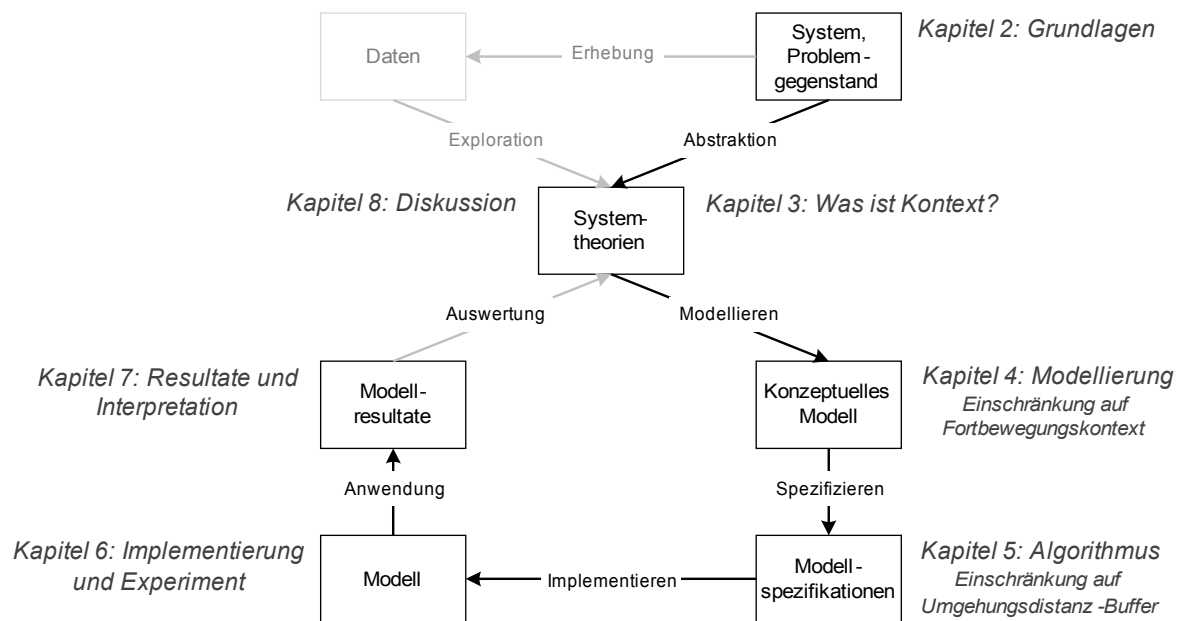


Abbildung 2.1: Prozess der Modellbildung (modifiziert nach Sargent 2005)

## 2.2.4 Das Random Walk Framework

Der *random walk*, auch bekannt als *drunkard's walk*, ist ein mathematisches Modell rein zufälliger Bewegung, das auf dem Gebiet der Analyse und Simulation von Bewegung von grosser Bedeutung ist.

Das *random walk framework* geht davon aus, dass sich Organismen in einer Serie von Bewegungen (*moves*) durch ihre Umwelt bewegen (Turchin, 1998: 128). Vor einem solchen *move* entscheidet sich der Organismus für die Richtung, die Geschwindigkeit und die Distanz des *moves*. Turchin (1998) nennt folgende gebräuchlichen *random walk* Modelle:

*Uncorrelated random walk*: Der vorherige *move* beeinflusst den nächsten *move* nicht. Jeder *move* ist daher zufällig und unabhängig vom vorherigen *move*.

*Correlated random walk*: Eigenschaften des vorherigen *moves* beeinflussen den nächsten *move*. So kann die Wahrscheinlichkeit, dass der nächste *move* in ungefähr die gleiche Richtung geht grösser sein als die Wahrscheinlichkeit eines *moves* in die Entgegengesetzte Richtung. Oder es ist wahrscheinlicher, dass auf einen langen *move* wieder ein *move* über grosse Distanz folgt.

*Biased random walk*: Die Richtung des *moves* ist von einer absoluten Richtung beeinflusst. Eine Bewegung in die festgelegte Richtung ist wahrscheinlicher als eine Bewegung in eine andere Richtung.

*Biased correlated random walk*: Jeder *move* wird vom vorherigen *move* und von einer absoluten Richtung beeinflusst.

*Constrained random walk*: Beobachtete Werte und Verteilungen werden zur Festlegung der *moves* verwendet. Die Werte für einen *move* werden zufällig aus einer Verteilung, die gemessenen Werten entspricht, bestimmt.

Turchin (1998:128) nennt in seiner Definition des *random walks* mehrere Faktoren, die einen *move* beeinflussen können. Diese Faktoren sind die Beeinflussung durch die Erfahrung, durch örtliche Zustände, durch Zustände am Zielort und durch eine absolute Richtung (z.B. die Windrichtung). In den gebräuchlichen Modellen kommt aber nur einer dieser Faktoren vor: die Beeinflussung durch eine absolute Richtung. Erfahrung des Individuums, örtliche Zustände am Ausgangspunkt und am Zielort der Bewegung, fließen nicht in die Modelle ein.

Imfeld (2000: 2) nennt die Idee, dass die Bewegung von Tieren *random walks* sind eine weit verbreitete falsche Auffassung in der Wildtierbiologie. Seiner Meinung nach haben Tiere klare Muster in ihrem Bewegungsverhalten, die aber zu komplex sind, um mit einfachen Methoden gefunden zu werden. Wenn wir annehmen, dass Individuen eine Form von *random walk* ausführen, nehmen wir implizit auch an, dass sie keine externe Information zur Navigation verwenden (Nathan et al. 2008: 19056).

Nathan et al. (2008) nennen das Zufalls-Paradigma als vorherrschendes Paradigma in der Bewegungsökologie. Die gebräuchlichsten Modelle sind *random walk* und Diffusionsmodelle. Sie kommen zum Schluss, dass die Zeit reif sei für eine Erweiterung dieser Herangehensweise um Faktoren wie die Heterogenität der Umgebung, die Wahrnehmung der Umgebung, Erinnerung und andere Mechanismen, die zu verschiedenen Bewegungsmustern führen.

### 2.2.5 Verhalten

In der MO-Forschung taucht der Begriff Verhalten ab und zu auf. Andrienko et al. (2008b: 28) definieren *behaviour*:

*“We introduce the notion of behaviour: this is the configuration of characteristics corresponding to a given reference (sub)set. The notion of behaviour is a generalisation of such notions as distribution, variation, trend, dynamics, trajectory.”*

Diese Definition ist fokussiert auf Muster in den Trajektorien, und mag für diesen Zweck geeignet sein. Im Rahmen dieser Arbeit wäre eine einfachere Definition wünschenswert. Das Feld der *Artificial Intelligence* hat sich ausführlich mit Verhalten auseinandergesetzt. Intelligenz drückt sich im Verhalten aus, und um künstliche Intelligenz zu erzeugen muss dieser Begriff verstanden werden. Pfeifer und Scheier (1999: 646) definieren den Begriff Verhalten folgendermassen:

“[Behaviour is] *what an autonomous agent is observed doing. Always the result of an interaction of an agent with its environment.*”

Der amerikanische Psychologe Herbert Simon, bekannt für seine Forschung auf dem Gebiet der *Behavioural Economics* und *Artificial Intelligence*, benutzte die Metapher von einer einfachen Ameise die sich auf einem Strand bewegt, um die Prinzipien rund um den Begriff Verhalten zu illustrieren (Simon 1969). Die Metapher illustriert die Probleme, denen wir begegnen, wenn wir uns mit Verhalten auseinandersetzen. Sie hat ihren Weg in moderne Lehrbücher der *Artificial Intelligence* gefunden (Pfeifer und Scheier 1999: 112-114; Pfeifer und Bonguard 2006).

Die Metapher beginnt mit einer Ameise, die sich auf einem Strand bewegt. Vielleicht hat sie nach Nahrung gesucht und kehrt zum Bau zurück. Der Strand ist keine homogene Fläche. Steine, Kiesel und Treibgut liegen herum, es hat Pfützen, es hat Flächen mit gröberem und solche mit feinerem Sand, steile und flache Parteien. Die Trajektorie der Ameise würde für einen Beobachter aussehen wie ein komplexer Pfad zwischen verschiedenen Hindernissen hindurch. Für die Ameise sieht die Welt total anders aus als für den Beobachter. Die Ameise ist viel kleiner und hat andere Sensoren für die Wahrnehmung der Welt um sie herum. Sie hat ein anderes Gehirn um die Signale von ihren Sensoren zu prozessieren und einen anderen Körper um sich in der Welt zu bewegen. Wir können nicht darauf schließen, dass die Komplexität des Pfades der Ameise auf die Komplexität der Mechanismen zurückzuführen ist, die ihr Verhalten steuern. Vielleicht hat die Ameise nicht viel mehr gemacht, als nach links auszuweichen, wenn sie ein Hindernis auf der rechten Seite wahrgenommen hat, und umgekehrt. Die Komplexität des Pfades ist das Resultat der Interaktion zwischen Ameise und ihrer Umgebung. Was würde passieren, wenn die Ameise um ein hundertfaches grösser wäre? Sie würde sich in einer geraden Linie über den Strand bewegen. Ihre Sensoren hätten gar nicht die Auflösung, um die Inhomogenitäten des Strandes wahr zu nehmen. Um das Verhalten der Ameise zu verstehen müssen wir also sowohl die Umgebung, wie auch interne Mechanismen und die Interaktion zwischen den beiden berücksichtigen.

Das Beispiel der Ameise lässt uns die drei wichtigsten Probleme, die wir in Betracht ziehen müssen, wenn wir über Verhalten sprechen, identifizieren. Auf dem Gebiet der *Artificial Intelligence* sind diese Probleme unter dem Namen *frame-of-reference problem* bekannt (Clancey 1991, Pfeifer und Scheier 1999).

1. *Perspective issue*: Beschreibungen von Verhalten aus der Perspektive eines Beobachters müssen von den internen Mechanismen, die zu diesem Verhalten führen, unterschieden werden. Um die internen Mechanismen zu beschreiben, muss die Perspektive des Agenten berücksichtigt werden.

2. *Behaviour-versus-mechanism issue*: Das Verhalten eines Agenten ist immer das Resultat von Interaktion mit der Umgebung und kann nicht alleine durch interne Mechanismen erklärt werden.
3. *Complexity issue*: Beobachtete Komplexität korrespondiert nicht immer mit der Komplexität der darunter liegenden Mechanismen.

Bewegung ist Teil des Verhaltens. Bewegungsdaten alleine können nur Hinweise auf Verhalten liefern, aus dem eine Änderung der Position eines Objekts resultiert. Wenn aus Bewegungsdaten Rückschlüsse auf interne Mechanismen des Objekts und auf Interaktionen mit der Umgebung gezogen werden sollen, ist Wissen über den Kontext der Bewegung nötig. Dies ist der Punkt, an dem auf *frame-of-reference* Probleme geachtet werden sollte.

### 2.3 Forschungsfelder und Forschungsfragen

In diesem Abschnitt werden die wichtigsten Forschungsgebiete, die sich mit Bewegungsdaten beschäftigen, diskutiert. Zuerst wird auf anwendungsorientierte Gebiete eingegangen, die sich vor allem mit spezifischen Fragestellungen an konkreten MO auseinandersetzen. Danach wird auf Gebiete eingegangen, die in erster Linie an allgemeinen Methoden forschen.

Die **Ökologie** ist ein Forschungsgebiet, das sich seit jeher mit der Bewegung von Organismen auseinandersetzt. Wie sich Organismen bewegen hat einen grossen Einfluss auf ihre Fitness, auf die Dynamik von Populationen, Gemeinschaften und Ökosystemen (Nathan et al. 2008: 19052). Nicht nur Tiere, auch Pflanzen sind, über einen genug grossen Zeitraum betrachtet, mobile Lebensformen (Laube 2005: 13). So kommt es, dass in der Ökologie eine ausgereifte, umfangreiche Methodik zur Erhebung und Analyse von Bewegungsdaten vorhanden ist (Turchin 1998).

Holyoak et al. (2008) führten eine gross angelegte Literaturstudie über Forschung zum Thema Bewegung in der Ökologie durch. Eine Schlüsselwortsuche brachte rund 26'000 Aufsätze hervor. Diese wurden zwischen 1997 und 2006 in 496 wissenschaftlichen Zeitschriften veröffentlicht. Davon wurden 1'000 zufällig gewählt. 73% dieser Artikel wurden als relevant für die Studie befunden. In 339 dieser Aufsätze war die Bewegung von Individuen ein zentrales Thema. Es ist klar, dass bei einer so umfangreichen Literatur zur Bewegung von Organismen hier nur ein verschwindend kleiner Teil berücksichtigt werden kann. Denn, wie Holyoak et al. (2008: 19060) sagen:

*„It is therefore a formidable task to identify papers that focus on organismal movement and contribute importantly to our understanding.“*

Welch gewaltige Arbeit in der Ökologie schon geleistet wurde, um das Bewegungsverhalten von bestimmten Arten zu verstehen, lässt sich am Beispiel der Wüstenameise *Cataglyphis* veranschaulichen. Das Navigationsverhalten dieses Insekts wird von Pfeifer und Bongard (2006: 42, 84-85, 105, 183) zur Illustration zahlreicher Konzepte diskutiert. Von aussen sieht man vom Nest dieser Ameisenart nur ein Loch im Boden. Der Lebensraum ist flach und karg, bietet also keine grossräumigen Merkmale, die zur Navigation verwendet werden könnten. Und dennoch findet *Cataglyphis* den direkten Weg zurück zum Nest, trotz Distanzen von bis zu 200 Metern, die bei der Nahrungssuche zurückgelegt wurden. In einer Serie von Experimenten, sowohl in freier Wildbahn, als auch in eigens präpariertem Umfeld im Labor, wurde das Navigationsverhalten dieser Ameisen studiert (Wehner und Räder 1979, Wehner und Srinivasan 1981, Müller und Wehner 1988, Wehner et al. 1996). Es scheint so, als würde *Cataglyphis* zwei Mechanismen für die Navigation verwenden. Anhand der Polarisierung, des in der Wüste meist wolkenlosen Himmels kann die Richtung der Bewegung abgeschätzt werden. Anhand des *optic flows*, also der Geschwindigkeit mit der sich Objekte im Gesichtsfeld bewegen, wird die Distanz bestimmt. Die Ameise weiss immer die ungefähre Richtung und Distanz zum Nest. Der zweite Mechanismus ist ein Vergleich von Mustern. Die Ameise macht sich beim Verlassen des Nests ein Bild des Horizonts. Wenn der erste Mechanismus sie in die Nähe des Nests geführt hat, wird dieses Bild mit dem derzeitigen Horizont verglichen. *Cataglyphis* bewegt sich nun, bis der Horizont, den sie sieht, mit dem Bild beim Verlassen des Nests übereinstimmt. Sobald dies der Fall ist, hat sie den Eingang zum Nest gefunden. Um dieses Modell zu testen, wurde neben zahlreichen Experimenten mit den Ameisen auch ein Roboter gebaut, der diese Navigationsstrategien verwendet. Der von Lambrinos et al. (1999) entwickelte Roboter wurde im natürlichen Habitat der *Cataglyphis* getestet. Er konnte über weite Strecken navigieren und zuverlässig zum Ausgangspunkt zurückkehren.

Dieses Beispiel veranschaulicht, dass es einzelne Organismen gibt, bei denen bestimmte Aspekte ihrer Bewegung, und der damit verbundene Kontext, bis ins Detail erforscht wurden.

Ein Gebiet, das sich mit der Bewegung von Menschen auseinandersetzt, ist die *time geography*. Die *time geography* (Hägerstrand 1970) ist ein Teil der Humangeographie, der sich mit den Einschränkungen von Individuen, in ihrer Möglichkeit den Raum zu nutzen, beschäftigt. Um solche Einschränkungen zu untersuchen werden Strassen, Verkehrsmittel, Hindernisse und Verpflichtungen berücksichtigt. Die tägliche Bewegung von jemandem, der von 8 bis 12 Uhr und von 13 bis 17 Uhr im Büro sein muss, und nebenbei einen Arbeitsweg von zwei Stunden hat, ist grossteils determiniert. In der *time geography* ist folglich der Kontext der Bewegung von grosser Bedeutung. Die realisierte Bewegung eines Individuums, in Form von Trajektoriendaten, ist aber nur selten im Fokus der Forschung.

Die Modellierung von *accessibility* und *daily potential path area* benötigt keine detaillierten Trajektorien. Die verwendeten Daten stammen meist aus Befragungen und nicht von Messungen der Bewegung. Der Mensch als Untersuchungsobjekt hat gegenüber anderen Organismen den Vorteil, dass man ihn über sein Verhalten im Raum befragen kann.

Die Konzepte *accessibility*, *daily potential path area* und *space-time-cube* lassen sich in GIS modellieren (Miller 1991). Durch die Verwendung von GIS und leistungsfähiger Computer hat auch die Analyse von grossen Datensätzen mit detaillierten Bewegungsdaten Einzug in die *time geography* gefunden. Während in der frühen *time geography* die Datensätze klein und die verfügbare Rechenkapazität gering war, sind Heute grosse Datensätze und viel Rechenleistung die Regel (Laube et al. 2005: 641).

Neben Ökologie und *time geography* sind viele andere Forschungsgebiete an Bewegungsdaten interessiert. Verkehr und Stadtplanung, Sicherheit und Überwachung, Militär, Sport, Medizin, um nur einige zu nennen.

***Moving object databases*** (MODB) sind Datenbanken, die speziell für Bewegungsdaten entwickelt wurden. Standard-Datenbanksysteme sind ungeeignet um ständig sich ändernde Daten, wie die Position von MO, zu verwalten. Dies ist der Fall, weil angenommen wird, dass Daten konstant sind, sofern sie nicht explizit geändert werden (Wolfson et al. 1998). MODB sind ein sehr aktives Forschungsgebiet, mit dutzenden wenn nicht gar hunderten von Publikationen in den letzten zehn Jahren. Ein wichtiges Ziel dieser Forschung ist es, die Positionen von MO in Echtzeit zu verwalten. Anwendungsgebiete, wie z.B. die Luftfahrt, Militär oder mobile Kommunikationssysteme, benötigen diese Verwaltung in Echtzeit (Laube 2005: 36). Es gibt aber auch MODB, die lediglich vergangene Positionen von MO als Trajektorien speichern (Frentzos et al. 2008). Die Forschung reicht von Datenmodellen über Abfragesprachen zu Implementierungsfragen wie effizientes Indexieren, Prozessieren von Abfragen und Optimierungstechniken (Frentzos et al. 2008: 151).

***Data mining*** ist der Prozess der automatisierten Entdeckung von nützlichen Informationen in grossen Datenquellen (Giannotti und Pedreschi 2008: 2). Wenn, wegen dem grossen Volumen der Daten, traditionelle Analysemethoden nicht eingesetzt werden können, kommt *data mining* zu Einsatz. Gemäss Gianotti und Pedreschi (2008) sind die drei wichtigsten Techniken:

*Predictive modelling*. Ein Klassifikationsmodell wird entwickelt. Dieses kann den Wert einer Variable oder die Klasse auf Grund von anderen Variablen vorhersagen. Das Modell lernt aus vergangenen Messungen und wird verwendet, um Vorhersagen zu neuen Messungen zu machen.

*Pattern discovery.* In den Daten wird nach Mustern gesucht, welche Korrelationen oder Assoziationen in den Daten beschreiben.

*Cluster analysis.* Das Ziel dieser Methode ist es den Datensatz in Gruppen von ähnlichen Daten aufzuteilen.

Gianotti und Pedreschi (2008) schlagen für das *mining* von Bewegungsdaten einen Ablauf vor. Dieser besteht aus der Rekonstruktion der Trajektorien aus Rohdaten, der Gewinnung von Wissen durch Methoden des *data minings* und der abschliessenden Aufbereitung des Wissens durch Interpretation und Erklärung.

Andrienko et al. (2008b: 35) diskutieren Beziehungen zwischen *data mining* Techniken (Fayyad et al. 1996, Miller und Han, 2001) und den Arten von Fragen, die in der MO Analyse gestellt werden. *Dependency analysis* hat das Ziel Regeln zu finden, aufgrund deren sich ein Attribut aus Werten andere Attribute vorhersagen lässt. *Dependency analysis* kann daher verwendet werden, um Abhängigkeit der Bewegung und von Charakteristik von Orten, Zeit und externen Phänomenen und Ereignissen zu finden.

Auch die **Visualisierung** von Bewegungsdaten ist ein aktives Forschungsgebiet. Für eine Übersicht über Herausforderungen und Methoden sei auf Andrienko et al. (2007, 2008a) verwiesen. Laube (2005: 38-41) verwendet vier Kategorien von Visualisierungen für Bewegungsdaten.

*Temporal maps and Cartograms:* Die einfachste Methode für die Visualisierung von MO ist, die komplette Trajektorie auf einer Karte aufzuzeichnen. Die Zeit kann, entsprechend dem *Space-Time-Cube* der *Time Geography*, als dritte Dimension verwendet werden (Abbildung 2.2). Graphische Variablen kön-

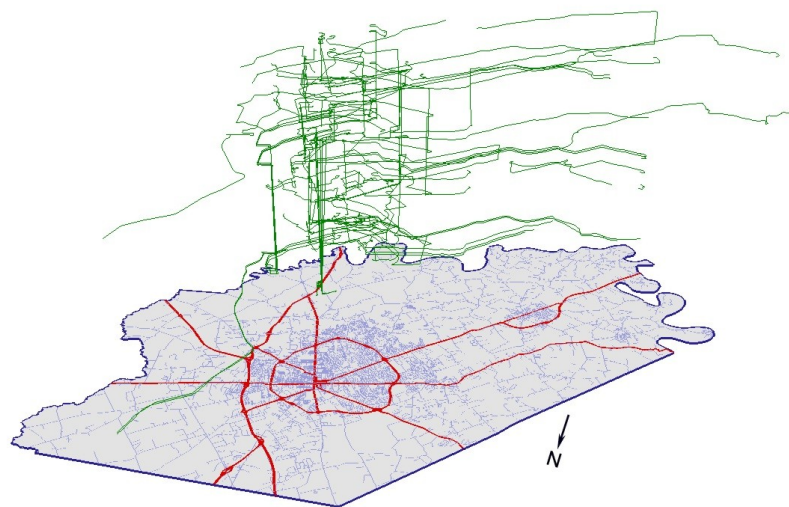


Abbildung 2.2: GPS Trajektorien im Space-Time-Cube (Kwan und Lee, 2004)

nen Geschwindigkeit, Beschleunigung oder Richtung codieren (Dykes und Mountain, 2003).

*Multiple Views*: Mehrere verschiedene Visualisierungen desselben Prozesses erlauben unterschiedliche Einsichten.

*Animations and dynamics*: Die Bewegung eines Punktobjekts eignet sich hervorragend zur Animation. Die Änderung mit der Zeit wird durch die Animation ersichtlich.

*Interactive maps*: Interaktion erlaubt es dem Benutzer einer Visualisierung diese selbst zu verändern. Z.B. kann er, wenn er etwas Interessantes in der Visualisierung sieht, diese so verändern, dass es besser ersichtlich wird.

Nebenbei ist anzumerken, dass eine Visualisierung von Trajektorien nicht zwingend den geographischen Raum abbilden muss. Zaho et al. (2008) z.B. argumentieren, dass in ihren Daten die Zeit und die Aktivitäten die wichtigsten Faktoren sind. Diese werden in einem kreisförmigen Diagramm visualisiert, das die Tageszeit auf den Kreis abbildet. Auch ist es nicht unbedingt nötig die Zeit in den Daten auf die Zeit in einer Animation abzubilden.

**Bewegungsmuster** (*movement patterns*) waren in jüngster Zeit vermehrt im Fokus der MO-Forschung. Ein Bewegungsmuster ist eine räumliche und zeitliche Regelmässigkeit oder eine interessante Beziehung in Bewegungsdaten (Dodge et al. 2008: 241). Muster in der Bewegung von Individuen oder Gruppen werden durch *data mining* Methoden des *pattern discovery* in den Bewegungsdaten gesucht. Laube und Imfeld (2002) definierten mehrere solche Muster und stellten Methoden vor, mit welchen diese Muster gefunden werden können. Seither wurden mehrere Bewegungsmuster genauer untersucht und die Algorithmen, mit denen die Muster gefunden werden, wurden verbessert. Gudmundson und Kreveld (2006) stellen einen Algorithmus vor, mit dem Bewegungen als Gruppen (*flocks*) gefunden werden können. Eine neuere Version dieses Algorithmus wird in Gudmundson et al. (2008) vorgestellt. Anderson et al. (2007) präsentieren ein Verfahren zur Detektion von *leadership patterns*. Buchin et al. (2008) stellen einen Algorithmus vor, der das Auffinden von *single file movement* ermöglicht.

Dodge et al. (2008) stellen fest, dass es wenige theoretische Grundlagen zu Bewegungsmustern gibt. Es ist nicht klar, welches die relevanten Muster sind und es gibt nur wenige Definitionen. Dodge et al. (2008) unterscheiden zwischen generischen Mustern, welche primitiv oder zusammengesetzt sein können, und Verhaltensmustern. Die oben erwähnten Bewegungen in Gruppen und das *leadership* Muster sind Verhaltensmuster. Zwei MO zur gleichen Zeit am gleichen Ort (*co-location in space*) sind ein Beispiel eines primitiven generischen Musters. Die Konvergenz mehrerer MO ist ein Beispiel für ein zusammenge-

setztes generisches Muster. Bewegungsmuster können auch rein zufällig auftreten. Laube und Purves (2006) schlagen vor, dass, von der Verteilung der Originaldaten ausgehend, mit einem *constraint random walk* Modell Bewegungsdaten simuliert werden. Um zu zeigen, dass Muster nicht zufällig sind, müssen diese in den Bewegungsdaten signifikant häufiger vorkommen als in den simulierten Daten.

Laube et al. (2008) zeigen, dass das Sammeln von Trajektorien Daten keine Voraussetzung für das Auffinden von Bewegungsmustern ist. *Wireless sensor nodes* haben die Eigenschaft, dass sie miteinander kommunizieren können. Laube et al. (2008) stellen einen Algorithmus vor, der es *wireless sensor nodes* ermöglicht, *flocks* zu erkennen, ohne dass dafür Positionsbestimmung durch GPS nötig wäre.

## 2.4 Kontext in der MO-Analyse

Der Kontext, in dem Bewegung stattfindet, ist nur in wenigen Arbeiten auf dem Gebiet der MO-Analyse ein Thema. Wenn über den Kontext der Bewegung geschrieben wird, ist das Thema meist nur angeschnitten. In der wenigen Literatur, die auf das Thema eingeht, lässt sich aber ein Konsens feststellen, dass das Bewegungsverhalten der Untersuchungsobjekte von der Umgebung, in der die Bewegung stattfindet, beeinflusst wird. Einige Autoren scheinen sich einig zu sein, dass die Frage nach dem Kontext der Bewegung eine grosse Herausforderung für die MO-Forschung ist (Laube et al., 2007, Spaccapietra et al., 2008, Andrienko et al., 2008b, Gudmundson et al., 2008, Nathan et al., 2008).

### 2.4.1 Ist Kontext wichtig?

Laube et al. (2007: 498) sagen, dass die Untersuchung von Zusammenhängen zwischen Eigenschaften von Trajektorien, der geographischen Umgebung, sowie den räumlichen Prozessen, in die das Individuum involviert ist, ein zentraler Untersuchungsgegenstand der MO-Analyse sein sollte. Gudmundsson et al. (2008: 82) stellen fest, dass Bewegungsmuster in Bezug zur umgebenden Geographie gesetzt werden müssen, um zu verstehen wo, wann und warum sich Objekte bewegen. Als Beispiele nennen sie die Vegetationsabhängigkeit gewisser Bewegungsmuster grasender Schafe. Auch Bewegungsmuster von Tauben, die zu ihrem Schlag zurückkehren, die im Zusammenhang mit linearen Landschaftselementen wie Flüssen oder Strassen auftreten, stehen im Bezug zur umgebenden Geographie. Gudmundsson et al. (2008: 82) fordern, dass Muster entwickelt werden, die es erlauben Bewegung mit der Umgebung, in der sie stattfindet, zu verknüpfen. Huisman und Forrer (2005) finden, dass man beachten sollte, dass Bewegungsmuster das Resultat sind von Individuen, die in ihrem Umfeld handeln, als Reaktion auf Stimuli und basierend auf ihren Präferenzen und ihrem Wissen. Eine ähnliche Aussage machen Dykes und Mountain (2003: 590) und nennen auch einige der Stimuli und Faktoren:

„[...] *Our continual operation under the constraints of time, money, need, right of way, knowledge, experience, fear, safety, transportation networks, transport services and a range of other factors that influence our daily lives and are also likely to shape our spatio-temporal behaviour.*“

Laube (2005: 42) unterscheidet zwei verschiedene geographische Perspektiven bei der Analyse von Bewegung. Bei der MO-Analyse im eigenschaftslosen Raum liegt der Fokus der Analyse auf der Bewegung und ihren geometrischen und topologischen Eigenschaften. Bei der MO-Analyse mit Bezug zur Umgebung wird versucht, Eigenschaften der Umgebung zu finden, die die Bewegung beeinflussen.

Auch Andrienko et al. (2008b: 16) beziehen Stellung für die Integration von Kontext. Die Umgebung, in der die Bewegung stattfindet, und die Eigenschaften der sich bewegenden Entitäten können einen wichtigen Einfluss auf die Bewegung selbst haben. Deshalb sollten sie berücksichtigt werden, wann immer Bewegung untersucht wird. Des Weiteren sei die Bewegung selbst nicht immer im Fokus der Untersuchung. Das Ziel einer Untersuchung von Bewegungsdaten kann sein, Wissen über die sich bewegenden Entitäten zu gewinnen, oder über die Umgebung, in der die Bewegung stattfindet. Gemäss Andrienko et al. (2008b: 20) haben die meisten Forscher, erkannt, dass Bewegungsanalyse nicht nur Trajektorien, sondern auch deren raumzeitlichen Kontext untersuchen sollte.

### 2.4.2 Der *Places of Interest* Ansatz

In der Datenbank-Forschungsgemeinschaft wurde lange Zeit wenig Interesse an der Sichtweise der Anwender und spezifischer Anwendungen gezeigt (Spaccapietra et al., 2008). Prototypen für räumlich-zeitliche Datenbanksysteme existieren, aber die Integration von Trajektorien mit relevanter geographischer Information obliegt den Benutzern (Alvares et al., 2007b). Der Schwerpunkt der Forschung an Bewegungsdatenbanken liegt auf dem Gebiet der effizienten Speicherung und Abfrage von Trajektorien. Arbeiten über die Modellierung von Bewegungen in Netzwerken (Güting et al., 2006), und insbesondere solche über die Anreicherung von Bewegungsdaten mit relevanter geographischer Information (Alvares et al., 2007a, Alvares et al., 2007b, Spaccapietra et al., 2008, Borogny und Wachowicz, 2008, Gómez et al., 2009) zeigen, dass ein Umdenken stattfindet.

Spaccapietra et al. (2008) präsentieren zwei Modelle für Bewegungsdaten, die eine Anreicherung mit semantischer Information unterstützen. Sie definieren Trajektorien als einen benutzerdefinierten *Record* der Entwicklung der als Punkt wahrgenommenen Position eines Objekts, das sich während einem gegebenen Zeitintervall im Raum bewegt um ein Ziel zu erreichen (Spaccapietra et al. 2008: 130). Trajektorien sind also Bewegungen, die für eine im Kontext der Untersuchung bedeutungsvolle Reise stehen, die einen Ausgangspunkt und ein Ziel hat. Bewegungsdaten werden in solche semantisch bedeutungsvolle

Trajektorien aufgeteilt. Ein Modell für solche Trajektorien muss also die Festlegung von Start- und Endpunkten der Trajektorien erlauben. Daneben erlaubt das Datenmodell auch die Aufteilung in weitere Unterkomponenten, das Definieren von Attributen, semantische und topologische Einschränkungen (z. B. Störche fliegen nicht in der Nacht, ein Auto muss sich auf einem Strassennetzwerk bewegen) und Verknüpfungen mit anderen Objekten der Anwendung.

Auf dem in Spaccapietra et al. (2008) publizierten Modell aufbauend, präsentieren Alvares et al. (2007b) ein Modell zur Datenvorprozessierung, das es erlaubt, Trajektorien mit semantischer Information aus dem jeweiligen Anwendungsgebiet anzureichern. Sie argumentieren, dass diese semantische Information Abfrage, Analyse und *mining* von Bewegungsdaten erleichtern kann. Weil räumliche Verknüpfungen rechenintensiv sind ist es effizienter die Trajektorien mit der gewünschten geographischen Information in einer Vorprozessierung zu verknüpfen, als bei jeder Abfrage die benötigten Verknüpfungen neu zu berechnen. Der Algorithmus von Alvares et al. (2007b) unterteilt Trajektorien in *stops* und *moves*. *Stops* stehen für Teile der Trajektorien die innerhalb oder in der Nähe von interessanten räumlichen Objekten, sogenannten *places of interest*, liegen. Sie haben einen Beginn, ein Ende und ein assoziiertes räumliches Objekt. *Moves* sind Teile der Trajektorien, die von einem *stop* zum nächsten führen. Das Modell wird anhand eines Beispiels von Konferenzbesuchern illustriert, die sich zwischen *stops* wie Hotels, touristischen Attraktionen und einem Konferenzzentrum hin und her bewegen. Alvares et al. (2007b) denken, dass ein solches Zusammenfügen von Bewegungsdaten mit geographischer Hintergrundinformation das Entdecken von semantischen Mustern erlaubt, die durch Methoden, die nur die geometrischen Eigenschaften von Trajektorien berücksichtigen, nicht gefunden werden könnten.

Die vorgestellten Modelle für die Anreicherung von Bewegungsdaten mit semantischer Information wurden als Ausgangspunkt für weitere Arbeiten verwendet. *Stops* und *moves* wurden verwendet, um ein Lernspiel mit GPS-fähigen Mobiltelefonen und um Verkehrsstaus zu analysieren (Borogny und Wachowicz, 2008). Eine Abfragesprache für in *stops* und *moves* eingeteilte Trajektorien wurde entwickelt (Gómez et al., 2008).

Einen ähnlichen Ansatz haben auch du Mouza und Rigaux (2004, 2005) präsentiert. Sie verwenden Einteilungen des Raumes, wie z.B. administrative Einheiten oder Nutzungsklassen. Jede Trajektorie wird zu einer Sequenz von Regionen, die das MO nacheinander traversiert hat. Danach können Muster in diesen Sequenzen gesucht werden. So kann man z.B. an den Trajektorien interessiert sein, die von einem Wohnquartier in ein Industriequartier und wieder zurück führen.

### 2.4.3 Klassifikationen

Andrienko und Andrienko (2007) präsentieren eine Liste von Faktoren, die Bewegung und Verhalten von Entitäten beeinflussen können. Diese Liste wird in Andrienko et al. (2008b) eingehend diskutiert. Eine ähnliche Liste ist in Dodge et al. (2008) zu finden.

Andrienko und Andrienko (2007: 122-123) teilen Variablen, die das Bewegungsverhalten beeinflussen können, in vier Kategorien ein (Tabelle 3.2). Die Kategorien sind Eigenschaften des Raumes, Eigenschaften der Zeit, Eigenschaften und Aktivitäten der sich bewegenden Objekte sowie verschiedene räumlich-zeitliche Phänomene. Unter Eigenschaften des Raumes zählen sie Charakteristik von Terrain und Oberfläche, Erreichbarkeit eines Ortes, an einem Ort vorhandene Objekte, und die Bedeutung, die ein Ort für eine Entität hat. Eigenschaften der Zeit sind Zyklen, damit verbundene physikalische Charakteristiken (z.B. das Vorhandensein von Tageslicht) und die zeitlich unterschiedliche Nutzung eines Ortes. Eigenschaften und Aktivitäten von sich bewegenden Entitäten sind individuelle Eigenschaften (z.B. Alter), Fortbewegungsweise, Ziel und Ursache der Bewegung und Aktivitäten, die während der Bewegung ausgeführt werden. Zu den räumlich-zeitlichen Phänomenen werden Wetter und Klima, kulturelle Ereignisse, Regulationen und Bräuche, sowie auch Preise gezählt.

Auch Dodge et al. (2008) nennen vier Kategorien von Einflüssen. Die erste Kategorie sind die intrinsischen Eigenschaften des sich bewegenden Objekts, wie z.B. die maximale Geschwindigkeit, die erreicht werden kann, oder das Beschleunigungsverhalten. Die zweite Kategorie sind räumliche Einschränkungen für die Bewegung wie Barrieren oder ein Netzwerk in dem die Bewegung stattfindet. Die Umgebung in der die Bewegung stattfindet und die Objekte enthalten kann, die auf das sich bewegende Objekt eine anziehende oder abstossende Wirkung haben können, ist die dritte Kategorie. Die vierte Kategorie widmet sich dem Einfluss, den andere bewegte Entitäten auf das Untersuchungsobjekt haben können. Dodge et al. (2008) kritisieren an der Einteilung von Andrienko und Andrienko (2007), dass es keine Kategorie für den Einfluss bewegter Entitäten aufeinander gibt, und dass ein solcher Einfluss auch sonst nicht explizit erwähnt wird.

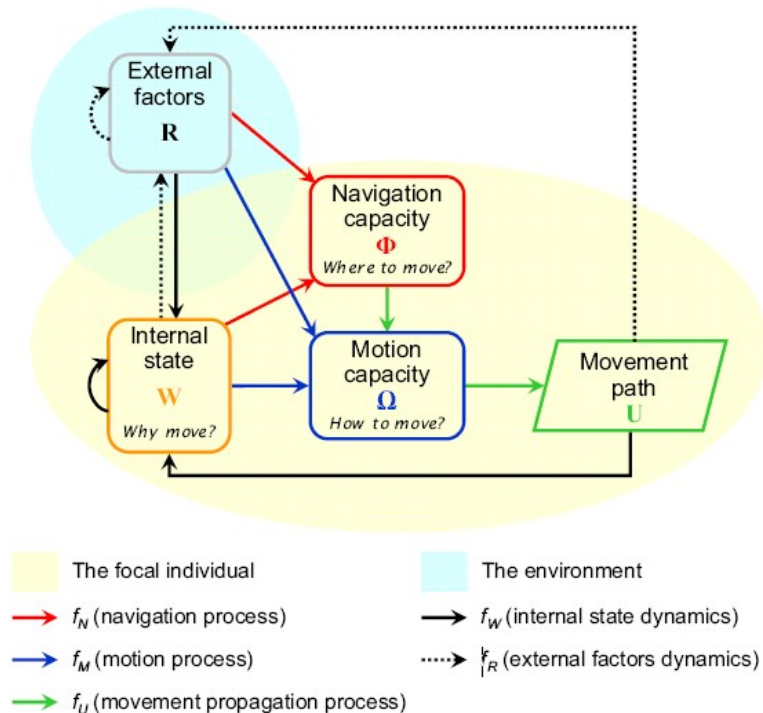


Abbildung 2.3: Bewegungsökologisches Modell von Nathan et al. (2008)

#### 2.4.4 Ein *Movement Ecology Framework*

Das von Nathan et al. (2008) vorgestellte konzeptuelle Modell für die Bewegungsökologie hat fünf Bestandteile (Abbildung 2.3). Der sich bewegende Organismus wird durch seinen internen Zustand, seine Kapazität sich zu bewegen und seine Kapazität zur Navigation modelliert. Diese werden von externen Faktoren beeinflusst. Aus diesen vier sich gegenseitig beeinflussenden Komponenten resultiert als fünfte Komponente der Pfad der Bewegung. Dazu kommen die externen Faktoren als fünfter Bestandteil des Modells.

**Externe Faktoren** repräsentieren im Modell alle Aspekte des biotischen und abiotischen Umfelds, die die Bewegung beeinflussen. Nathan et al. (2008) gehen nicht im Detail darauf ein, was externe Faktoren sein können, da dies von Organismus zu Organismus variiert. Externe Faktoren ändern sich, wenn der Organismus sich im Raum bewegt. Welche Faktoren relevant sind hängt vom internen Zustand des Organismus ab. Da die Umgebung des Organismus sich auch verändern kann, wenn sich der Organismus nicht bewegt, können sich die externen Faktoren auch von selbst verändern. Externe Faktoren können, gemäss dem Bezugssystem, sowohl den internen Zustand, als auch Navigations- und Bewegungskapazität beeinflussen.

Die **Bewegungskapazität** repräsentiert die Möglichkeiten, die ein Organismus besitzt um sich fort zu bewegen. Viele Organismen haben mehrere Bewegungsarten, deren sie sich bedienen können. Eine Tauchente z.B. kann fliegen, gehen, schwimmen und tauchen. Da

diese Bewegungsarten gewisse physikalische Gesetzmässigkeiten ausnützen stellen sie immer Ansprüche an die Beschaffenheit der Umgebung. Ein Organismus hat also eine oder mehrere Bewegungsarten, die jeweils spezifische Eigenschaften in Bezug auf Geschwindigkeit und Beschleunigung haben, sowie Ansprüche an die Beschaffenheit der Erdoberfläche stellen.

Die **Navigationskapazität** repräsentiert die Fähigkeit eines Organismus sich in Raum und Zeit zu orientieren und zu entscheiden, wann und wohin er sich bewegt. Externe Faktoren wie das Vorhandensein von Licht oder visuelle Hinweise können diese Fähigkeit beeinflussen.

Der **innere Zustand** eines Organismus wird als multidimensional beschrieben. Er hat in erster Linie mit Zielen und mit Motivation zu tun. Sich zu bewegen kann einem Organismus viele Vorteile bringen, von denen einige sofort offensichtlich sind, und andere vielleicht erst unter Einbezug evolutionärer Faktoren erklärbar sind.

Aus dem Zusammenwirken von innerem Zustand, Navigationskapazität, Bewegungskapazität und externen Faktoren kommt der realisierte **Pfad der Bewegung** zustande. Dieser Pfad kann durch Positionsmessungen am MO erfasst werden. Wenn sich der Organismus bewegt, ändert sich sein aktuelles Umfeld, und somit auch die externen Faktoren. Aus ihrem Modell leiten Nathan et al. (2008) fünf grundlegende Fragen ab, die über Bewegung gestellt werden können:

- (1) Was ist die Motivation für die Bewegung, was sind die inneren Ziele?
- (2) Wie wird die Bewegung ausgeführt?
- (3) Wann und in Richtung welches Zieles wird die Bewegung ausgeführt?
- (4) Welche externen Faktoren beeinflussen die Bewegung?
- (5) Wie interagieren die vier Komponenten um die Bewegung zu produzieren?

### 2.4.5 Weitere Aspekte

In diesem Abschnitt werden Arbeiten diskutiert, die auf Kontext eingehen, aber keinem der Themengebieten der vorhergehenden Abschnitte entsprechen.

Dykes und Mountain (2003) benutzen ein Beispiel von Touristen, die sich auf einer Inselgruppe bewegen, um zu illustrieren, was physische Einschränkungen für einen Einfluss auf Trajektorien haben können. Bewegung an Land folgt oft den Strassen, die wiederum den Küstenlinien folgen. Bewegung zwischen den Inseln auf Schiffen führt geradlinig von einem Hafen zum anderen. Dykes und Mountain (2003) machen sich auch Gedanken darüber, welchen Einfluss die Art der Fortbewegung auf die Trajektorie hat. Bewegung zu Fuss hat eine geringe Reichweite und eine tiefe Geschwindigkeit, bietet aber auch eine grosse Bewegungsfreiheit. Motorisierte Bewegung hat eine hohe Geschwindigkeit und

Reichweite, ist aber an Netzwerke (im Falle eines Automobils an die Strassen) gebunden. Bei einem Linienflug ist die Geschwindigkeit und Reichweite noch grösser, das Individuum hat aber keinerlei Einfluss auf die Route.

Brinkhoff (2002) diskutiert die Einflüsse die auf MO wirken, die sich mit Fahrzeugen fortbewegen. Solche MO bewegen sich in der Regel in Netzwerken. Automobile folgen Strassen, Züge folgen Schienen und Flugzeuge folgen Korridoren. Eigenschaften des Netzwerkes und das Verkehrsaufkommen beeinflussen die Geschwindigkeit eines MO. Ereignisse wie *rush hour* oder Wetterphänomene haben auch einen Einfluss auf das MO. Zusätzlich gibt es MO, die einen festen Fahrplan haben, dem sie folgen müssen.

Imfeld (2000) präsentiert die *radial distance function*. Diese Methode kann verwendet werden, um die Umgebung eines MO im Verlaufe der Bewegung zu analysieren. In Abschnitt 5.1.1 wird näher darauf eingegangen.

Das *9-intersection model* definiert die möglichen topologischen Relationen zwischen zwei statischen Geometrien im zweidimensionalen Raum (Egenhofer und Herring, 1991). Aufbauend auf der *qualitative theory of movement* von Galton (1995) wurde ein Modell für die topologischen Beziehungen zwischen sich bewegenden Punkten entwickelt (Van de Weghe et al. 2005, Van de Weghe et al. 2006). Im *9-intersection model* sind nur zwei Beziehungen zwischen Punkten möglich. Entweder sind sie gleich (*equal*) oder sie sind nicht gleich (*disjoint*). Der *qualitative trajectory calculus* (Van de Weghe et al. 2005) erlaubt die Spezifikation einer Vielzahl von Relationen, indem die Relativbewegung der Punkte zueinander in Betracht gezogen wird. Statt zwischen zwei kann so zwischen 81 verschiedenen Relationen unterschieden werden. In Abschnitt 2.4.3 wurde die Klassifikation für beeinflussende Faktoren der Bewegung von Dodge et al. (2008) vorgestellt. Diese weist eine Kategorie nur für beeinflussende Faktoren zwischen MO aus. Für die Analyse solcher Einflüsse könnte ein Modell, das die topologischen Relationen zwischen MO beschreiben kann, von grossem Nutzen sein.

Einen ähnlichen Ansatz verfolgen Noyon et al. (2007). Bewegung sollte ihrer Meinung nach aus der Perspektive des MO modelliert werden. Der von ihnen präsentierte Ansatz konzentriert sich daher auf relative Positionen und relative Geschwindigkeiten zwischen MO. Im Gegensatz zum Modell von Van de Weghe et al. (2005, 2006), wird das Modell von Noyon et al. (2007) auch auf bewegte Linien und Polygone erweitert.

Laube et al. (2007) prägen den Begriff *context operators* für beschreibende Variablen von Trajektorien. Dabei wird von den Operationsklassen der *map algebra* (Tomlin, 1990) ausgegangen. Sie definieren vier Klassen (Abbildung 2.4), entsprechen der Klassen *local*, *focal*, *global* und *zonal* der *map algebra*. *Instantaneous* entspricht *local*; die beschreibende Variable wird für einen einzigen Moment bestimmt. *Interval* entspricht *zonal*; hier werden alle Werte in einem sich bewegenden Intervall verwendet. *Episodal* entspricht *zonal*; die Variable wird für eine abgegrenzte Episode der Trajektorie berechnet. *Total* entspricht *glo-*

*bal*; bei einer solchen Operation wird ein einziger Wert für die gesamte Trajektorie berechnet. Die Geschwindigkeit kann *instantaneous* berechnet werden. Die Berechnung der Sinuosität einer Trajektorie hingegen braucht mindestens einen *interval context operator*. Der Kontextbegriff wird alleine auf die Trajektorie bezogen verwendet. Die verschiedenen *context operators* verwenden einen unterschiedlich grossen Teil der Trajektorie für die Berechnung einer Variable, die die Trajektorie beschreibt.

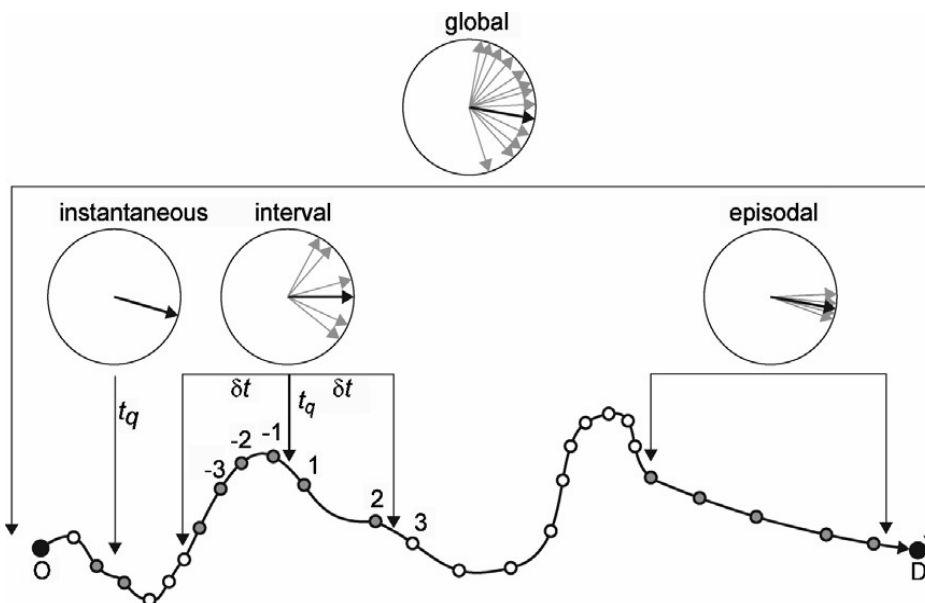


Abbildung 2.4: Die vier Perspektiven für beschreibende Variablen von Trajektorien (Laube et al. 2007)

Die meisten von Laube et al. (2007) genannten Beispiele für *context operators* (Geschwindigkeit, Beschleunigung, Sinuosität, usw.) verwenden jeweils nur  $(t, x, y)$  und keine zusätzlichen Informationen. *Navigational displacement* und *approaching rate* verwenden Distanz und Entfernung zu einem Objekt im Umfeld der Bewegung. Es wäre aber durchaus denkbar, dass auch andere Attribute der Bewegungsdaten verwendet würden. Diese Attribute könnten den räumlichen Kontext, in dem die Bewegung stattfindet, repräsentieren. So kann z.B. die Distanz zu einer Nahrungsquelle für jeden *fix* der Trajektorie bestimmt werden, aber auch für ein Intervall, für einen Abschnitt oder für den ganzen Pfad der Bewegung.

Forscher, die sich mit Karten auf mobilen Geräten (PDA, GPS-Geräte, Handy,...) auseinandersetzen, haben erkannt, dass Informationen über den Kontext, in dem sich der Benutzer befindet, gebraucht werden können, um die Karten der Situation anzupassen und damit zu verbessern. Kontext wird in diesem Gebiet immer über die Anwendung (in diesem Fall die mobile Karte) definiert. Nach Chen und Kotz (2000) ist Kontext die Menge der Zustände und Konfigurationen der Umgebung, die entweder das Verhalten einer Applikation bestimmt oder in der ein Ereignis in der Applikation eintritt.

Sarjakoski und Nivala (2005) nennen neun Klassen des Benutzungskontexts mobiler Applikationen. Vier davon sind Systemeigenschaften, Nutzungszweck, Eigenschaften des Nutzers und der kulturelle und soziale Kontext. Rein räumliche Kontext-Klassen sind die Position, die Orientierung und die physische Umgebung. Die Zeit ist eine temporale Kontext-Klasse. Der räumlich-zeitliche Kontext ist die *navigation history*, also der bereits zurückgelegte Pfad. Mountain (2005) entwickelte geographische Filter, die Informationen über die Bewegung des Nutzers verwenden. Diese Information wird verwendet, um die Antwort auf Abfragen der räumlichen Situation des Nutzers anzupassen. Bei mobilen Applikationen, die den Benutzungskontext berücksichtigen, ist also die Bewegung ein Teil des Benutzungskontexts. In dieser Arbeit ist die Bewegung selbst im Mittelpunkt des Interesses. Das Benutzen einer mobilen Anwendung ist von dieser Perspektive aus ein Teil des Kontexts der Bewegung. Es ist anzunehmen, dass die Verwendung eines solchen Gerätes die Bewegung beeinflusst.

Auch in Arbeiten auf dem Gebiet der *time geography* taucht der Begriff Kontext auf. Weber und Kwan (2003) stellen die Frage, inwiefern der geographische Kontext in urbaner Umgebung die *accessibility* eines Individuums beeinflusst. Hierzu definieren sie fünf Variablen, die den Kontext repräsentieren. Dies sind:

- (1) Anzahl der Meilen Strassen in der *daily potential path area* (PPA).
- (2) Anzahl der Gelegenheiten in der *daily* PPA (Arbeit, Einkauf, Unterhaltung, ...).
- (3) Wichtigkeit und Attraktivität der Gelegenheiten.
- (4) Gebäudehöhe im Verhältnis zur Grösse der Parzellen.
- (5) Vorhandensein von Gelegenheiten im Verlauf der Zeit.

Eine schrittweise multivariate Regression zeigte, dass diese Kontextvariablen nicht von Bedeutung sind. Weber und Kwan (2003) liefern keine Erklärung, was genau sie unter geographischem Kontext verstehen. Deshalb wird auch nicht klar, wieso die fünf gewählten Variablen eine geeignete Operationalisierung des urbanen geographischen Kontexts sein sollen.

## 2.5 Schlussfolgerungen

Das *random walk* Paradigma ist das vorherrschende Paradigma in der Bewegungsökologie. Wer *random walk* Modelle verwendet nimmt an, dass Bewegung weitgehend zufällig ist. Dies bedeutet auch, dass die Umgebung keinerlei Einfluss auf die Bewegung hat. Wenn die Forschung von einem solchen Paradigma geleitet wird, hat die Erforschung von Relationen zwischen Kontext und Bewegung geringe Priorität. Die Definition des *random walks* von Turchin (1998) beinhaltet Kontextfaktoren, diese sind aber nicht in die mathe-

matischen Modelle integriert. Es ist, wie von Nathan et al. (2008) festgestellt wurde, an der Zeit, weitere Faktoren in die *random walk* Modelle zu integrieren. Damit könnten diese um Aspekte des Kontexts erweitert werden.

Die MO-Forschung scheint in gewisser Hinsicht zweigeteilt zu sein. Auf der einen Seite sind die angewandten Gebiete, die sich vor allem mit konkreten MO und deren Bewegung auseinandersetzen. Holyoak et al. (2008) stellen fest, dass es nur wenige Artikel in der bewegungsökologischen Literatur gibt, die sich mit allgemeinen Methoden beschäftigen. In der *time geography* gibt es sowohl Forschung am Objekt wie auch Arbeiten zu allgemeinen Methoden. Auf diesem Gebiet ist die Verwendung von hoch aufgelösten Trajektorien-daten aber noch relativ neu. Die angewandten Gebiete setzen sich oft mit dem Kontext der Bewegung auseinander, verfügen aber über wenige allgemeingültige Methoden oder Theorien. In den angewandten Gebieten ist bei den Analytikern meist ein umfangreiches Wissen über das Untersuchungsobjekt vorhanden. Dies könnte ein Grund sein, wieso es wenige Theorien über den Kontext der Bewegung gibt. Ein Ornithologe kennt das Verhalten von Zugvögeln. Wenn ein Vogel während der Migration eine Pause einlegt kann er das erklären. Der Zugvogel hat in einem nahrungsreichen Gebiet eine Pause eingelegt, um Energie zu sammeln. Der Ornithologe kann das Muster dank seinem Wissen erklären, ohne dafür eine Theorie über den Kontext, in dem Bewegung stattfindet, zu benötigen.

Andererseits gibt es die Gebiete, die sich mit allgemeinen Methoden beschäftigen. Forscher dieser Gebiete legen in der Regel Wert auf eine möglichst grosse Allgemeingültigkeit ihrer Methoden. Das heisst, dass man sich mit dem kleinstmöglichen gemeinsamen Nenner von Bewegungsdaten auseinandersetzt, die  $(t, x, y)$ -Form der Bewegungsdaten. Jegliche Art von Bewegung von MO kann so modelliert werden. Sobald weitere Attribute hinzukommen, würden diese von aber Fall zu Fall verschieden sein. Um die Methoden allgemein zu halten wird daher angenommen, dass der Raum eigenschaftslos ist, und es wird alleine die Bewegung analysiert.

Abschnitt 2.4 zeigt, dass ein Umdenken stattfindet. Methodisch orientierte Forscher haben mehrfach auf die Wichtigkeit von Kontext hingewiesen. Es wurde erkannt, dass in diesem Zusammenhang grosser Forschungsbedarf besteht. Erste Anstrengungen wurden unternommen um einige der bestehenden Lücken zu füllen. Die Klassifikationen von Dodge et al. (2008) und Andrienko und Andrienko (2007) geben Hinweise, worauf man alles achten sollte. Der *qualitative trajectory calculus* (Van de Weghe et al. 2005, 2006) könnte sich als hilfreich herausstellen um Relationen zwischen sich beeinflussenden MO festzustellen. Auch einige Datenbankforscher setzen sich seit kurzem mit dem Kontext der Bewegung auseinander. Für den *places of interest* Ansatz müssen die wichtigen Orte für die MO bekannt sein. Der Fokus liegt auf der Bewegung von Ort zu Ort. Wie diese Bewegung stattfindet, welche Muster dabei auftreten, ist nicht von Bedeutung. Der *places of interest* Ansatz ist nicht anwendbar, wenn die Bewegung selbst als dynamischer Prozess analysiert

werden soll. Genauso scheitert er, wenn nicht bekannt ist, welche Orte für die MO wichtig sind. Von den vielen möglichen Fragen über den Kontext der Bewegung kann mit *places of interest* Methoden nur ein kleiner Teil beantwortet werden.

Der wohl vielversprechendste Beitrag zum Verständnis von Kontext kommt von Seite der eher anwendungsorientierten Bewegungsökologie. Das *movement ecology framework* von Nathan et al. (2008) modelliert die MO als miteinander interagierende Komponenten, die von externen Faktoren, also vom Kontext, beeinflusst werden. Ein Forscher soll sich Gedanken darüber machen, wie sein Objekt sich bewegt, wie es Navigiert und was seine Motivation ist. Denn wenn dies bekannt ist, ist auch klar, wie das Objekt durch seine Umgebung beeinflusst wird. Für die Navigation muss die Umwelt wahrgenommen werden. Welche Sinne hat das MO, mit denen es die Umwelt wahrnehmen kann? Um sich fortzubewegen, muss das MO mit seiner physischen Umwelt interagieren. Welche Fortbewegungsarten kann ein MO nutzen und was stellen diese für Ansprüche an die Umwelt? Im Gegensatz zu den in Abschnitt 2.4.3 vorgestellten Klassifikationen ist das Modell prozessorientiert. Das *movement ecology framework* hilft uns, Bewegung als Prozess zu verstehen, der in den geographischen Raum eingebettet ist und durch diesen beeinflusst wird.

## 2.6 Position der Geographie

Die Positionen der *time geography*, die sich schon seit den 1970er Jahren mit Bewegung auseinandersetzen, sind bezogen. Welche Beiträge können die geographischen Informationswissenschaften (GIScience) auf dem für sie relativ neuen Gebiet der Analyse von Bewegung liefern?

Fragen nach dem 'Warum?' der Bewegung gehören nicht zum Aufgabengebiet der GIScience. In Zusammenarbeit mit Anwendern, die das nötige Fachwissen über die Untersuchungsobjekte haben, können aber Methoden entwickelt werden, die helfen solche Fragen zu beantworten.

Eine Stärke der Geographie ist es, räumliche Zusammenhänge zu finden. Die GIScience kann daher Methoden entwickeln, die Beziehungen zwischen Mustern und Eigenschaften der Bewegung und ihrem räumlichen Kontext finden können. Solche Methoden würden sich auf externe Faktoren und den Pfad der Bewegung im Modell von Nathan et al. konzentrieren. Aus solchen Zusammenhängen mehr über Navigations-, Bewegungskapazität und inneren Zustand der MO herauszufinden, ist dann wiederum Aufgabe der anwendungsorientierten Wissenschaften. Diese haben das nötige Wissen über die untersuchten MO.

Eine weitere Aufgabe der Geographie kann sein, Daten über das Umfeld, in dem die Bewegung stattfindet, zusammenzutragen. Inwiefern können schon vorhandene Datensätze verwendet werden? Datensätze über Infrastruktur und Oberflächenbedeckung sind für

## 2 Grundlagen

---

grosse Teile der Erdoberfläche erhältlich. Auch digitale Geländemodelle gibt es für viele Gebiete. Viele andere Daten, z.B. über das Wetter, werden für diverse Zwecke gesammelt. Eine Aufgabe der Geographie und GIScience könnte sein diese Daten zusammenzutragen und in eine Form zu bringen, in der sie zusammen mit Trajektorien Daten verwendet werden können.

Die Geographie kann auf dem interdisziplinären Gebiet der MO-Forschung eine Vermittlerrolle einnehmen. Als Disziplin, die sich mit konkreten MO, aber auch mit allgemeinen Methoden beschäftigt, ist sie an der Schnittstelle zwischen Anwendung und Theorie positioniert.

Zwei grosse Herausforderungen, denen sich die GIScience stellen muss, sind die Erarbeitung von Datenmodellen für Kontext und die Formulierung von räumlichen Relationen zwischen MO und ihrem Kontext. Diesen Fragen werden wir in den nächsten Kapiteln nachgehen.

# 3 Was ist Kontext?

Kapitel 2 hat klar gemacht, dass Kontext in der Analyse von Bewegungsdaten eine Rolle spielen sollte. In diesem Kapitel wird versucht den Begriff Kontext weiter zu präzisieren. Zuerst wird zwischen verschiedenen Arten des Kontexts unterschieden. Danach werden Strategien diskutiert, die angewandt werden können, um Kontext in die Analyse von Trajektorien zu bringen. Abschliessend werden die bisher vorgestellten Klassifikationen und Modelle miteinander verknüpft, um mehr über den Kontext-Begriff zu erfahren.

## 3.1 Zwei Typen von Kontext

Weil Kontext ein Begriff ist, der schwierig zu definieren ist und vielerlei Dinge bedeuten kann, sollte damit vorsichtig umgegangen werden. Eine klare Trennung zwischen dem Kontext, in dem die Analyse stattfindet, und dem Kontext der Bewegung ist wünschenswert. Dieser Abschnitt beschreibt den Analyse- und den Bewegungskontext und zeigt auf, wie die beiden interagieren.

Dass eine solche Präzisierung und Einschränkung des Kontext-Begriffs sinnvoll und nützlich ist, zeigt der in Abschnitt 2.4.5 vorgestellte Begriff des *usage context*. Der Einbezug des räumlichen und nicht-räumlichen Kontexts des Benutzers erlauben es Programmen, die auf mobilen Geräten laufen, auf Abfragen Resultate zu liefern, die der Situation angepasst sind.

### 3.1.1 Analysekontext

Der Analysekontext bezieht sich allein auf die Analyse, nicht aber auf die Bewegung. Die Ziele der Analyse, die Fragen, die beantwortet werden sollten und die Methoden, die dafür zur Verfügung stehen, sind alles Komponenten des Analysekontexts. Die Forschungsfragen, die gestellt werden und die Methoden, die verwendet werden, legen fest, inwiefern der Bewegungskontext in die Analyse integriert werden soll oder integriert werden kann. Wenn die Analyse von Bewegungsdaten mit dem Ziel erfolgt, festzustellen wann und wo sich Gruppen von MO bilden, reicht die in den Rohdaten vorhandene Information vollkommen aus. Ganz anders die Frage, auf welchem Terrain sich eine bestimmte Tierart bei starken Winden bevorzugt fortbewegt. Diese Frage lässt sich nur unter Einbezug von Terrain- und Wetterdaten beantworten, und dies auch nur, wenn entsprechende Methoden zu deren Integration und anschliessender Analyse vorhanden sind.

Auch die Form der Daten ist Teil des Analysekontexts. Wie gross ist die mit den gemessenen Positionen verbundene Unsicherheit? Ist die temporale Auflösung hoch genug, um die Bewegung in einem für die Fragestellung genügend hohen Detailgrad abzubilden? Ein Teil

des Analysekontexts ist kognitiver Natur. Wie verstehen wir Bewegung? Wie werden Bewegungsmuster konzeptualisiert? Wie kann Bewegung kognitiv adäquat visualisiert werden?

Wie diese kurzen Ausführungen zeigen mögen besteht auch zu Fragen über den Analysekontext ein Forschungsbedarf. So interessant solche Fragen aber auch sein mögen, auf den Analysekontext wird im Rahmen dieser Arbeit nicht vertieft eingegangen.

#### 3.1.2 Bewegungskontext

Ein MO bewegt sich in der realen Welt und interagiert mit dieser. Es muss die Perspektive des MO beachtet werden. Die Umwelt wird vom Individuum wahrgenommen. Das MO nutzt vorhandene Informationen über seine Umgebung, um darin entsprechend seiner Motivation und Zielen zu navigieren. Bewegung kann nur stattfinden, wenn das MO die physikalischen Begebenheiten ausnützen kann. Es muss Kräfte auf seine Umgebung ausüben oder Kräfte der Umgebung auf sich einwirken lassen. Der Bewegungskontext ist die Menge solcher Einflüsse auf die Bewegung.

Der Analysekontext bestimmt inwiefern der Bewegungskontext bei der Analyse von Trajektorien berücksichtigt wird. Der Analysekontext hat aber keinen Einfluss auf den Bewegungskontext. Bewegung findet statt, auch wenn sie nicht beobachtet oder gemessen wird. Genauso gibt es den Bewegungskontext, auch wenn dieser nicht beachtet wird.

Vom bewegungsökologischen Modell von Nathan et al. (2008) ausgehend, wird in dieser Arbeit der Bewegungskontext in drei Teile gegliedert.

Der **Motivationskontext** hat mit dem inneren Zustand des Objekts zu tun. Das Objekt verfolgt mit seiner Bewegung bestimmte Ziele. Diese können über unterschiedlichste Skalen formuliert werden. Die Aufrechterhaltung der funktionalen Balance, die Weitergabe des Erbguts und die Erhaltung der Art sind Ziele, die über die Lebensspanne eines Individuums betrachtet werden müssen, oder sogar auf evolutionärer Skala. Ein Mensch, der einen Brief zur Post bringt oder ein Vogel, der zu seinem Nest zurückkehrt, ist eher kurzfristig motiviert. Externe Faktoren können einen Einfluss auf diese Ziele haben. Wenn plötzlich eine Löwin aus dem hohen Gras der Savanne auftaucht, wird sich das Verhalten einer Gazelle von Nahrungsbeschaffung zu Flucht ändern. Es ist zu beachten, dass Ziele und Motivation meist vom Beobachter formuliert werden. Nur der Mensch kann über seine Ziele befragt werden. Wenn man sich mit dem Motivationskontext auseinandersetzt, müssen *frame-of-reference* Probleme beachtet werden.

Der **Navigationskontext** hat mit der Wegfindung zu tun. MO verwenden Information, die sie mit ihren Sinnen wahrnehmen, um ihre Bewegung zu steuern. Ein gutes Beispiel ist die schon in Abschnitt 2.3 erwähnte Ameise *Cataglyphis*. Sie navigiert unter anderem anhand der Polarisation des blauen Himmels. Tauben navigieren mit Hilfe von linearen Land-

schaftselementen, wenn sie zu ihrem Schlag zurückkehren (Laube et al. 2007: 498). Turchin (1998: 37) hält fest, dass wenige Organismen in der Lage sind, sich über weite Distanzen in einer geraden Linie fort zu bewegen, ohne externe Hinweise zu nutzen.

Der **Fortbewegungskontext** beschäftigt sich mit der Mechanik der Bewegung. Jede Fortbewegungsart macht sich physische Eigenschaften der Umgebung zunutze. Gehen funktioniert nur mit Gravitation und genügend grosser Reibung. Schwimmen nutzt Auftrieb, Reibung und Trägheit des Wassers. Samen eines Baumes lassen sich vom Wind tragen. Es müssen also gewisse Voraussetzungen erfüllt sein, damit eine Fortbewegungsweise genutzt werden kann. Wenn die Fortbewegungsstrategie eines MO bekannt ist, kann also darauf geschlossen werden, welcher Kontext vorhanden sein muss, damit es sich dieser Strategie bedienen kann.

Der Bewegungskontext darf nicht als die Summe von Motivations-, Navigations- und Fortbewegungskontext betrachtet werden. Diese Einzelteile interagieren miteinander. Die Motivation eines Subjekts kann die Navigationskapazität beeinflussen. Die gewählte Bewegungsart kann den inneren Zustand verändern. Der Bewegungskontext ist das Resultat eines solchen Zusammenspiels seiner Komponenten.

## 3.2 Zwei Herangehensweisen

Obwohl erst wenige Methoden bestehen, die Bewegungskontext in die MO-Analyse integrieren, lassen sich zwei grundsätzlich unterschiedliche Ansätze erkennen. Diese Ansätze unterscheiden sich in ihrem Arbeitsablauf und in den Fragen, die mit ihnen beantwortet werden können.

### 3.2.1 Erkundender Ansatz

Der erkundende Ansatz (Abbildung 3.1) beschäftigt sich mit der Frage, welche Objekte und Eigenschaften der Umgebung einen Einfluss auf die Bewegung haben. Trajektorien werden zusammen mit Daten über den Kontext analysiert und Zusammenhänge werden gesucht. Dabei können explorative Techniken wie z.B. *data mining* oder Visualisierungen verwendet werden. Die *radial distance function* von Imfeld (2000) ist ein Beispiel einer Methode, die dem erkundenden Ansatz entspricht.

Resultate von erkundenden Methoden liefern Aufschluss darüber, welche Aspekte der Umgebung oder des MO relevant für die Bewegung sind. So gewonnenes Wissen kann verwendet werden, um das MO in seinem Bewegungskontext zu modellieren, oder um bereits bestehende Modelle zu verbessern. Solche Modelle können verwendet werden, um die Bewegung eines MOs zu erklären.

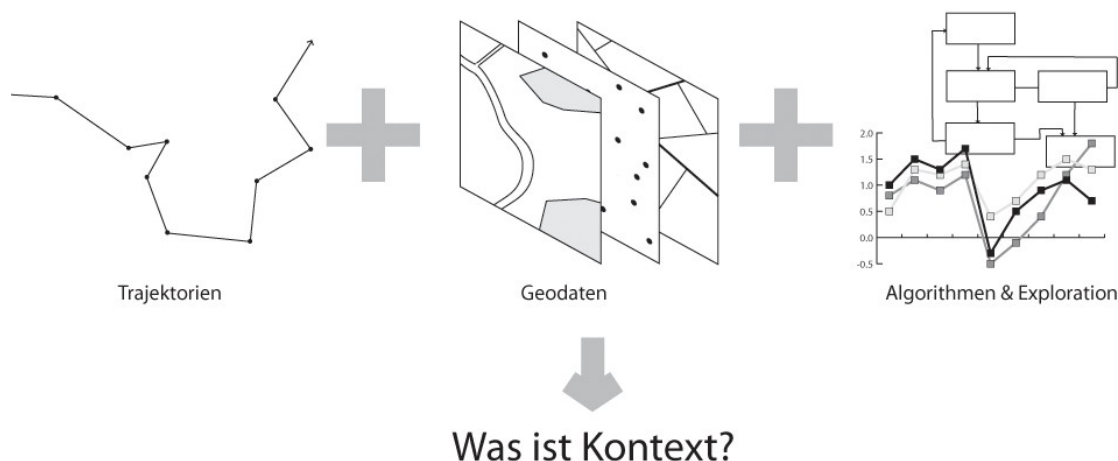


Abbildung 3.1: Schematischer Ablauf des erkundenden Ansatzes  
Trajektorien und Geodaten über den Kontext werden mit speziell dafür entwickelten Methoden analysiert. Dabei werden Beziehungen zwischen der Bewegung und dem Umfeld, in dem sie stattfindet, gesucht.

#### 3.2.2 Erklärender Ansatz

Um Fragen nach dem Wie?, Wann?, Wo? und Warum? der Bewegung eines MO zu beantworten, muss die Bewegung in Relation zu Verhalten und Aktivitäten gesetzt werden. Verhalten ist immer eine Interaktion zwischen einem System und seiner Umgebung (Abschnitt 2.2.5). Also kann der Bewegungskontext verwendet werden, um Bewegung als Resultat von Verhalten zu erklären und interpretieren. Das Ziel dieser Herangehensweise an die Analyse von Bewegungsdaten ist, die Bewegung zu erklären.

Um Bewegungsdaten zu erklären braucht der Analytiker ein Modell, das die Zusammenhänge zwischen Bewegung und Kontext aufzeigt. Die Relationen zwischen MO, relevanten Aspekten der Umgebung, Eigenschaften des MOs und dem realisierten Pfad sind also zumindest teilweise bekannt. Mit dem Modell können Daten über den Kontext in eine Form gebracht werden, in der sie mit den Trajektorien kombiniert werden können. Die mit Kontextinformation angereicherten Trajektorien können dann analysiert, erklärt und interpretiert werden. Dieser Prozess ist in Abbildung 3.2 visualisiert.

Ein Beispiel für ein solches Vorgehen ist der *places of interest* Ansatz. Der Analytiker weiss, welches im Rahmen seiner Analyse die wichtigen Orte (*places*) sind. Er hat also ein Modell, das einen Teil des räumlichen Bewegungskontexts erklärt. Er verwendet diese Kontextinformation, um die Trajektorien in eine neue Form zu bringen. Wo immer die Trajektorien in der Nähe eines wichtigen Ortes liegen, wird aus der Trajektorie ein *stop*. Dazwischen bewegt sich das MO in *moves*. Nun kann der Analytiker die *stops* auswerten. Es können Aussagen gemacht werden wie: dieser Tourist ging am Morgen von seinem Hotel in ein Museum, blieb dort mehrere Stunden, ging dann zurück ins Hotel. Die Bewegung zwischen Hotel und Museum wird durch die *stops*, die Ausgangspunkte und Ziele der Bewegung sind, erklärt.

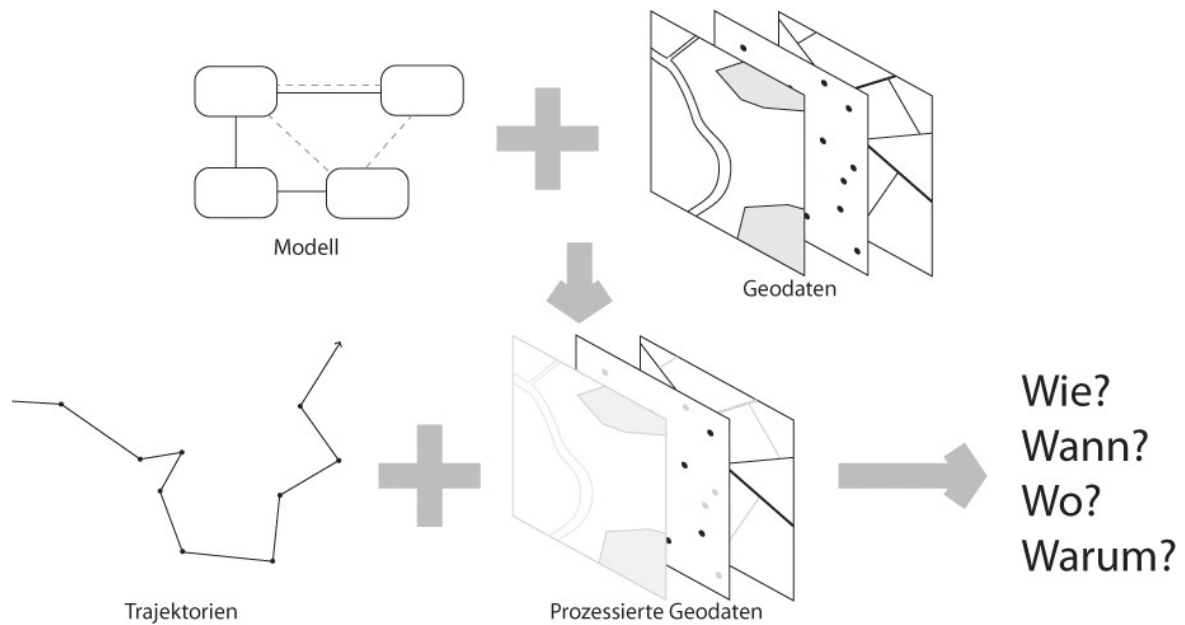


Abbildung 3.2: Schematischer Ablauf des erklärenden Ansatzes

Ein Modell von MO und Kontext wird benutzt um Geodaten vorzubereiten. Trajektorien und prozessierte Geodaten geben Aufschluss über den Bewegungskontext und erklären die Bewegung.

### 3.3 Verknüpfung bestehender Klassifikationen

Ziel (2) dieser Arbeit ist es Konzepte für den Kontext der Bewegung zu entwickeln. Ziel (2) verlangt aber auch ausdrücklich, dass uns diese Ziele helfen sollen Kontext besser zu verstehen. In diesem Abschnitt werden die vorgeschlagenen Kontextklassen mit bereits bestehenden Klassifikationen verknüpft. Wenn dies zu einem besseren Verständnis führt, sind die Kontextklassen nützlich.

Dodge et al. (2008) und Andrienko und Andrienko (2007) haben Kontextfaktoren eingeteilt. Die Klassen dieser Einteilungen oder zumindest die genannten Beispiele sollten sich auf das Modell von Nathan et al. (2008) abbilden lassen. Die einzelnen Faktoren lassen sich Motivationskontext, Navigationskontext und Fortbewegungskontext zuteilen, auch wenn diese Zuteilung nicht immer offensichtlich oder eindeutig ist. Tabelle 3.1 zeigt das Resultat für die Klassifikation von Dodge et al. (2008). Diese Zuteilung wird im Detail diskutiert. Für die umfangreichere Einteilung von Andrienko und Andrienko (2007) in Tabelle 3.2 wird auf eine ausführliche Diskussion verzichtet. Dafür wird auf problematische Stellen und schwierige Zuweisungen eingegangen.

### 3 Was ist Kontext?

Tabelle 3.1: Einteilung der Faktoren der Klassifikation von Dodge et al. (2008) nach Motivations-, Navigations- und Fortbewegungskontext.

	Motivations-Kontext „Why move?“	Navigations-kontext „Where to move?“	Fortbewegungs-kontext „How to move?“
Intrinsische Eigenschaften des MO			
Räumliche Einschränkungen			
Umgebung			
Einfluss anderer MO			

#### **Intrinsische Eigenschaften des MO**

Dodge et al. (2008) nennen als Beispiele die Maximalgeschwindigkeit und das Beschleunigungsverhalten des Objekts. Dies sind streng genommen keine Eigenschaften des MO per se, sondern resultieren aus einer Interaktion mit der Umgebung. Ein Automobil auf einer trockenen Strasse hat ein anderes Beschleunigungsverhalten als dasselbe Fahrzeug auf einer schneebedeckten Strasse oder in einem frisch gepflügten Feld. Ein Sprinter über 100 Meter hat bessere Chancen, eine persönliche Bestzeit zu laufen, wenn Rückenwind herrscht, als wenn er gegen den Wind ankämpfen muss. Die hier erwähnten Eigenschaften lassen sich im Modell von Nathan et al. (2008) an der Schnittstelle zwischen externen Faktoren und Bewegungskapazität einordnen. Sie gehören daher zum Fortbewegungskontext.

#### **Räumliche Einschränkungen**

Was eine Barriere ist, hängt von der Art der Fortbewegung ab. Ebenso sind gewisse Fortbewegungsarten an Netzwerke gebunden. So ist z.B. eine Strasse wichtig, wenn auch nicht zwingend notwendig, für Fortbewegungsarten, die sich Räder zunutze machen. Schiffe brauchen für ihre Fortbewegung nicht nur Wasser, sie brauchen eine gewisse Wassertiefe, um nicht auf Grund zu laufen. So ist eine Sandbank, einige Dezimeter unter der Wasseroberfläche, für eine Ente kein Hindernis, kann aber für ein grosses Frachtschiff eine unüberwindbare Barriere sein. Räumliche Einschränkungen physischer Form sind wiederum dem Fortbewegungskontext zuzuordnen.

#### **Umgebung**

Ob Objekte den Pfad eines MO durch anziehende oder abstossende Wirkung beeinflussen können, hängt im *movement ecology framework* von Nathan et al (2008) von zwei Faktoren ab. Erstens muss der innere Zustand des Objekts derart sein, dass es sich zu einem bestimmten Objekt hin bewegt, oder von einem Objekt weg bewegt. Zweitens muss das MO in der Lage sein das Objekt zu erkennen und seine Bewegung, als Folge des Erkennens, in

die gewünschte Richtung zu lenken. Die anziehende und abstossende Wirkung der Umgebung ist also Teil des Navigations- und Motivationskontexts.

#### **Einfluss anderer MO**

Andere MO sind Teil der Umgebung. Daher gilt auch hier, dass sie anziehend oder abstossend auf ein MO wirken können. Die Trennung zwischen anderen MO und dem Rest der Umgebung ist nützlich, da es zwischen MO komplizierte Bewegungsmuster gibt. Die Muster von Flucht und Verfolgung, die zwischen Räuber und Beute entstehen können, sind wesentlich komplizierter als Bewegungsmuster, die es zwischen einem statischen und einem mobilen Objekt gibt. Andere MO können allen drei Arten des Bewegungskontexts zugeordnet werden. So kann sich z.B. ein kleineres MO die Bewegung eines grösseren MOs zunutze machen und sich von diesem transportieren lassen.

Die Einteilung von Andrienko und Andrienko (2007) lässt sich, da eher Aufzählung als Klassifikation, nur schwierig auf das bewegungsökologische Modell und unsere Klassen für den Bewegungskontext abbilden. Ein Versuch einer Abbildung ist in Tabelle 3.2 zu sehen. Anstatt die ganze Tabelle hier zu diskutieren, werden die dabei auftretenden Probleme anhand einiger Einträge illustriert.

Alltägliche Begriffe wie das Wetter lassen sich nur schwer in das Modell von Nathan et al. (2008) einordnen. Mit dem Wetter verbundene Phänomene könnten auf den inneren Zustand, die Navigationskapazität und die Bewegungskapazität einen Einfluss haben. Nebel erschwert die Navigation, ein Autofahrer verpasst eine Abzweigung und muss wenden (Navigationskontext). Auf nassen Strassen sollte nicht all zu schnell gefahren werden, eine vom Regen durchnässte Böschung ist schwierig zu erklettern (Fortbewegungskontext). Jemand verzichtet auf seinen Spaziergang, weil es windig ist und Regen droht (Motivationskontext). Während der Begriff des Wetters schwierig einzuteilen ist, können die eben gemachten Präzisierungen im bewegungsökologischen Modell problemlos zugeordnet werden.

Ein anderes Beispiel ist Müdigkeit. Bei Andrienko und Andrienko (2007) würde Müdigkeit zu den individuellen Eigenschaften eines MOs gezählt. Auch hier könnten alle drei Klassen zum Zug kommen. Müdigkeit kann die Fähigkeit zur Navigation einschränken. Ein übermüdeten Körper ist weniger leistungsfähig, die Bewegungskapazität ist eingeschränkt. Auch die Motivation kann von der Müdigkeit beeinflusst werden, das Bedürfnis nach Erholung verdrängt andere Ziele.

### 3 Was ist Kontext?

Tabelle 3.2: Einteilung der Faktoren der Klassifikation von Andrienko und Andrienko (2007) nach Motivations-, Navigations- und Fortbewegungskontext.

	Motivations-Kontext „Why move?“	Navigations-kontext „Where to move?“	Fortbewegungs-kontext „How to move?“
<b>Properties of space</b>			
Characteristics of the terrain			
Accessibility / constraints			
Properties of surface			
Objects present in location			
Function / mode of use			
Meaning of place for moving entity			
<b>Properties of time</b>			
Temporal cycles			
Physical characteristics (e.g. light)			
Meaning in terms of typical activities			
<b>Properties and activities of moving entities</b>			
Individual properties (e.g. age, gender)			
Way of movement (e.g. by roads)			
Means of movement (e.g. vehicles)			
Purpose or cause of movement			
Activities performed during movement			
<b>Various spatial, temporal and spatiotemporal phenomena</b>			
Climate, weather			
Cultural events			
Legal regulations, customs			
Prices			

Die Einteilung des Bewegungskontexts in Motivations-, Navigations- und Fortbewegungskontext ist nützlich. Beide Klassifikationen können abgebildet werden. Problematische Begriffe wurden identifiziert. Allgemeine Begriffe, die eine ganze Reihe von Phänomenen bezeichnen, die in vielerlei Art und Weise mit der Bewegung interagieren können, müssen präzisiert werden. Fragestellungen können mithilfe der drei Arten des Bewegungskontexts genauer formuliert werden. Die Klassen für den Bewegungskontext helfen, Fragen über Zusammenhänge zwischen Kontext und Bewegung präzise zu stellen. Sie tragen damit zu einem besseren Verständnis des Kontext-Begriffs bei.

# 4 Modellierung

Dieses Kapitel präsentiert erste Schritte in Richtung eines eigenen Modells für Bewegungskontext. Zielsetzung (2) dieser Arbeit ist es, Grundlagen zu schaffen, die dann in einem weiteren Schritt in einer Anwendung implementiert werden können. Eine Anwendung, in die alle bisher diskutierten Aspekte des Bewegungskontext einfließen, würde zu komplex werden. Deshalb sind die Modelle in diesem Kapitel beschränkt auf eine der drei Klassen von Bewegungskontext.

Von den drei Kontextklassen ist der Fortbewegungskontext am einfachsten modellierbar. Wenn bekannt ist, welche Strategien das Objekt zur Fortbewegung anwendet, können die Eigenschaften der Trajektorien mit diesen Strategien verglichen werden. Wenn sich z.B. eine Ente ganz besonders schnell und in gerader Linie fortbewegt hat, liegt der Schluss nahe, dass die Ente geflogen ist. Wenn sie hingegen langsam unterwegs ist und ihr Pfad stark gewunden ist, kann sie unmöglich geflogen sein. Es kann also unter Umständen alleine aus den Trajektorien auf die verwendete Fortbewegungsart geschlossen werden.

## 4.1 Anforderungen an das Modell

Es gibt zwei Wege, um von einem Problemgegenstand ausgehend ein konzeptuelles Modell zu erstellen. Es wird entweder vom Modellgegenstand her abstrahiert, oder es werden Daten über das Modell gesammelt und diese werden exploriert. In dieser Arbeit wird bei der Modellierung abstrahierend vorgegangen. Dykes und Mountain (2003) benutzten einen explorativen Ansatz, um Aktivitätsklassen für Trajektorien der Bewegung von Menschen zu finden. Sie konnten in ihren Daten vier *activity classes* unterscheiden (Tabelle 4.1). Der von ihnen verwendete Begriff Aktivitätsklasse kann auch durch Fortbewegungsart ersetzt werden. Dykes und Mountain (2003) nennen zu jeder Fortbewegungsart was bei Geschwindigkeit, Sinuosität, Distanzen und Bewegungsfreiheit im Raum relativ zu den anderen Arten für Werte erwartet werden können.

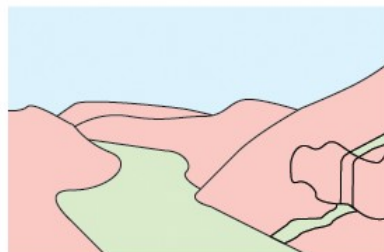
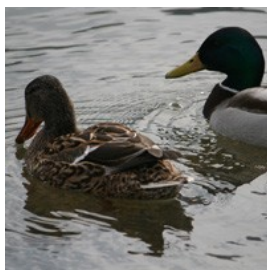
Tabelle 4.1: Aus Bewegungsdaten abgeleitete Aktivitätsklassen (nach Dykes und Mountain 2003: 588)

Activity Class	Speed	Sinuosity	Spatial Range	Freedom of movement
Foot	Low	High	Low	High (self-powered, unpredictable)
Low-speed motor	Higher	Less	Higher	High
High-speed motor	Fast	Low	High	Lower (constrained by network)
Flight	Very high	Lower	Very high	Entirely constrained (endpoint can not be changed when flying with an airline)

Beschreibende Variablen der Bewegung sind im Modell von Dykes und Mountain (2003) zentral für die Bestimmung der Bewegungsart. Das hier vorgestellte Modell sollte deshalb auch dazu verwendet werden können, Aussagen über erwartete Geschwindigkeiten und anderen Variablen aufgrund der Bewegungsart zu machen.

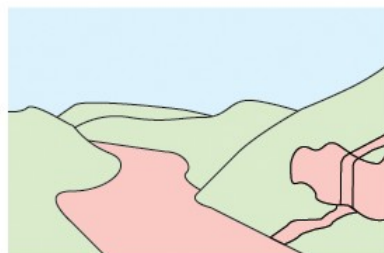
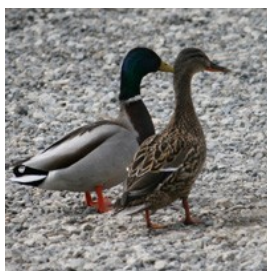
Fortbewegungsarten stellen bestimmte Ansprüche an die Umgebung, in der die Bewegung stattfindet (vgl. Abbildung 4.1). Geodaten über diese Umgebung können eingesetzt werden, um den Fortbewegungskontext zu untersuchen. Aus diesen ersten Betrachtungen lassen sich Anforderungen an ein Modell für den Fortbewegungskontext ableiten:

- (1) Das Modell muss verschiedene Bewegungsarten und ihre Relation zu Oberflächenarten abbilden können (Abbildungsmerkmal).
- (2) Es sollte eine Reduktion auf die wesentlichen Aspekte stattfinden (Verkürzungsmerkmal).
- (3) Das Modell wird unter anderem dazu entwickelt, um Geodaten über den Kontext der Bewegung in deren Analyse integrieren zu können. Daher sollte auf die Eigenschaften von Geodaten Rücksicht genommen werden (pragmatisches Merkmal).



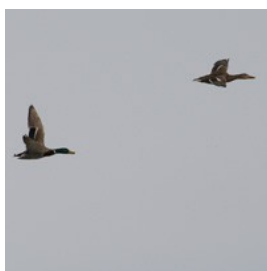
Fortbewegungsart: **Schwimmen**  
Terrain: Bach, Fluss, See  
Geschwindigkeit: langsam

Schwimmend bewegt sich die Ente in Gewässern fort. In einem Wasserfall wird sie aber nicht schwimmend anzutreffen sein.



Fortbewegungsart: **Gehen**  
Terrain: flache Landoberflächen  
Geschwindigkeit: langsam

Gehend kann die Ente sich an Land fortbewegen. Steiles Gelände ist für sie ein Hindernis. Auf dem Wasser kann sie nicht gehen.



Fortbewegungsart: **Fliegen**  
Terrain: beliebig  
Geschwindigkeit: schnell

Fliegend gibt es für die Ente keine Hindernisse. Durch Steigerung der Flughöhe kann sie selbst hohe Berge überfliegen.

Abbildung 4.1: Fortbewegungsart-Oberfläche Relationen am Beispiel der Ente

## 4.2 Ein einfaches Modell für Fortbewegungskontext

Ein MO hat zumindest eine Fortbewegungsart, sonst wäre es ein gewöhnliches *object* und kein *moving object*. Als physischer Prozess ist jede Fortbewegungsart auf bestimmte Eigenschaften der Umgebung angewiesen. Die Bewegungsart funktioniert also nur auf bestimmten Oberflächen oder in bestimmten Medien. Gehen ist für viele MO auf grossen Teilen des Festlandes der Erde möglich. In zu steilem Gelände ist diese Fortbewegungsart aber nicht möglich. Geschwommen werden kann in Wasser. Je spezialisierter die Fortbewegungsart, desto kleiner der Anteil der Erdoberfläche, auf dem sie genutzt werden kann. Ein Automobilist mit seinem Fahrzeug kann sich zwar nicht nur auf Strassen bewegen, er ist aber auf flachen, harten Boden angewiesen. Ein Lokomotivführer kann sich mit seinem Zug nur auf einem Schienennetzwerk bewegen. Fliegende Organismen hingegen sind in ihrer Bewegungsart nicht auf bestimmte Landoberflächen eingeschränkt. Wenn das fliegende Objekt hoch genug fliegt, kommt es über jegliche Hindernisse hinweg. In einem einfachen Modell für Fortbewegungskontext braucht es also das MO, seine Fortbewegungsart und Geländeoberflächen.

Ein MO hat mechanische Bewegungsstrategien, die auf bestimmten Oberflächen funktionieren. Gleichzeitig gibt es andere Oberflächenarten, auf denen diese Bewegung nicht funktioniert. Wenn wir eine Ente zuerst am Ufer eines Sees antreffen, und später auf einer Insel in diesem See, wissen wir, dass sie nicht gehend dorthin gelangt ist. Der See ist eine Oberfläche, auf der die Ente nicht gehen kann, sie muss eine andere Bewegungsart verwendet haben (vgl. Abbildung 4.1).

Abbildung 4.2 ist ein *entity-relationship* Modell (Chen 1976), das den Zusammenhang zwischen MO, seinen Bewegungsarten und der Oberfläche beschreibt. Oberfläche ist hier ein allgemeiner, wenig präziser Begriff. Darin fliessen relevante Eigenschaften, wie Oberflächenbedeckung, Steigung, Festigkeit und Reibungswiderstand mit ein. Die Effizienz einer Bewegungsart hängt davon ab, wie geeignet die Oberfläche für diese Strategie ist. Daher sind die Maximalgeschwindigkeit und -beschleunigung der Fortbewegung als Attribute der Beziehung zwischen Fortbewegungsart und Oberfläche modelliert. Schon im Zu-

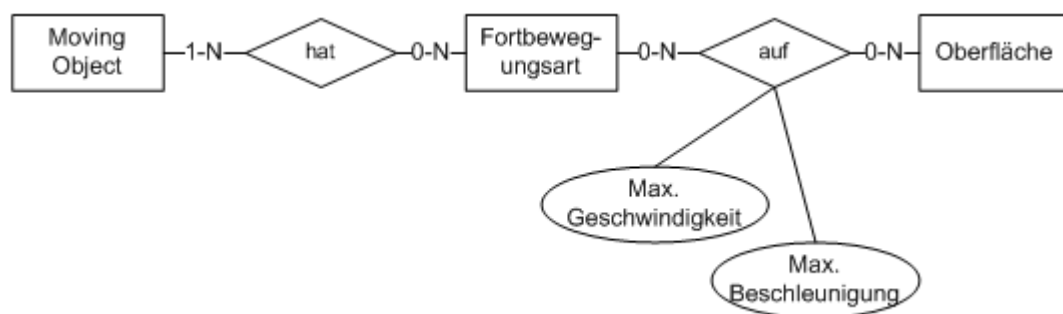


Abbildung 4.2: Einfaches ER-Modell für Fortbewegungskontext

sammenhang mit den Klassifikationen wurde als Beispiel dafür ein Auto erwähnt, das auf einer Strasse und in einem nassen Feld ein unterschiedliches Beschleunigungsverhalten hat.

### 4.3 Einbezug der Bewegung

Wie kann nun die Information über die Trajektorie des MO in das Modell für Fortbewegungskontext integriert werden? Wie im ER-Modell in Abbildung 4.3 zu sehen ist hat das *moving object* eine Trajektorie, die den Pfad seiner Bewegung repräsentiert. Die Trajektorie besteht aus *fixes*, die Trajektorie besteht im Normalfall aus vielen *fixes*. Jeder *fix* gehört zu genau einer Trajektorie.

Dieses Modell verknüpft das Datenmodell für Trajektorien mit dem Fortbewegungskontext. Während der ganzen Dauer der Bewegung benutzt das MO eine Fortbewegungsart. Also kann jedem *fix* die Fortbewegungsart zum Zeitpunkt der Positionsmessung zugewiesen werden. Da der *fix* eine Messung der Position des Objekts im Raum ist, kann jedem *fix* die Oberfläche an seiner Position zugeordnet werden. Für jeden *fix* kann eine aktuelle Geschwindigkeit und eine aktuelle Beschleunigung bestimmt werden. Diese können mit den für die Bewegungsart definierten Maxima verglichen werden.

Wenn die Bewegungsart für den *fix* nicht bekannt ist, können die aktuelle Geschwindigkeit, Beschleunigung und die Oberfläche an der Position des *fixes* helfen diese herauszufinden.

Dykes und Mountain haben zur Beschreibung einer Bewegungsart den Begriff *freedom of movement* verwendet. Das Fortbewegungskontext-Modell versucht genau diese Bewegungsfreiheit zu modellieren. Es ist die Tatsache, dass bestimmte Fortbewegungsarten nur auf gewissen Oberflächen möglich sind, die die Bewegung räumlich einschränkt.

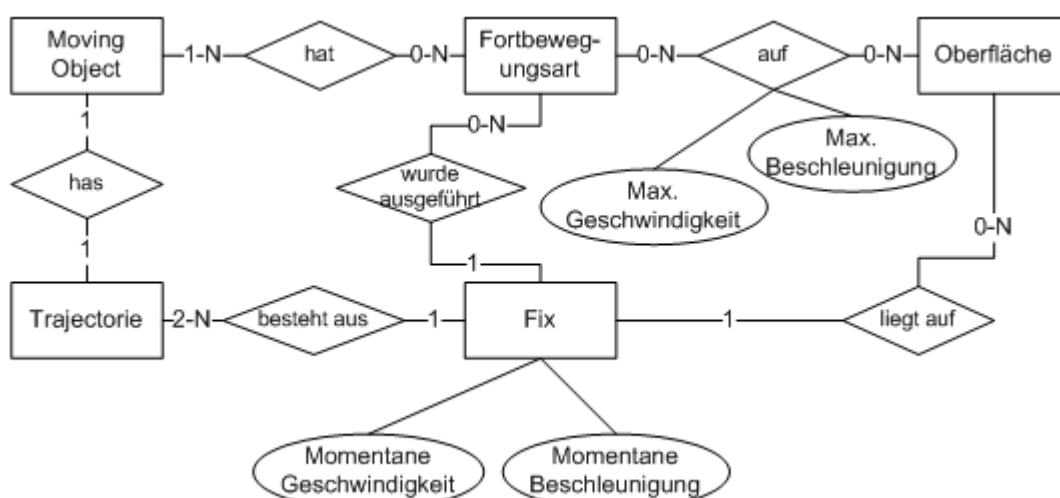


Abbildung 4.3: Um die Trajektorie erweitertes ER-Modell für Fortbewegungskontext

## 4.4 Probleme

Diese Modelle können eine **dynamische Welt** nur schlecht wiedergeben. Was ist, wenn ein Fluss im Winter zufriert? Er ist jetzt kein Hindernis mehr für MO, die nicht schwimmen können. Was sonst eine Barriere ist, kann nun problemlos überquert werden. Der Fluss wird aber kaum jedes Jahr zur gleichen Zeit zufrieren, und er wird auch nicht immer gleich lang eisbedeckt bleiben. Vielleicht führt er im Sommer manchmal so wenig Wasser, dass er von grossen MO durchwatet werden kann. Bei starkem Niederschlag führt der Fluss Hochwasser und schafft neue Barrieren. Nicht nur die MO, auch die Umwelt in der sie sich bewegen ist dynamisch. Die Modellierung einer dynamischen Umwelt stellt uns vor ähnlich grosse Herausforderungen wie die Modellierung der MO.

Eine zweite grosse Herausforderung ist die **Modellierung der Oberfläche**. Wie schon erwähnt müssen hier, je nach Bewegungsart, neben Oberflächenbedeckung auch andere Faktoren, wie Reibungswiderstand oder Steigung einfließen. Während es im einen Fall genügen kann, zwischen Land und Wasser zu unterscheiden, kann es in einem anderen Fall nötig sein viele Faktoren zu verknüpfen.

Eine dritte Herausforderung sind die räumliche und zeitliche **Auflösung der Daten**. Wie räumlich genau und zeitlich hoch aufgelöst sind die Trajektorien? Sind die Daten über das Gelände genau detailliert erfasst, um den Kontext der Bewegung auf dieser Skala abzubilden? Ein digitales Rasterhöhenmodell mit 100 Meter Auflösung ist nicht feinmaschig genug, um steile Felswände geringer Höhe abzubilden. Es eignet sich folglich nicht, um solche Barrieren zu finden. Wenn die Genauigkeit der Daten über den Fortbewegungskontext nicht ähnlich derjenigen der Bewegungsdaten ist, können Detailfragen über die Bewegung nicht beantwortet werden. Wenn der Kontext dynamisch ist, wir aber nur Daten über langjährige Durchschnittswerte haben, können wir keine Aussagen über den kurzfristigen Bewegungskontext machen.

Ein weiteres Problem ist die **Unsicherheit**, die mit der Zuweisung von Bewegungsart und Oberflächen verbunden ist. Stellen wir uns vor, Hirsche seien Untersuchungsobjekte, und wir haben für sie die Bewegungsarten 'Springen' und 'Gehen' modelliert. Zusätzlich haben wir eine Klasse der Oberflächenbedeckung, die wir Hecke nennen. Nun ist die Frage, ob Hecken ein Hindernis für Hirsche sind. Ist die Hecke dicht und hat der Hirsch ein grosses Geweih, wird er nicht einfach hindurchgehen können. Ist die Hecke nicht sehr hoch, kann er aber darüber hinweg springen. Ob die Hecke ein Hindernis ist, hängt also von deren Dichte, Breite und Höhe ab, aber auch von der Grösse des Hirsches und von seiner Sprungkraft. Wir können also ohne genaue Informationen nicht sicher sein, ob ein Objekt ein Hindernis ist oder nicht. Vielleicht können wir Aussagen über die Wahrscheinlichkeit machen, ob ein Objekt eine Barriere ist. Weiter erschwerend hinzu kommt, dass räumliche

Objekte, wie die erwähnte Hecke oder ein Fluss nicht homogen sind. Die Breite und Tiefe eines Flusses variieren, die Dichte, Höhe und Breite einer Hecke ebenfalls.

Auch im Zusammenhang mit Bewegungsarten treten Problem mit der **Perspektive** auf. Was ist eine Fortbewegungsart? Wo werden die Grenzen gesetzt? Soll für ein Pferd eine Bewegungsart 'Gehen' definiert werden, oder soll zwischen den bewegungsmechanisch unterschiedlichen 'Schritt', 'Trab' und 'Galopp' differenziert werden? Wo sollen beim Menschen die Grenzen gezogen werden? Reicht eine Klasse 'Motorisierter Verkehr' oder soll zwischen 'Privatauto', 'Motorrad', 'Bus', 'Taxi', usw. unterschieden werden? Der Analysekontext entscheidet über diese Fragen. Der Forscher sollte sich aber auf jeden Fall bewusst sein, dass eine Einteilung der Bewegungsarten eines MO vom Beobachter gemacht wird, und nicht eine gegebene Eigenschaft des MO ist.

### 4.5 Anwendungsmöglichkeiten

Was sind die Anwendungsmöglichkeiten für das soeben präsentierte Modell des Fortbewegungskontexts? Fünf potenzielle Anwendungsgebiete wurden identifiziert und werden hier beschrieben.

- (1) Zuweisung einer Bewegungsart zu *fixes* oder Episoden einer Bewegung. Mit Hilfe von Informationen über die Oberfläche, auf der die Bewegung stattfindet, kann unter Umständen auf die Bewegungsart geschlossen werden. Wenn eine direkte Schlussfolgerung nicht möglich ist, können beschreibende Variablen wie Geschwindigkeit, Beschleunigung oder Sinuosität herbeigezogen werden.
- (2) Identifikation von Hindernissen im Raum. Wenn bekannt ist, auf welchem Terrain eine Bewegungsart funktioniert, ist auch bekannt, auf welcher Oberfläche diese Bewegung nicht stattfinden kann. So können Hindernisse für die Bewegung identifiziert werden. Diese können dann in die Analyse miteinbezogen werden.
- (3) Ersetzen der euklidischen Distanz durch den kürzesten Weg um Hindernisse herum. Die Distanz zwischen zwei Punkten im Raum wird normalerweise euklidisch bestimmt. Aus der Perspektive des MO ist ein Objekt so weit entfernt, wie der kürzeste Weg zu diesem Objekt ist. Unter Einbezug der Hindernisse kann dieser kürzeste Weg berechnet werden.
- (4) Rückschlüsse auf den Weg zwischen zwei *fixes*. Wenn die Oberfläche zwischen zwei Messpunkten der Trajektorie und die Fortbewegungsart bekannt sind, können Aussagen darüber gemacht werden, wie sich das Objekt zwischen den zwei *fixes* bewegt haben könnte.

- (5) Simulation von MO. Das Modell des Fortbewegungskontexts kann bei der Simulation von sich bewegenden Objekten eingesetzt werden. Daten über Hindernisse können z.B. in Kombination mit einem *random walk* Modell verwendet werden. Die resultierenden Trajektorien haben die Eigenschaft, dass die simulierten MO sich nicht auf Oberflächen bewegen, auf denen echte MO sich gar nicht fortbewegen können.



# 5 Algorithmus

In diesem Kapitel wird ein Algorithmus präsentiert, der das Fortbewegungskontext-Modell aus Kapitel 4.3 verwendet. Dieser Algorithmus bildet die Grundlage für Ziel (3) dieser Arbeit, die Implementierung einer Anwendung, die Informationen über den Kontext der Bewegung verwendet. Hindernisse werden identifiziert, um dann bei der Berechnung der Distanz verwendet zu werden. Der Begriff der **Umgehungsdistanz** wird in dieser Arbeit verwendet, um die Länge des kürzesten Pfades zwischen zwei Punkten zu bezeichnen, der nicht durch das Innere von Hindernissen führt. Die Umgehungsdistanz kann die euklidische Distanz ersetzen.

Die Aufgabe des hier vorgestellten Algorithmus ist, ein Polygon zu berechnen, das alle Punkte in der Ebene enthält, deren Umgehungsdistanz zu einem Punkt kleiner oder gleich ist wie ein Schwellwert. Ein solches Polygon wird Umgehungsdistanz-Buffer genannt. Diese Polygone können dann in einer erweiterten Version der *radial distance function* von Imfeld (2000) eingesetzt werden. Die RDF ist Vertreter des erkundenden Ansatzes, das verwendete Modell des Fortbewegungskontexts ist erklärend. Es werden also die beiden Ansätze kombiniert.

## 5.1 Bezug zur Literatur

Dieser Abschnitt stellt den Bezug des Algorithmus zur Literatur her. Zuerst wird die schon mehrfach erwähnte *radial distance function* (RDF) von Imfeld (2000) vorgestellt (Abbildung 5.1). Dabei wird auch auf Ähnlichkeiten zwischen RDF und Konzepten der *time geography* eingegangen. Die Umgehungsdistanz ist über den kürzesten Pfad definiert. Die Berechnung von kürzesten Pfaden ist ein Thema, mit dem sich die Computerwissenschaften ausführlich auseinandergesetzt haben. Von den Methoden der *computational geometry* können Grundlagen und Lösungsansätze für den Algorithmus abgeleitet werden.

### 5.1.1 Die *Radial Distance Function*

In seiner Doktorarbeit stellt Imfeld (2000) die *radial distance function* (RDF) vor. Diese erkundende Analysemethode ist eine der wenigen Methoden, die Kontext in die Auswertung von Bewegungsdaten mit einbeziehen. Die RDF verbindet einen Punkt in der Trajektorie eines MO mit Eigenschaften der Umgebung.

Die RDF verwendet konzentrische Kreise um einen *fix* herum. Diese Kreise teilen die unmittelbare Umgebung in Ringe ein. Die Umgebung wird aufgeteilt in Klassen der Distanz zum Ausgangspunkt. Für jeden dieser Ringe wird ein Wert bestimmt. Dieser Wert kann z.B. der Anteil von Wald an der Gesamtfläche des Ringes sein (Abbildung 5.1).

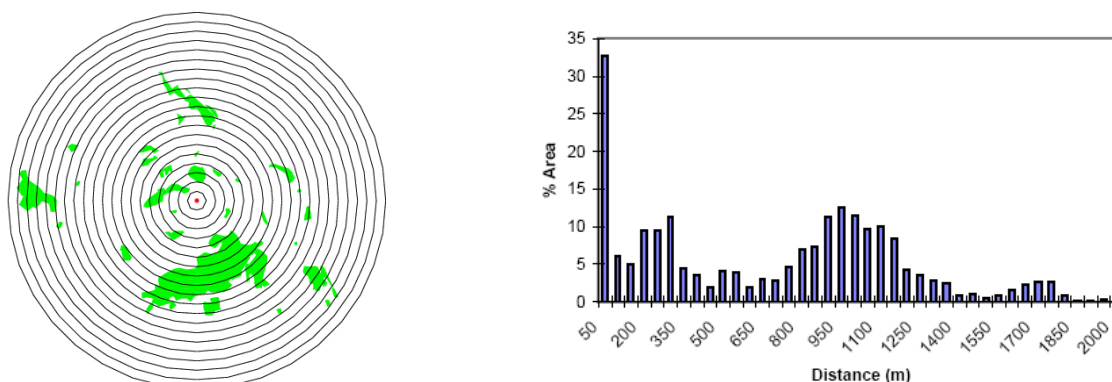


Abbildung 5.1: RDF und Statistik der Ringe (aus Imfeld 2000)  
 Links: Konzentrische Ringe der RDF, eine Bodenbedeckung (Wald) ist hervorgehoben.  
 Rechts: Anteil einer Bodenbedeckungskategorie an der Gesamtfläche der Ringe der RDF.

Um mehrere zeitlich aufeinander folgende RDF Funktionen zu visualisieren verwendet Imfeld (2000) die *temporal RDF-plots* (Abbildung 5.2). In diesen ist ein Anteil an der Fläche, als Funktion der Distanz, auf der y-Achse dargestellt. Die x-Achse repräsentiert die Zeit. Die Anteile sind so über einen Zeitraum visualisiert. Veränderungen in der Zusammensetzung der Umfeldes, als Funktion der Zeit, können in der Graphik erkannt werden. Der *temporal RDF-plot* visualisiert, wie sich ein bestimmter Teil des räumlichen Kontexts während der Bewegung verändert. Imfeld verwendete die RDF unter anderem dazu, den Waldanteil, um *fixes* von Trajektorien von Luchsen herum, zu analysieren. Imfeld konnte zeigen, dass die Nutzung des Raumes von einzelnen Individuen stark variiert. Bei einem Luchs wechselten sich Perioden in dicht bewaldetem Gebiet mit Perioden in offenem Gelände ab. Beim zweiten Luchs trat dieses Phänomen nicht auf.

Imfeld (2000: 103) hält fest, dass seine Anwendungen der RDF immer die Fläche als Mass verwenden und dass die für die Berechnung benutzte Form immer ein Kreis ist. Diese Beschränkung auf Kreis und Fläche hält Imfeld nicht für nötig. Anstelle der Fläche könnten zahlreiche andere Masse verwendet werden. Imfeld nennt Umfang, Länge, Anzahl von Objekten und Anzahl verschiedener Habitatstypen als Beispiele. Er schlägt vor, dass Barrieren in die Berechnung der RDF miteinbezogen werden. Nur Teile der Umgebung, die

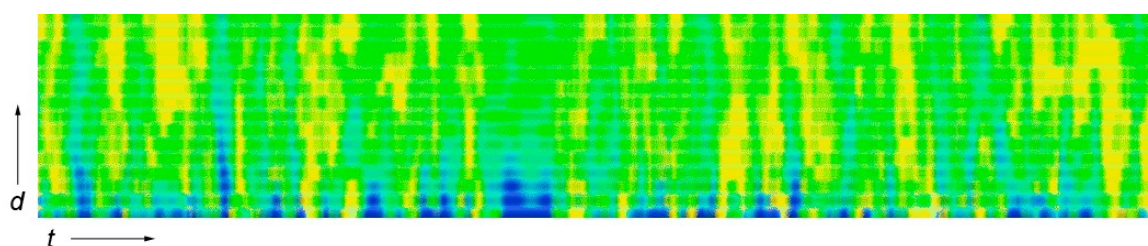


Abbildung 5.2: Temporal RDF-plot (Imfeld 2000)  
 Waldanteil als Funktion der Distanz, während einer Untersuchungsperiode. Hoher Anteil an Wald ist blau dargestellt, niedriger Anteil ist gelb.

vom MO auch erreicht werden können, sollen einen Einfluss auf das Resultat der RDF haben. Anstelle der euklidischen Distanz könnte auch die Distanz verwendet werden, die das MO zurücklegen muss, um an einen Ort zu gelangen. Genau dies wird mit der Umgehungsdistanz-RDF umgesetzt.

Die RDF hat gewisse Ähnlichkeiten mit Konzepten, die in der *time geography* Verwendung finden. Isochronen für die Reisezeit eines Individuums im homogenen Raum resultieren in ebensolchen Kreesen (Abbildung 5.3a). Isochronen im heterogenen Raum gleichen der Umgehungsdistanz-RDF (Abbildung 5.3b).

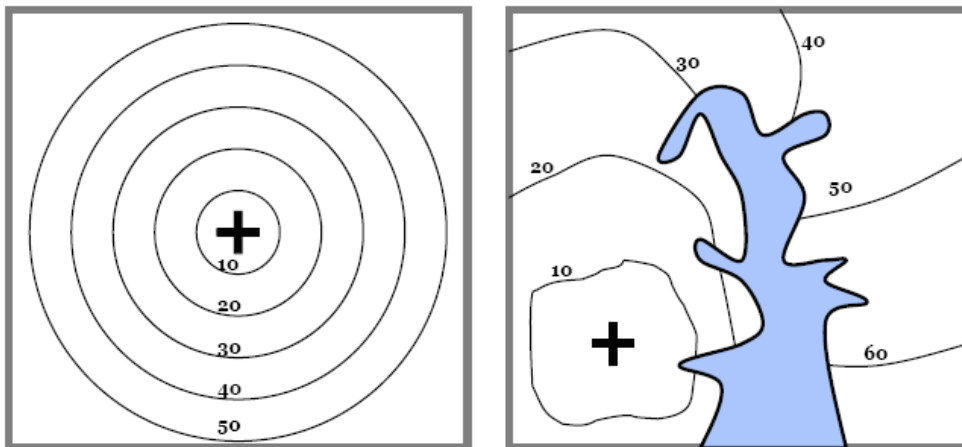


Abbildung 5.3: Isochronen für die für den Weg benötigte Zeit von einem Punkt aus. (Laurini und Thompson 1992, in Mountain 2005)

a) Im homogenen Raum

b) Im heterogenen Raum, um ein Gewässer herum

### 5.1.2 Shortest Path Probleme

Die Aufgabe der angepassten RDF kann folgendermassen umschrieben werden: Konstruiere ein Polygon, in dem alle Punkte der Ebene enthalten sind, deren Abstand zum Ausgangspunkt kleiner ist als ein Schwellwert. Da die Ebene Hindernisse enthält, ist der Abstand nicht einfach die euklidische Distanz, sondern muss unter Umgehung der Hindernisse berechnet werden. Die Berechnung des kürzesten Pfades zwischen zwei Punkten ist ein Problem, dem in der *computational geometry* viel Aufmerksamkeit geschenkt wurde.

Um das *shortest path* Problem zu lösen, gibt es zwei grundlegend verschiedene Strategien, die *visibility graph method* und die *shortest path map method* (Hershberger und Suri 1993). Um den Umgehungsdistanz Algorithmus für die erweiterte RDF zu entwickeln, wurden Ideen beider Strategien verwendet.

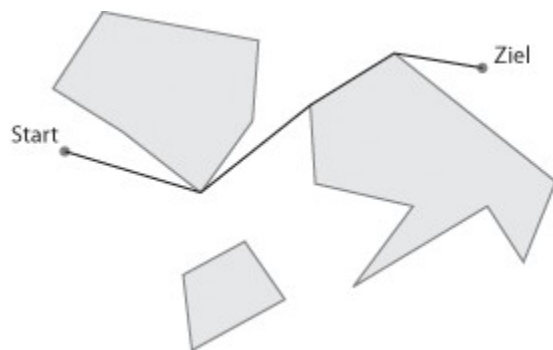


Abbildung 5.4: Kürzester Pfad zwischen zwei Punkten mit Hindernissen

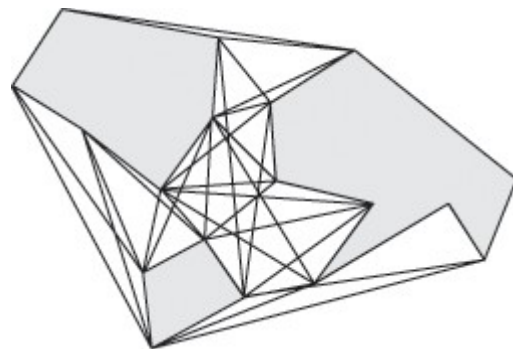


Abbildung 5.5: Visibility Graph für drei Polygone

De Berg et al. (2000) diskutieren eine *visibility graph* Methode. Die folgende Ausführung bezieht sich auf diesen Text. Die *visibility graph method* macht sich die Eigenschaft des *shortest path* Problems zunutze, dass die Punkte zwischen zwei Segmenten des kürzesten Pfades immer Eckpunkte der Hindernis-Polygone sind (Abbildung 5.4). Die Polygone müssen offene Mengen sein, das heißt, dass die Ränder der Polygone nicht zu den Polygonen gehören. Wenn die Polygone geschlossene Mengen wären, liesse sich für jeden noch so nahen Punkt ein Punkt finden, der noch näher am Polygon dran ist. Der kürzeste Pfad wird in einem sogenannten *visibility graph* berechnet. Das ist ein Graph, in dem jeweils zwei gegenseitig sichtbare Eckpunkte der Hindernisse verbunden sind (Abbildung 5.5). Ein Dijkstra Algorithmus (Dijkstra 1959) kann verwendet werden, um in diesem Graphen den kürzesten Pfad zwischen zwei Punkten zu finden.

Um den *visibility graph* zu berechnen, muss man eine Methode haben, um die Sichtbarkeit von Punkten zu bestimmen. Ein Punkt ist von einem anderen Punkt aus sichtbar, wenn die

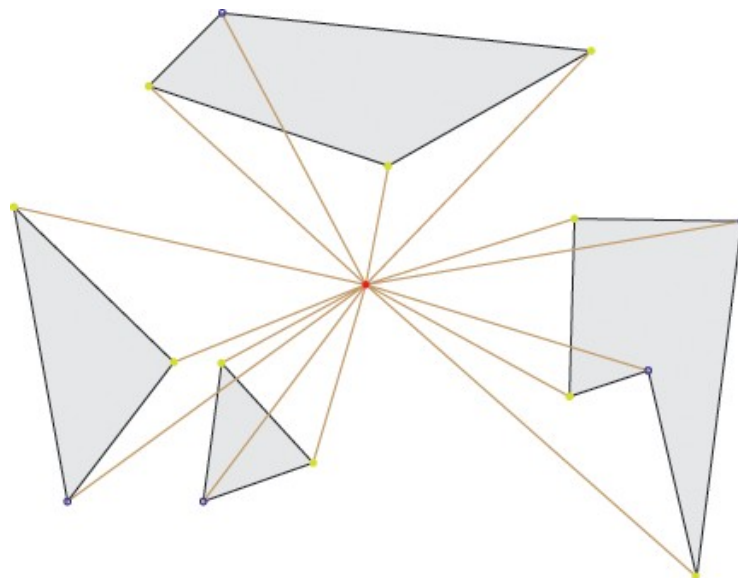


Abbildung 5.6: Sichtbarkeit von einem Punkt aus. Blau: nicht sichtbar, gelb: sichtbar (nach De Berg et al. 2000: 311)

sie verbindenden Gerade nicht durch das Inneren eines Hindernisses führt (De Berg et al. 2000: 311). Abbildung 5.6 zeigt die Sichtbarkeit von Eckpunkten mehrerer Polygone gegenüber einem Punkt in ihrer Mitte.

Bei der *shortest path map method* wird eine Datenstruktur aufgebaut, die die Ebene in Regionen aufteilt. Alle Punkte einer solchen Region haben dieselbe Sequenz von Eckpunkten in ihrem kürzesten Pfad zum Ursprung. Um diese Datenstruktur aufzubauen, kann eine simulierte Wellenfront verwendet werden (Hershberger und Suri, 1993). Von einem Punkt ausgehend breitet sie sich zwischen den Hindernissen aus (Abbildung 5.7). Die Wellenfront ist eine Menge von Kreisbögen. Zwei Ereignisse können die Topologie der Wellenfront verändern, das Aufeinandertreffen von Welle und Hindernis, sowie die Kollision zwischen zwei Wellen. Die Wellenfront wird nicht exakt berechnet, sie wird lediglich approximiert. Wenn die Wellenfront exakt berechnet würde, und die Geometrie jederzeit bekannt wäre, dann könnte das Umgehungsdistanz-Polygon mit diesem Algorithmus berechnet werden.

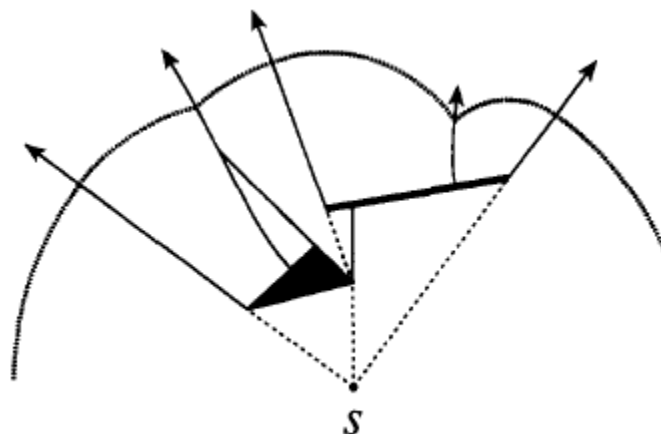


Abbildung 5.7: Propagierende Wellenfront (Hershberger und Suri 1993)

## 5.2 Der Algorithmus

Der Algorithmus benötigt folgenden Input: einen Mittelpunkt  $M$ , eine Maximaldistanz  $d$  und eine Menge Hindernis-Polygone. Die Hindernis-Polygone werden durch ein Fortbewegungskontext-Modell identifiziert. Das Resultat ist ein Umgehungsdistanz-Buffer  $U$ . Eine mögliche Konfiguration zeigt Abbildung 5.8.

Zuerst werden die relevanten Polygone ausgewählt. Polygone, deren Abstand zu  $M$  grösser ist als  $d$ , haben keinen Einfluss auf das Resultat und können daher ignoriert werden. Die gewählten Polygone werden danach in eine für den Algorithmus brauchbare Datenstruktur gebracht.

## 5 Algorithmus

Jetzt werden alle Punkte auf den Rändern der Polygone bestimmt, die einen Abstand von  $d$  zu  $M$  haben. Diese Punkte sind die äussersten Punkte, die noch in  $U$  liegen können. Diese Randpunkte werden in die Datenstruktur der Polygone an der entsprechenden Position eingefügt.

Als nächstes werden alle Eckpunkte der Polygone ausgewählt, die einen Abstand kleiner oder gleich  $d$  zu  $M$  haben. Für diese Punkte wird berechnet, ob sie von  $M$  aus sichtbar sind.

Nun werden die Eckpunkte der Polygone gesucht, die sichtbar sind, aber mindestens einen Nachbarpunkt haben, der nicht sichtbar ist. Für diese Punkte wird ein Schattenpunkt  $S$  berechnet. Dazu wird der Punkt von  $M$  aus auf das nächste Segment eines Polygons projiziert.  $S$  markiert die Stelle, an der das Segment auf der einen Seite von  $M$  aus sichtbar ist, auf der anderen Seite aber nicht (Abbildung 5.9).

Nun werden alle sichtbaren Eckpunkte und Randpunkte nach ihrem Winkel zu  $M$  sortiert. Die so sortierten Punkte werden vom Algorithmus traversiert und  $U$  wird konstruiert. Dabei werden jeweils der aktuelle Punkt  $P_n$  und der gerade zuvor behandelte Punkt  $P_{n-1}$  untersucht. Je nach Situation wird  $U$  ein Dreieck oder ein Kreissegment hinzugefügt. Dabei treten die folgenden Fälle auf:

- (1)  $P_n$  und  $P_{n-1}$  sind Nachbarn. Sie sind die beiden Ende eines Liniensegments des Randes eines Hindernis-Polygons. Ein Dreieck wird in  $U$  eingefügt. Es hat die Ecken  $P_n$ ,  $P_{n-1}$  und  $M$ . Dabei kann die Nachbarschaft auch zu einem Schattenpunkt von  $P_n$  oder  $P_{n-1}$  bestehen. Ist dies der Fall wird ein Dreieck mit dem Schattenpunkt als Ecke eingefügt.

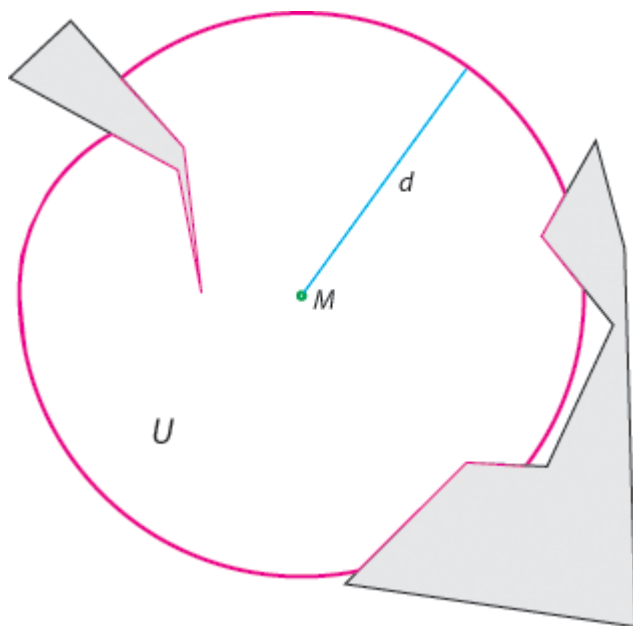


Abbildung 5.8: Ein Umgehungsdistanz-Buffer.  
Mittelpunkt  $M$ , Maximaldistanz  $d$  und das Polygon  $U$ .

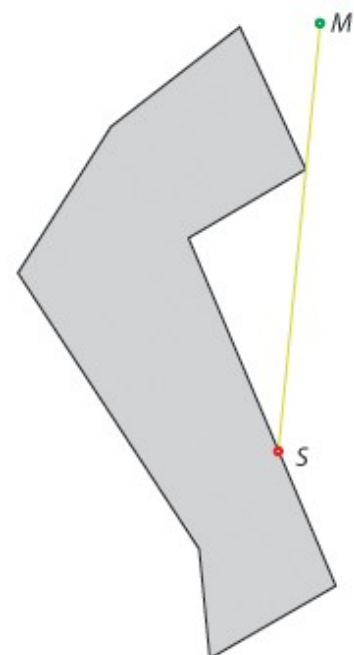


Abbildung 5.9: Schattenpunkt ( $S$ )

- (2)  $P_n$  und  $P_{n-1}$  sind keine Nachbarn.  $P_n$  und  $P_{n-1}$  sind entweder Randpunkte, haben also eine Distanz von  $d$  zu  $M$ , oder sie werfen einen Schatten, der aber nicht innerhalb der Distanz  $d$  von  $M$  aus auf ein Polygon trifft. Es wird ein Kreissegment in  $U$  eingefügt, das zwischen  $P_n$  und  $P_{n-1}$  liegt und als Zentrum  $M$  und als Radius  $d$  hat.
- (3)  $P_n$  hat mindestens einen nicht sichtbaren Nachbarpunkt. Hier kommt Rekursion zum Einsatz. Der Algorithmus ruft sich selbst auf, mit dem  $P_n$  als  $M_{\text{neu}}$  und der Differenz zwischen  $d_{\text{alt}}$  und der Distanz zwischen  $M_{\text{alt}}$  und  $M_{\text{neu}}$  als  $d_{\text{neu}}$ . Der von  $M_{\text{alt}}$  aus im Schatten liegende Teil der Ebene wird vom rekursiv aufgerufenen Algorithmus behandelt.

Die Fälle (1) und (2) können nicht gemeinsam auftreten. Entweder braucht es ein Kreissegment oder ein Dreieck. Fall (3) kann auftreten, wenn ein Kreissegment oder ein Dreieck hinzugefügt wurde. In Abbildung 5.10 ist ein Umgehungsdistanz-Buffer, in seine Bestandteile aufgeteilt, dargestellt. Die Einzelteile sind als Fall (1) und (2) markiert, und die Punkte bei denen eine Rekursion nötig ist, sind als (3) markiert.

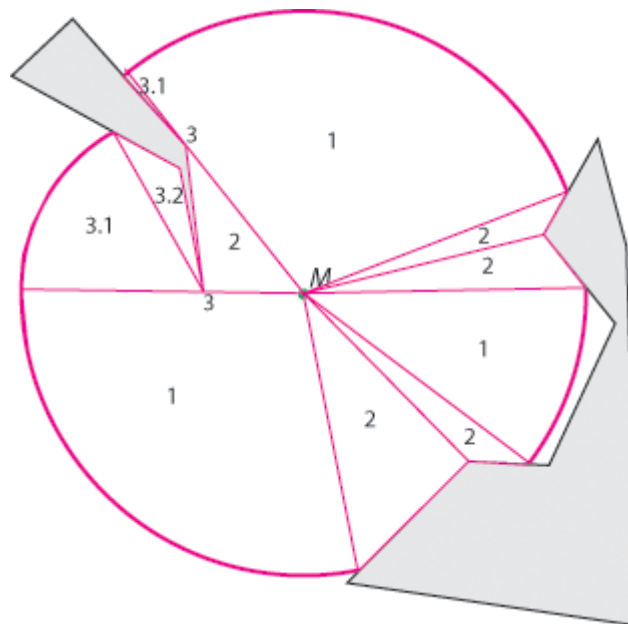


Abbildung 5.10: Umgehungsdistanz-Polygon zerlegt  
Kreissegmente (1), Dreiecke (2) und Rekursionspunkte (3)



# 6 Implementierung und Experiment

Dieses Kapitel zeigt auf, wie der im vorherigen Kapitel präsentierte Algorithmus implementiert und getestet wurde. Zuerst werden die gewählten Hilfsmittel und Technologien diskutiert. Danach wird eine erste Übersicht über die Implementierung gegeben. Die Klassen und ihre Aufgaben werden anschliessend im Detail besprochen. Eine alternative, GIS-basierte Lösung die Rasterdaten verwendet wird skizziert. Der zweite Teil dieses Kapitels bespricht, wie der Algorithmus getestet wurde, und wie Vergleichsdaten gesammelt wurden, damit qualitative und quantitative Aussagen über den Algorithmus gemacht werden können.

## 6.1 Implementierung

Um den Umgehungsdistanz-RDF Algorithmus zu implementieren, wurde die Programmiersprache Java verwendet. Java bietet mit AWT ein *framework* für die Visualisierung von zweidimensionalen Geometrien<sup>2</sup>. Umfangreiche *open source* Programmbibliotheken für zweidimensionale Geometrie sind ebenfalls erhältlich. Dieser Abschnitt gibt eine Übersicht über die Implementierung einer analytischen Lösung der Umgehungsdistanz-RDF.

### 6.1.1 Die Java Topology Suite

Die Implementierung eines komplexen räumlichen Algorithmus setzt viele grundlegende räumliche Algorithmen voraus. Damit solche Basismethoden, wie das Auffinden des Schnittpunkts zwischen zwei Linien, nicht extra programmiert werden mussten, wurde nach einer Programmbibliothek gesucht, die solche Methoden und Klassen anbietet. Mit der *Java Topology Suite* (JTS) wurde eine solche Bibliothek gefunden. JTS ist eine *open source* Java-Bibliothek. JTS bietet Klassen an, die einfache Geometrien im zweidimensionalen euklidischen Raum repräsentieren (Abbildung 6.1). Zusätzlich sind fundamentale räumliche Algorithmen implementiert. JTS wird unter anderem in der Open Source GIS Anwendung OpenJUMP verwendet<sup>3</sup>. In dieser Arbeit wurde die Version 1.10 von JTS<sup>4</sup> verwendet.

Die grundlegende Geometrieklasse `Geometry` ist eine abstrakte Klasse, was bedeutet, dass keine Instanzen davon gemacht werden können. Operationen, die für alle Geometrieklassen zur Verfügung stehen, wie z.B. eine `buffer` und eine `union` Operation sind hier definiert. Punkt, Linie und Fläche werden durch die Klassen `Point`, `LineString` und

---

<sup>2</sup> <http://java.sun.com/j2se/1.4.2/docs/api/java/awt/Graphics2D.html> Zugriff: 11 August 09

<sup>3</sup> <http://tsusiatsoftware.net/> Zugriff: 17 Juni 09

<sup>4</sup> <http://sourceforge.net/projects/jts-topo-suite/files/> Zugriff: 27 Mai 09

## 6 Implementierung und Experiment

Polygon repräsentiert. MultiPoint, MultiLine und MultiPolygon sind jeweils ein Array mit mehreren Punkten, Linien oder Flächen. Die GeometryCollection ist ein Array beliebiger Geometrien. Eine geschlossene Linie, also eine Linie deren Anfangs- und Endpunkt identisch sind, wird durch die Klasse LinearRing repräsentiert.

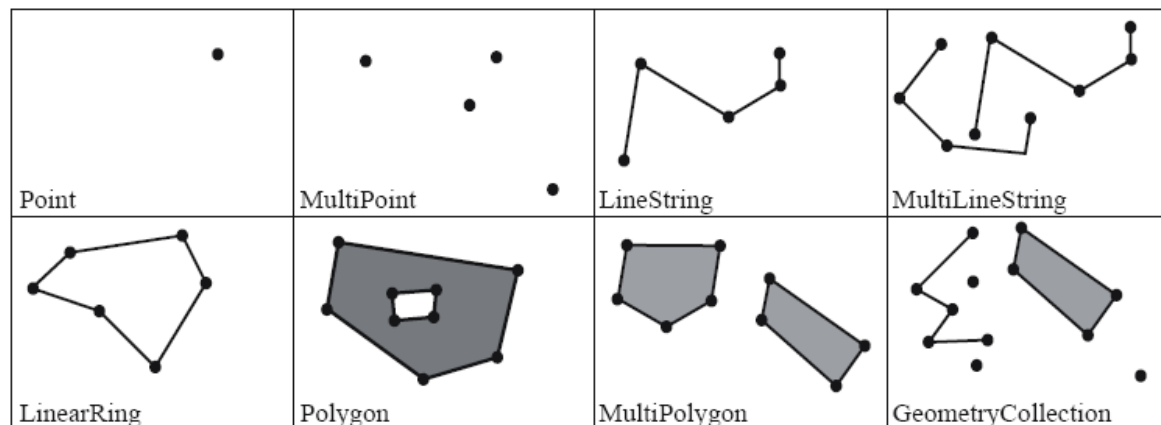


Abbildung 6.1: Geometrieklassen der JTS (aus Schmid, 2008)

### 6.1.2 Architektur

Ein Ziel bei der Implementierung war, eine einzige Klasse zu haben, die den Ablauf des Algorithmus steuert. Diese Klasse liefert auch das Resultat. Das Zustandekommen des Resultats ist aber Aufgabe vieler Klassen, die jeweils ein spezifisches Problem lösen.

Es wurde zwischen drei Arten von Aufgaben unterschieden. Erstens die Klassen, die den Ablauf des Algorithmus selbst steuern. Diese liefern das Resultat. Sie sind im *package algorithm* gesammelt. Zweitens gibt es Klassen, die ein bestimmtes Problem behandeln. Ein solches Problem ist zum Beispiel, festzustellen, ob ein Punkt von einem anderen Punkt aus sichtbar ist. Diese Klassen sind in das *package tools* eingegliedert. Die dritte Art von Aufgaben ist das Bereitstellen von Datenstrukturen für den Algorithmus. Klassen, die sich dieser Aufgabe annehmen, sind im *package structure* zu finden.

Das Klassendiagramm in Abbildung 6.2 zeigt die Klassen des Algorithmus und die Verbindungen dieser Klassen untereinander. Der Übersicht halber sind nur einige wenige exemplarische Methoden und Attribute aufgelistet.

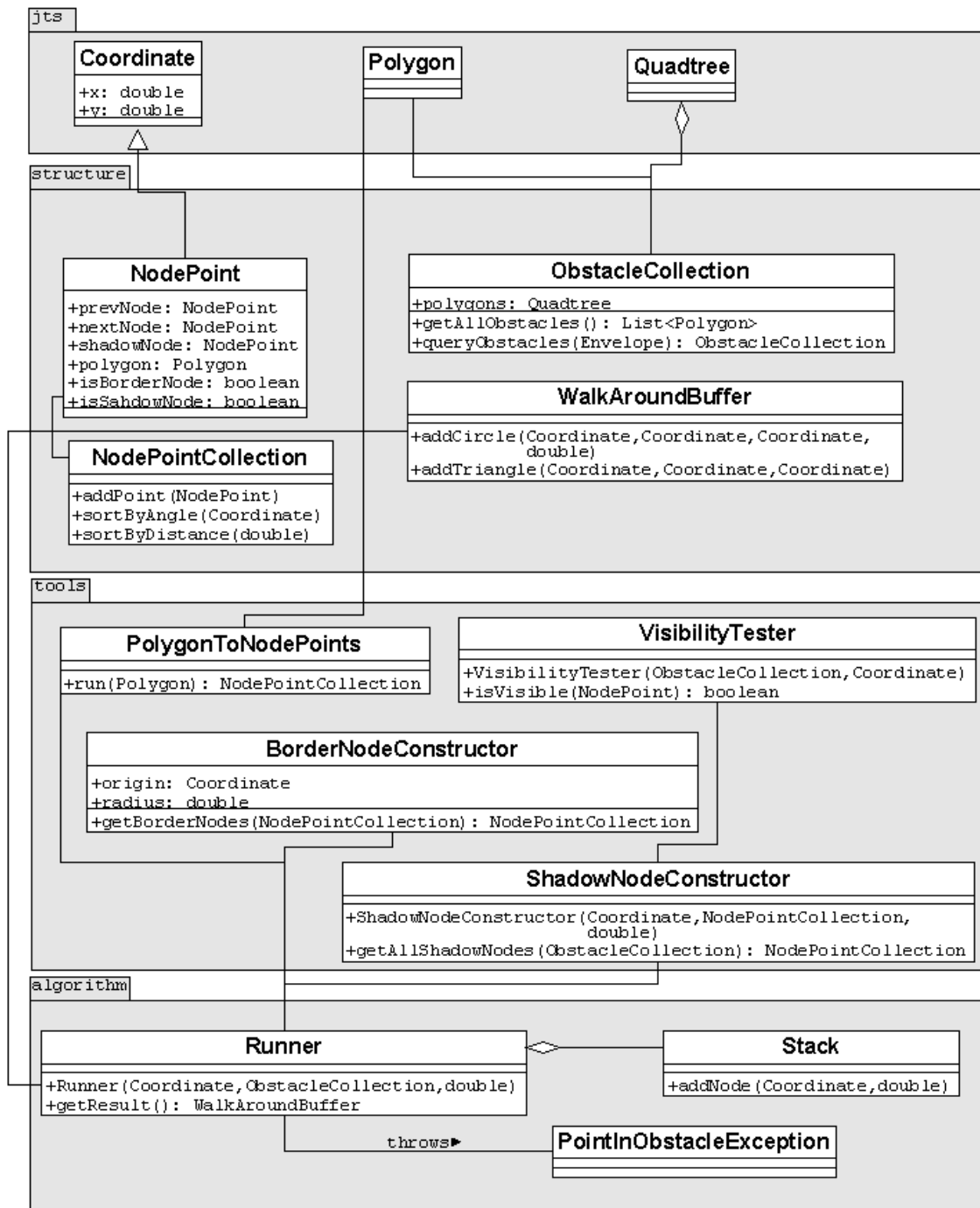


Abbildung 6.2: Klassendiagramm mit ausgewählten Attributen und Methoden

### 6.1.3 Die Klassen und ihre Aufgaben

In diesem Abschnitt werden die einzelnen Klassen und ihre Funktion im Algorithmus diskutiert. Zu jeder Klasse gibt es einen kurzen Text, der die wichtigsten Methoden und Attribute nennt und ihre Funktionsweise erläutert.

#### **structure.NodePoint**

Der `NodePoint` erbt von der JTS-Klasse `Coordinate`. Er hat also wie die `Coordinate` einen  $x$ - und einen  $y$ -Wert und kann in der JTS überall dort eingesetzt werden, wo auch eine `Coordinate` eingesetzt werden kann. Der Name `NodePoint` wurde gewählt, weil die Klasse die Eigenschaften eines Knotens in einem Graphen hat, aber gleichzeitig den Eckpunkt eines Polygons repräsentiert. Der `NodePoint` ist das wichtigste Element in der Datenstruktur des Algorithmus. Als Knoten eines Graphen hat der `NodePoint` zwei benachbarte `NodePoints`, diese sind der vorherige und nächste Eckpunkt im Polygon. Die `NodePoints` eines Polygons bilden eine zweifach verlinkte Liste (Ottmann und Widmayer, 2002). Der `NodePoint` hat weitere Eigenschaften, die im Zusammenhang mit jenen Klassen diskutiert werden, die diese verwenden.

#### **structure.NodePointCollection**

Die `NodePointCollection` ist eine Klasse, die `NodePoints` verwaltet und Methoden für den Zugriff darauf zur Verfügung stellt. Die `NodePoints` können nach Distanz zu einer Koordinate oder nach Winkel zu einer Koordinate sortiert werden. Das Sortieren ändert die Verkettung zwischen den Nodes nicht. Des Weiteren stellt die `NodePointCollection` eine Methode zur Verfügung, die als Rückgabewert eine neue `NodePointCollection` mit allen Punkten hat, die weniger weit als eine angegebene Maximaldistanz von einer Koordinate entfernt ist.

#### **structure.ObstacleCollection**

Hindernisse werden durch die JTS-Klasse `Polygon` repräsentiert. Die `ObstacleCollection` verwaltet diese Polygone. Innerhalb der `ObstacleCollection` sind die Polygone in einem `Quadtree` gespeichert. Der `Quadtree` ist eine in der JTS implementierte Datenstruktur. Ein `Quadtree` erlaubt die Abfrage mit einem *minimum boundig rectagle* (MBR). Polygone, die innerhalb dieses MBR liegen oder dieses schneiden, sind das Resultat einer solchen Abfrage. Dies erlaubt einen schnellen Zugriff auf alle relevanten Hindernis-Polygone.

#### **structure.WalkAroundBuffer**

Der `WalkAroundBuffer` repräsentiert das Ergebnis des Algorithmus. Dabei wird der Umgehungsdistanz-Buffer aus einzelnen Teilen zusammengesetzt. Der `WalkAround-`

`Buffer` hat Methoden, um dem Resultat solche Einzelteile zuzufügen. Die Methode `addTriangle` kann mit drei Punkten als Argument aufgerufen werden. Sie fügt dem Resultat ein Dreieck mit dessen Eckpunkten hinzu. Eine zweite Methode hat als Argument ebenfalls drei Punkte, aber zusätzlich auch eine `double`-Variable. Diese Methode erweitert das Resultat um einen Kreisbogen. Der `double`-Wert ist der Radius des Kreisbogens, der erste Punkt ist der Mittelpunkt und die zwei weiteren Punkte bestimmen den Winkel. Der `WalkAroundBuffer` hat auch eine Methode, die als Rückgabewert das resultierende Polygon liefert. Um die Einzelteile zu vereinen wird die `buffer`-Methode verwendet. Ein *buffer* mit Abstand null wird um alle Einzelteile zusammen konstruiert. Dies hat den gleichen Effekt wie das iterative Anwenden der `union`-Funktion, ist aber schneller.

### **`tools.PolygonToNodePoints`**

Diese Klasse stellt eine Methode zur Verfügung, die aus einem `JTS-Polygon` `NodePoints` erstellt. Aus den Eckpunkten der `JTS-Polygone`, die als Sequenz von Koordinaten vorliegen, werden `NodePoints` erstellt. Dabei werden die `NodePoints` untereinander verlinkt, so dass aufeinander folgende Eckpunkte des Polygons als `Next`- und `PreviousNode` in den `NodePoints` abgespeichert werden. Das Resultat ist eine zweifach verlinkte Liste von `NodePoints`, die in Form einer `NodePointCollection` zurückgegeben wird.

### **`tools.BorderNodeConstructor`**

Der `BorderNodeConstructor` hat die Aufgabe zusätzliche `NodePoints` zu konstruieren. Diese `NodePoints` sind am äussersten möglichen Rand des resultierenden Polygons. Sie liegen auf einem Kreis mit dem gleichen Mittelpunkt wie das Umgehungsdistanz-Polygon, und mit einem Radius gleich der `Maximaldistanz` (Abbildung 6.3). Der `BorderNodeConstructor` erstellt einen solchen Kreis und verschneidet ihn mit den Segmenten der Polygonränder. Aus den Schnittpunkten werden neue `NodePoints`, die zwischen bestehenden `NodePoints` eingefügt werden. Dies ist gleichbedeutend mit dem Einfügen eines zusätzlichen Eckpunktes auf einem Segment eines Polygons.

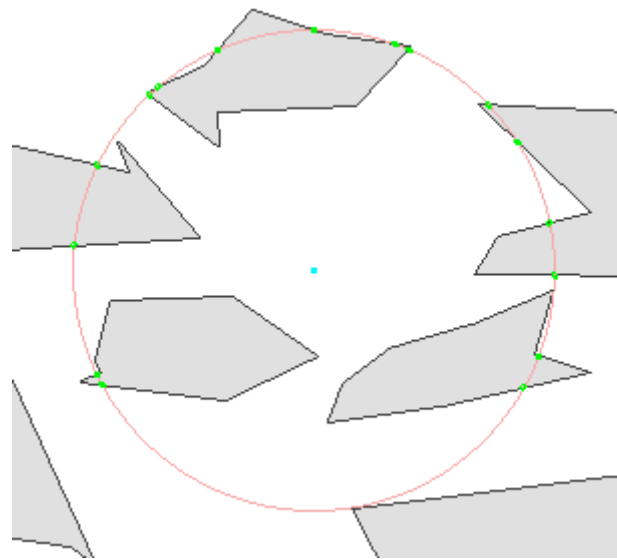


Abbildung 6.3: BorderNodes (grün)

### **tools.VisibilityTester**

Der `VisibilityTester` wird mit einer `ObstacleCollection` und einer `Coordinate`, dem Ausgangspunkt, initialisiert. Danach kann für `NodePoints` getestet werden, ob diese von der Quelle aus sichtbar sind. Um dies festzustellen, wird eine Linie von der Quelle zum `NodePoint` gezogen. Dann werden die Hindernis-Polygone mit dieser Linie verschnitten und die Schnittpunkte werden bestimmt. Wenn ein Schnittpunkt gefunden wird, der zwischen Anfang und Ende der Linie liegt, steht fest, dass ein Hindernis zwischen den zwei Punkten liegt und der `NodePoint` nicht sichtbar ist.

Als Sonderfall gilt es zu beachten, dass benachbarte `NodePoints` jeweils voneinander aus sichtbar sein sollten. Das heisst, dass der Rand der Polygone beim Testen der Sichtbarkeit nicht zum Polygon zählen sollte. Die Polygone müssen also, wie schon in Abschnitt 5.1.2 besprochen, mengentheoretisch gesehen offene Mengen sein, die Punkte beliebig nahe an ihrem Rand beinhalten, nicht aber den Rand selbst.

Dies wird erreicht, indem geprüft wird, ob der `NodePoint` einen Nachbarn hat, der räumlich gleich ist wie der Quellpunkt. Wenn dies der Fall ist, ist der `NodePoint` auf jeden Fall sichtbar. Der oben beschriebene Test mit der verbindenden Linie wird nicht ausgeführt.

### **tools.ShadowNodeConstructor**

Der `ShadowNodeConstructor` findet sichtbare `NodePoints`, die einen unsichtbaren Nachbar-`NodePoint` haben. Diese `NodePoints` und ihre angrenzenden Segmente können einen „Schatten“ auf Segmente werfen. In diesem Schatten sind `NodePoints` und Teile von Segmenten oder ganze Segmente nicht sichtbar. Der `ShadowNodeConstructor`

`tor` berechnet den Punkt auf einem Segment, an dem solch ein Schatten aufhört (Abbildung 6.4). Wenn ein solcher `NodePoint` keinen Schatten auf ein anderes Segment wirft, wird der Grenzpunkt des Schattens auf dem Kreis, der die Maximaldistanz repräsentiert, berechnet.

Ein `NodePoint`, der einen Schatten wirft, wird gekennzeichnet, weil diese Information für den Algorithmus von Bedeutung ist. Der Grenzpunkt des geworfenen Schattens ist auch ein `NodePoint`. Er ist als `isShadowNode` gekennzeichnet. Ein `NodePoint` kann also ausser zwei Nachbarn auch noch einen `ShadowNode` haben.

Ein Schattenpunkt wird folgendermassen berechnet:

1. Es wird eine Linie vom `NodePoint` senkrecht auf den Maximaldistanz-Kreis erstellt.
2. Alle Schnittpunkte der Linie mit Polygonsegmenten werden berechnet.
3. Der Schnittpunkt, der am nächsten beim `NodePoint` ist, aber nicht räumlich identisch mit ihm, ist der gesuchte Schattenpunkt.
4. Gibt es keinen solchen Schnittpunkt, wird ein Schatten auf dem Maximaldistanz-Kreis berechnet.

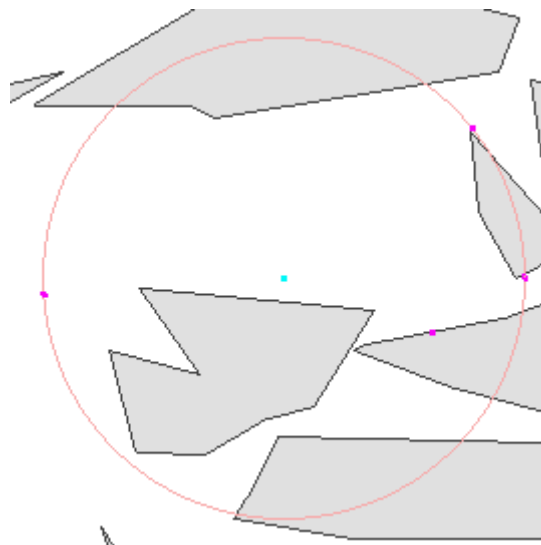


Abbildung 6.4: *ShadowNodes (pink)*

### **algorithm.Runner**

Der `Runner` ist das Herzstück des Algorithmus. Ein `Runner`-Objekt wird mit einem Mittelpunkt (`Coordinate`), den Hindernissen (`ObstacleCollection`) und einem Radius (`double`) initialisiert. Das `Runner`-Objekt benutzt die `PolygonToNodePoints`, die `BorderNodeConstructor` und die `ShadowNodeConstructor` Klassen, um alle be-

## 6 Implementierung und Experiment

---

nötigten `NodePoints` zu erstellen. Diese werden in einer `NodePointCollection` gesammelt. Ein `WalkAroundBuffer`-Objekt speichert und verwaltet das Resultat-Polygon.

Als erstes werden alle `NodePoints` durchlaufen. Für sichtbare `NodePoints` wird dabei überprüft, ob die Nachbarn auch sichtbar sind. `NodePoints`, die Schattenpunkte sind, sind per Definition sichtbar. Die Kombinationen von verschiedenen `NodePoints`, die auftreten können, sind in Abbildung 6.5 ersichtlich. Es sind dies (1) `ShadowNode` und normaler `NodePoint`, (2) zwei normale `NodePoints` und (3) zwei `ShadowNodes`. Wenn einer dieser drei Fälle eintritt, wird dem `WalkAroundBuffer` ein Dreieck hinzugefügt. Dies ist eine *brute-force* Lösung, gewisse Punktkombinationen werden doppelt überprüft. Durch sortieren der `NodePoints` nach Winkel könnten die Dreiecke effizienter gefunden werden. Die *brute-force* Lösung ist aber einfacher zu programmieren und zu testen.

Nachdem alle Dreiecke gefunden wurden, kümmert sich der `Runner` um die Kreissegmente. Dabei wird die `NodePointCollection` nach dem Winkel zum Mittelpunkt sortiert und so durchlaufen. Es werden immer zwei aufeinander folgende `NodePoints` betrachtet. Wenn beide `NodePoints` entweder `BorderNodes` sind oder einen Schatten auf den Maximaldistanz-Kreis werfen, muss ein Kreissegment zwischen ihnen konstruiert werden. Es gibt einige Sonderfälle, die es zu beachten gilt. Ein Sonderfall tritt auf, wenn die beiden betrachteten `NodePoints` `BorderNodes` sind. Dann ist es möglich, dass sie Nachbarn sind. Ist dies der Fall, wurde bereits ein entsprechendes Dreieck konstruiert. Der zweite Sonderfall tritt auf, wenn der Mittelpunkt selbst ein `NodePoint` ist, und als solcher der Nachbar-Node der beiden überprüften `NodePoints`.

Das bisher erstellte Polygon deckt den vom Mittelpunkt aus sichtbaren Bereich des Umgehungs-Polygons ab (Abbildung 6.6). Nun wird für jeden `NodePoint`, der einen Schatten wirft, ein neues `Runner`-Objekt erstellt. Der `NodePoint` ist der neue Mittelpunkt. Die Maximaldistanz wird um die Entfernung zum alten Mittelpunkt verkleinert. Durch das Zusammenfügen der Resultate aller `Runner`-Objekte kommt das Umgehungs-Polygon zustande.

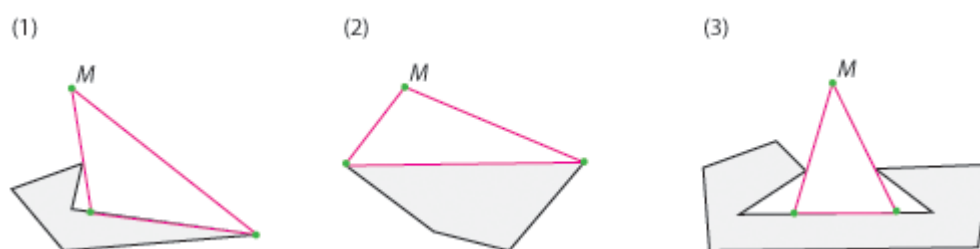


Abbildung 6.5: Konfigurationen beim Hinzufügen eines Dreiecks

**algorithm.Stack**

Der `Stack` sorgt dafür, dass die Berechnungen der Resultate für die Mittelpunkte in der Rekursion der `Runner`-Objekte in einer geeigneten Reihenfolge ablaufen. Der `Stack` verwaltet alle globalen Informationen, die benötigt werden, um zu verhindern, dass in der Rekursion eines `Runner`-Objekts derselbe Punkt mehrfach behandelt wird. Es wird dafür gesorgt, dass jeweils das `Runner`-Objekt mit der grössten Maximaldistanz zuerst ausgeführt wird.

**algorithm.PointInObstacleException**

Diese Ausnahme entsteht wenn der Punkt von dem aus der `buffer` berechnet werden sollte in einem der Hindernisse liegt. In diesem Fall kann kein `buffer` berechnet werden. Es ist dem Benutzer überlassen wie mit dieser Ausnahme umgegangen wird. Wenn die Distanz zum Rand des Polygons klein ist, könnte der Punkt an den Rand verschoben werden.

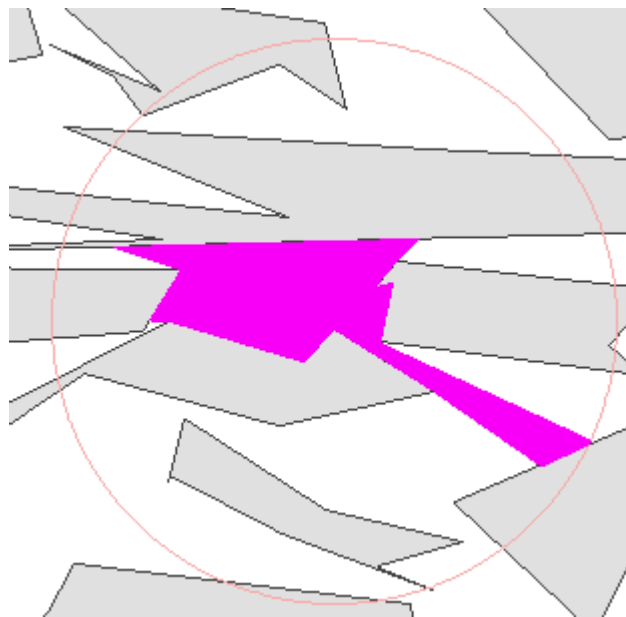


Abbildung 6.6: Resultat eines `Runner`-Objekts ohne Rekursion (pink).

### 6.1.4 Rasterlösung

Ein Umgehungsdistanz-Buffer kann auch rasterbasiert berechnet werden. Die dafür benötigten Operationen sind in marktüblichen GIS vorhanden. Hier wird aufgezeigt, wie ein Umgehungsdistanz-Buffer mit den *Spatial Analyst Tools* in der ArcToolbox erstellt werden kann. Dazu benötigt werden ein Rasterdatensatz und ein Punkt. Im Rasterdatensatz sind die Hindernis-Polygone codiert. Der Punkt hat dabei entweder in Raster- oder Vektorform. Er repräsentiert den Mittelpunkt der RDF. Zur Berechnung des *buffers* sind folgende Schritte nötig:

- (1) Aus dem Raster mit den Hindernissen wird eine Kostenoberfläche erstellt. In dieser Kostenoberfläche hat jede Zelle einen Wert, der angibt, was es kostet eine Distanzeinheit in dieser Zelle zurückzulegen (Abbildung 6.7). Jeder Zelle wird mittels der *Reclassify* Funktion ein neuer Wert zugeordnet. Den Hinderniszellen wird ein hoher Wert zugeordnet. Dieser sollte höher sein als die Maximaldistanz des Umgehungsdistanz-Buffers. In Hinderniszellen eine Distanzeinheit zurückzulegen, kostet also mehr als der maximale Wert, der noch im Buffer liegt. Den Zellen ohne Hindernisse wird ein Wert von 1 zugeordnet.
- (2) Die Kostenoberfläche und ein Mittelpunkt in Vektor oder Rasterform werden als Input für die *Spatial Analyst* Funktion *Cost Distance* verwendet. Diese Funktion berechnet für jede Zelle, was es kostet, diese Zelle von einem Punkt aus zu erreichen. Dabei werden Zellen entweder gerade oder diagonal durchquert. Für eine gerade Durchquerung ist die Weglänge die Zellgröße, für eine diagonale Durchquerung  $\sqrt{2}$  mal die Zellgröße. Die Weglänge wird mit

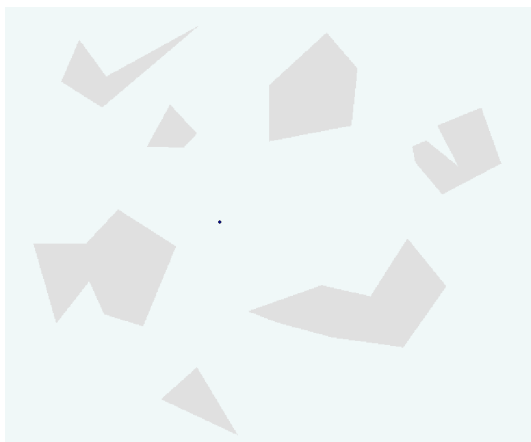


Abbildung 6.7: Hindernisse als Rasterdaten

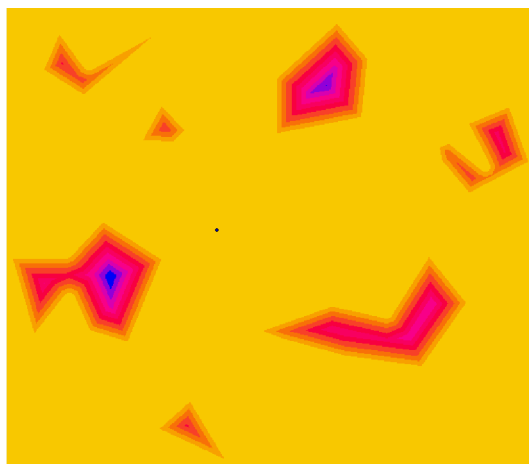


Abbildung 6.8: Kostenoberfläche  
Tiefe Werte gelb, hohe Werte blau.

dem Kostenwert für die Zelle multipliziert. In unserem Fall ist dieser Wert entweder 1 (kein Hindernis) oder sehr gross (Hindernis). Das Resultat ist ein Raster, in dem in jeder Nicht-Hinderniszelle ihre Distanz zum Mittelpunkt steht (Abbildung 6.8). Hinderniszellen haben einen Wert, der so hoch ist, dass die Zelle unmöglich zur Umgehungsdistanz-RDF gehören kann.

- (3) Durch die *Reclassify* Funktion kann der Distanz-Raster in Distanzklassen für die RDF umgewandelt werden. Dabei wird jeweils einem Intervall von Werten eine Zahl zugeordnet, die die Klasse identifiziert.

Abbildung 6.9 zeigt eine mit der Rasterlösung berechnete RDF. Dabei fällt die oktagonale Form der einzelnen Ringe auf. Diese sind eine Folge davon, dass eine Zelle nur entweder gerade oder diagonal durchquert werden kann. Der Weg zu einer Zelle besteht daher nur aus waagrechten, senkrechten und um  $45^\circ$  geneigten Stücken.

Alternativ zur Funktion *Cost Distance* kann auch die *Path Distance* verwendet werden. Diese unterscheidet sich darin, dass neben horizontalen auch vertikale Distanz und Kosten berücksichtigt werden. So kann der Höhenunterschied zwischen zwei Zellen in die Berechnung der Distanz für die Umgehungsdistanz-RDF miteinbezogen werden.

Bei der Rasterlösung ist es einfach, komplizierte Kostenoberflächen zu verwenden. So kann der Raum auch anders eingeteilt werden als in Hindernisse und nicht-Hindernisse. Oberflächen, auf denen die Bewegung weniger effizient ist, können höhere Kostenwerte im Raster erhalten. Wenn sehr viel über die Bewegungskapazität der Untersuchungsobjekts bekannt ist, kann eine solche Lösung wünschenswert sein.

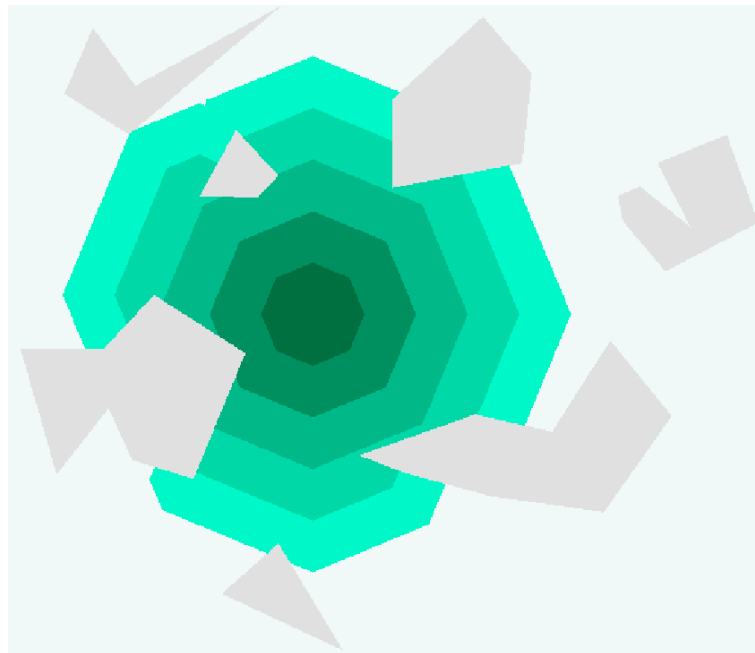


Abbildung 6.9: Rasterlösung für die Umgehungsdistanz-RDF

### 6.2 Experiment

Der implementierte Algorithmus wurde ausgiebig getestet. Um Hindernis-Polygone zu generieren wurde eine Klasse verwendet, die zufällige, topologisch korrekte Polygone mit acht Ecken erstellte. Dabei variierte die Anzahl der Polygone sowie auch deren Grösse. Um die Resultate des Algorithmus zu überprüfen, wurden diese visualisiert. Der Umgebungsdistanz-Buffer und die Hindernisse wurden gemeinsam visuell dargestellt. Dafür wurde das Java-AWT *framework* verwendet.

Die Klassen für die Generierung der Hindernis-Polygone und für die Visualisierung bildeten auch die Grundlage für ein Experiment. Für eine Konfiguration von Hindernissen wurden jeweils die Ringe für eine RDF mit Umgebungsdistanz und eine RDF mit euklidischer Distanz berechnet. Diese wurden visuell und numerisch verglichen.

Das Ziel des Experiments ist es, Unterschiede zwischen den beiden Varianten der RDF aufzuzeigen. Diese Unterschiede werden auch an zufälligen Hindernis-Konfigurationen ersichtlich. Selbst entworfene Geometrien sind geeignet, um spezielle Fragestellungen zu beantworten, die sich aus den Vergleichen mit zufälligen Hindernissen ergeben. Für eine Analyse der geometrischen Eigenschaften der Umgebungsdistanz-RDF werden daher keine echten Bewegungs- und Kontextdaten benötigt.

Ein Vergleich zwischen den zwei Varianten benötigt folgenden Input:

- (1) Eine Hindernis-Konfiguration;
- (2) einen Mittelpunkt für die RDF;
- (3) eine Menge von Maximaldistanzen für die Bildung der Distanzklassen.

Die Hindernis-Konfiguration wurde entweder zufällig erstellt oder für eine spezielle Fragestellung entworfen. Die zufälligen Polygone haben jeweils acht Ecken und variieren in Form und Grösse. Die Anzahl der Polygone wurde von Vergleich zu Vergleich verändert, und so nimmt auch der Anteil von Hindernissen an der Gesamtfläche verschiedene Grössen an. Die Hindernis-Konfigurationen für spezifische Fragestellungen wurden von Hand entworfen. Für jede solche Konfiguration wurde eine Methode geschrieben, die die entsprechende `ObstacleCollection` erstellt. Wenn die Hindernisse zufällig sind, ist es nicht nötig den Mittelpunkt auch zufällig zu bestimmen. Auch wenn der Mittelpunkt in jedem Vergleich an der selben absoluten Position ist, sind die Polygone um ihn herum zufällig. Die Platzierung des Mittelpunkts relativ zu den Hindernissen ist bei den spezifischen Konfigurationen ein Teil des Entwurfes. Die verwendeten Maximaldistanzradien waren jeweils 40, 80, 120, 160 und 200 Pixel, bei einer Gesamtausdehnung des Ausschnitts von 1000 mal 600 Pixeln.

Berechnet wurden pro Versuch die folgenden Resultate:

- (1) Eine Visualisierung der Umgehungsdistanz-RDF zusammen mit den Hindernissen (Abbildung 6.10).
- (2) Eine Visualisierung einer Vergleichs-RDF zusammen mit den Hindernissen (Abbildung 6.11).
- (3) Eine Ausgabe der Fläche der Ringe der RDF.

Um die Ringe der Umgehungsdistanz-RDF zu konstruieren, wird der Algorithmus mit mehreren Maximaldistanzen ausgeführt. Der Mittelpunkt bleibt dabei jeweils derselbe. Der Unterschied zwischen zwei Umgehungsdistanz-Buffern mit aufeinander folgenden Radien ist jeweils ein Ring der Umgehungsdistanz-RDF. Für die Berechnung der Fläche der Ringe konnte die *area*-Methode der JTS-Klasse `Polygon` verwendet werden.

Die RDF für den Vergleich entspricht nicht ganz der von Imfeld (2000) vorgestellten Version. Um die Vergleichbarkeit mit der Umgehungsdistanz-RDF zu gewährleisten, gehören die Hindernisse nicht zu den Ringen der Vergleichs-RDF. Es existiert in JTS keine Klasse, die einen Kreis repräsentiert. Ein Kreis kann in JTS erstellt werden, indem ein *buffer* um einen Punkt herum berechnet wird. Die Hindernisse können mit der *difference*-Funktion aus einem Kreis entfernt werden. Auch für die Vergleichs-RDF gilt, dass der Unterschied zweier aufeinander folgende Kreise einem Ring der RDF entspricht.

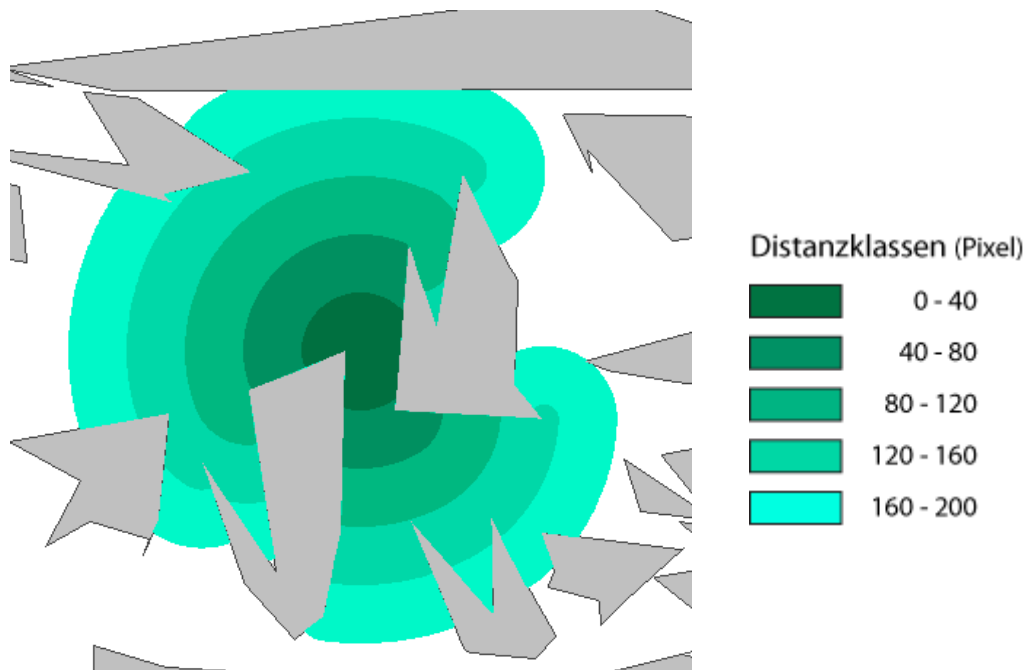


Abbildung 6.10: Umgehungsdistanz-RDF und Hindernisse

Einige der erstellten visuellen Vergleiche, sowie die entsprechende Verteilung der Flächen der Ringe, werden in Kapitel 7 diskutiert.

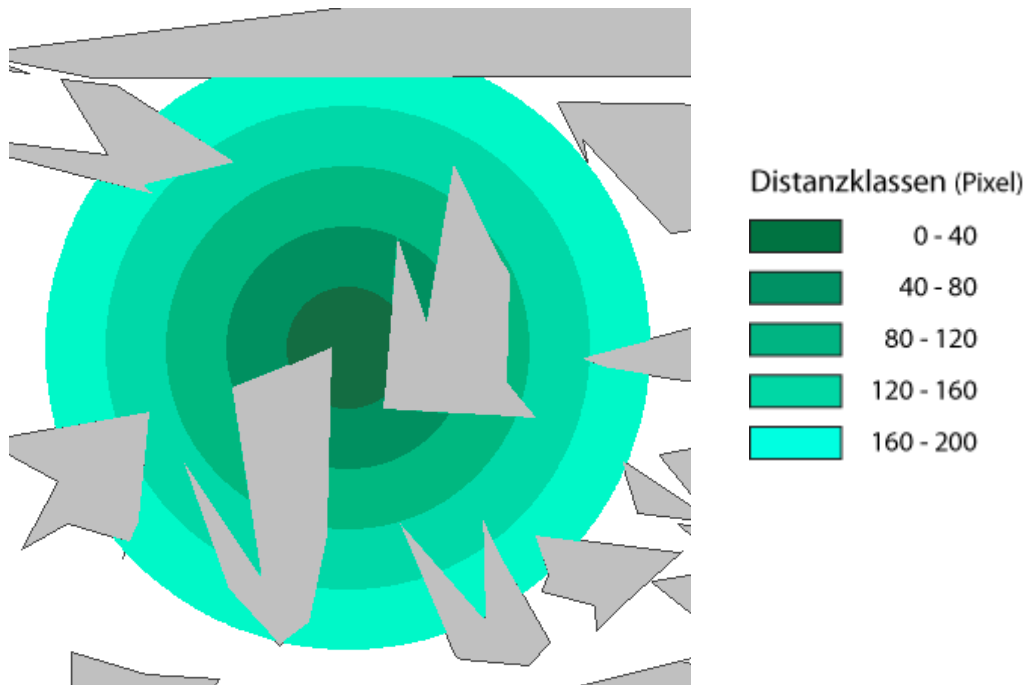


Abbildung 6.11: Vergleichs-RDF und Hindernisse

# 7 Resultate

In diesem Kapitel werden Resultate des Experiments mit dem Umgehungsdistanz-Buffer Algorithmus vorgestellt. Dabei werden jeweils drei Visualisierungen verwendet. Zwei davon sind die Umgehungsdistanz-RDF (U-RDF) und die Vergleichs-RDF. Die Erstellung und Visualisierung der beiden RDF-Varianten wurde bereits in Abschnitt 6.2 diskutiert. Die dritte Visualisierung ist ein Diagramm der Flächen der einzelnen Ringe. Zusätzlich ist jeweils die Anzahl Hindernisse aufgeführt. Der Anteil der Hindernisse an der Fläche des grössten Kreises der Vergleichs-RDF ist in Prozent angegeben.

Die Polygone in den ersten fünf Vergleichen wurden zufällig erstellt. Dabei wurde die Anzahl der Polygone, und damit deren Anteil an der Fläche der RDF variiert. Die Polygone in den letzten vier Vergleichen wurden konstruiert, um spezielle Fragestellungen zu beantworten.

## 7.1 Zufällige Konfigurationen

Die hier präsentierten Visualisierungen wurden aus der Gesamtheit der berechneten Vergleiche ausgewählt, weil in ihnen bestimmte geometrische Phänomene des Umgehungsdistanz-Buffers sichtbar sind.

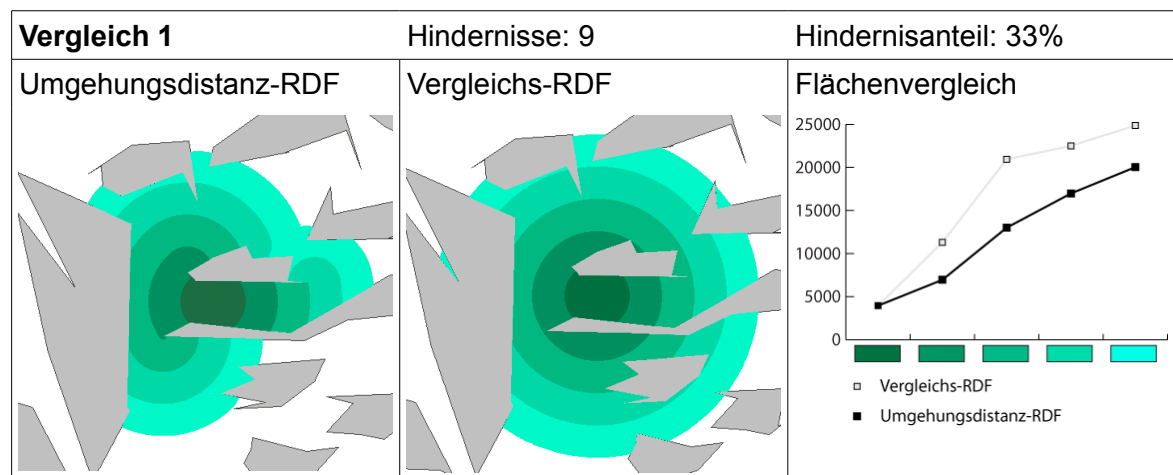


Abbildung 7.1: Vergleich 1 – Zufällige Hindernisse

In Vergleich 1 ist der Unterschied in der linken Bildhälfte klein, in der rechten Hälfte aber beachtlich. Die U-RDF wird nach rechts hin zwischen zwei Polygonen kanalisiert. Dies führt zu einer zuerst viel flacheren Kurve im Vergleichsdiagramm der Flächen. Erst ab dem vierten Ring findet allmählich eine Ausbreitung statt. Nach links hin findet keine solche Kanalisierung statt. Das grosse Hindernis auf der linken Seite hat einen ähnlichen Effekt auf beide Varianten.

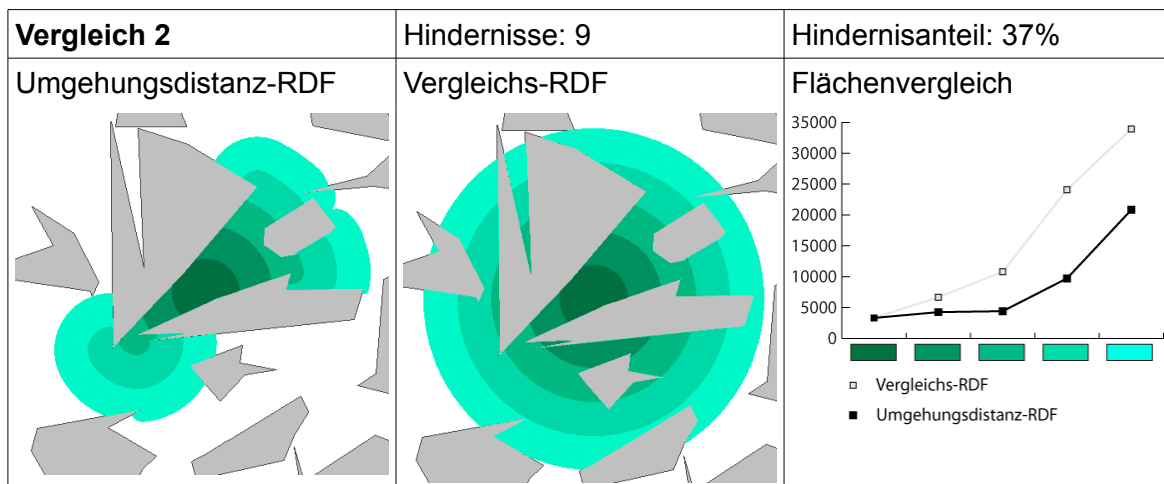


Abbildung 7.2: Vergleich 2 – Zufällige Hindernisse, grosse Hindernis-Fläche

Von den hier präsentierten Vergleichen ist Vergleich 2 derjenige mit dem grössten Unterschied zwischen den zwei Varianten. Die Fläche der U-RDF ist nur 61.4% derjenigen der Vergleichs-RDF. Der Mittelpunkt liegt zwischen zwei Polygonen, deren längste Achse tangential zu den Kreisen liegt. So wird die U-RDF zwischen diesen Hindernissen kanalisiert. Beide Kurven der Flächen steigen daher zuerst nur langsam an. Ab dem vierten Kreis nimmt die Fläche dann schnell stark zu. Punkte im Raum direkt hinter den beiden grossen Polygonen haben eine hohe Umgehungsdistanz. Die euklidische Distanz zu diesen Punkten ist aber gering.

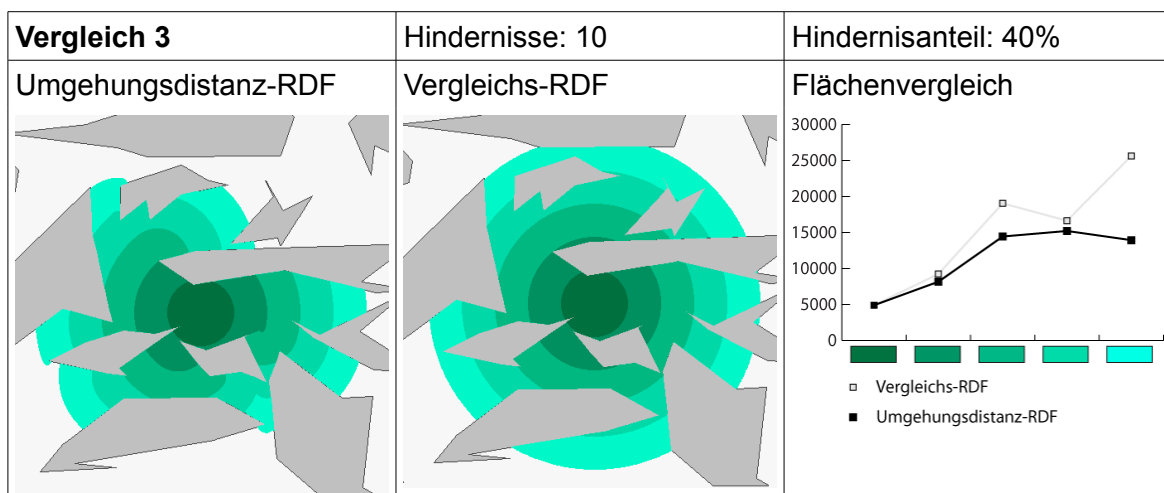


Abbildung 7.3: Vergleich 3 – Zufällige Hindernisse, hohe Dichte

Mit 40% hat Vergleich 3 von den hier präsentierten Resultaten den höchsten Hindernisanteil. Schön zu sehen ist hier, wie die U-RDF um die kleineren Polygone herum fließt. Die beiden Teile vereinigen sich hinter einem nahen Hindernis wieder. Die kleinen Polygone haben aber keinen sehr grossen Einfluss auf die Form der U-RDF. Es ist wieder ein langes, dünnes Polygon, das für den visuell auffälligsten Unterschied verantwortlich ist.

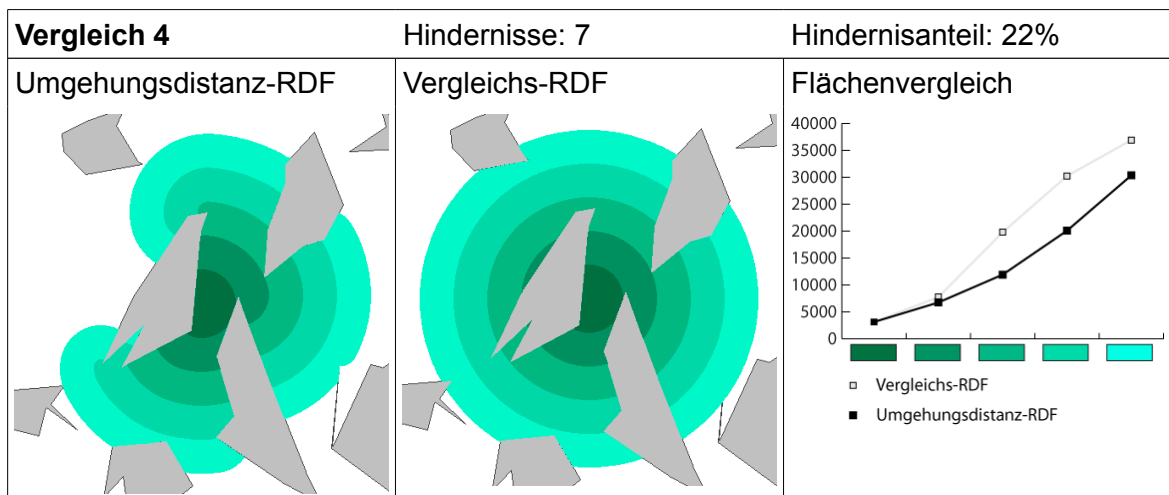


Abbildung 7.4: Vergleich 4 – Zufällige Hindernisse, geringer Flächenanteil

Vergleich 4 hat einen deutlich geringeren Hindernisanteil als die ersten drei Vergleiche. Die Fläche der einzelnen Ringe der beiden RDF-Varianten unterscheiden sich nicht sehr stark. Auf die Gesamtfläche gerechnet macht der Unterschied aber doch 26.2% aus. Verantwortlich für den grössten Teil des Unterschieds ist das Polygon zur Linken. Seine längste Achse ist wiederum tangential zu den Kreisen der RDF.

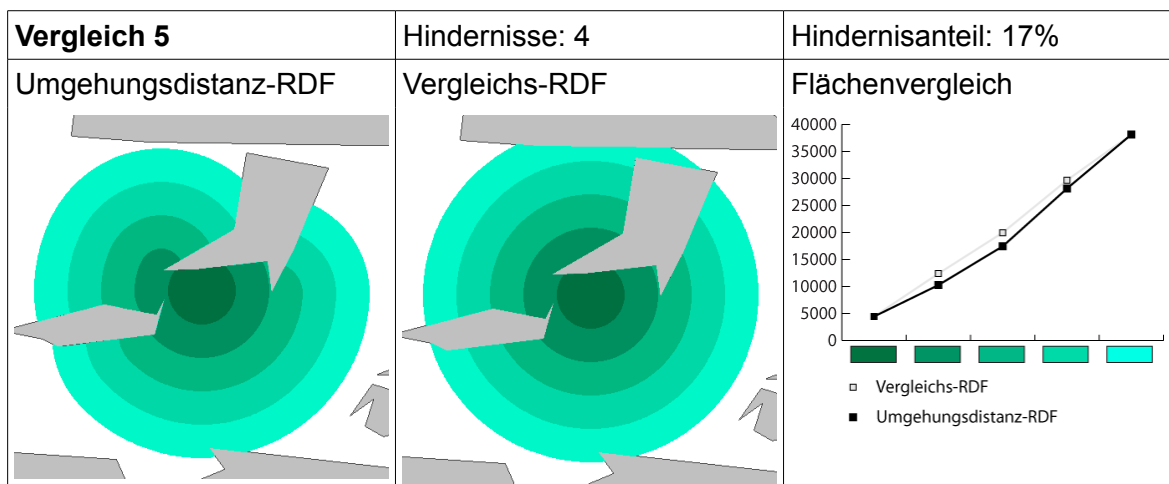


Abbildung 7.5: Vergleich 5 – Zufällige Hindernisse mit geringem Einfluss

In Vergleich 5 ist der Unterschied in der Fläche der beiden Varianten – bei nur vier Hindernissen – gering. Die Gesamtfläche der U-RDF ist lediglich 6.0% kleiner. Die Achse des Hindernisses auf der linken Seite zeigt ungefähr auf den Mittelpunkt der RDF. Ein solches Hindernis hat eine geringe Differenz zur Folge. Das zweite Hindernis nahe am Zentrum der RDF hat eine ähnliche Orientierung. Da es aber breiter ist, ist sein Einfluss grösser. Die Orientierung der Hindernisse im Raum spielt also eine wichtige Rolle und determiniert den Unterschied zwischen den zwei RDF-Varianten.

## 7.2 Spezifische Konfigurationen

Es wurden auch spezielle Fälle getestet. Die zufälligen Hindernisse haben alle eine bestimmte Grösse und Form. Wie gut eignet sich die U-RDF aber für andere Formen und Grössen? Um diese Frage zu beantworten werden Vergleiche mit linearen, besonders grossen und besonders kleinen Hindernissen betrachtet.

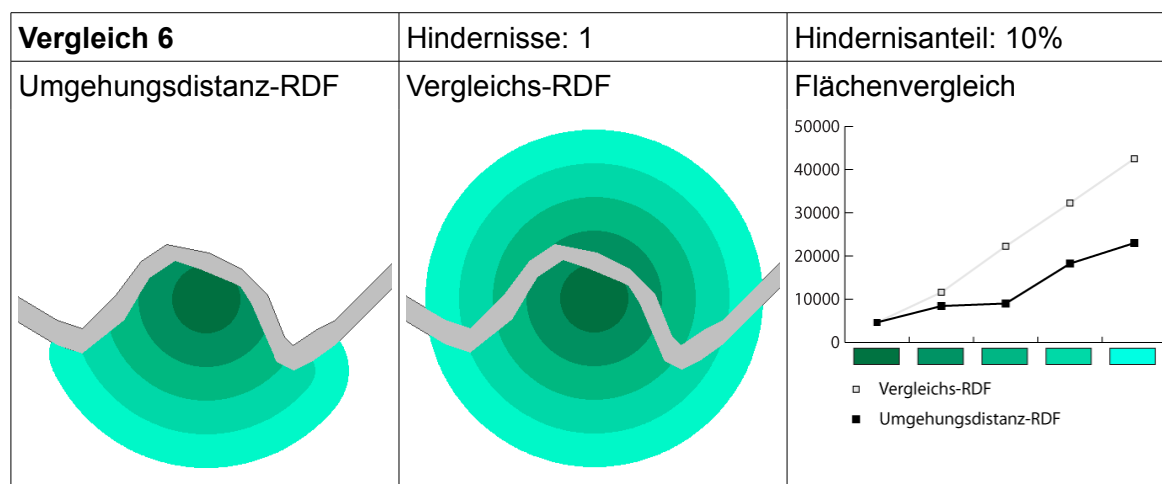


Abbildung 7.6: Vergleich 6 – Ein lineares Objekt als Hindernis

Lange, dünne Objekte kommen im Raum häufig vor. Flüsse sind für viele MO ein Hindernis. Auch Strassen können Barrieren sein. In Vergleich 6 ist das einzige Hindernis ein solches lineares Objekt. Der Vergleich 6 zeigt, dass in diesem Fall der Unterschied zwischen den zwei Formen der RDF beachtlich ist. Dies ist der Fall, weil die U-RDF zusammenhängend sein muss. Der Raum jenseits des Hindernisses ist vollkommen unerreichbar. Die Fläche der U-RDF ist knapp halb so gross wie diejenige der Vergleichs-RDF, beide Kurven des Flächenvergleichs steigen aber nahezu linear an. Ein linienähnliches Hindernis kann den lokalen Raum aufteilen. Deshalb hat es einen grossen Einfluss auf die Umgehungsdistanz zwischen Punkten auf unterschiedlichen Seiten des Hindernisses.

Der Teil der Fläche der Vergleichs-RDF, der auf der anderen Seite des Hindernisses ist, könnte weggelassen werden. Dann wäre der Unterschied in Form und Fläche geringer. Die Teile der beiden RDFs unterhalb des Hindernisses sind sich visuell sehr ähnlich.

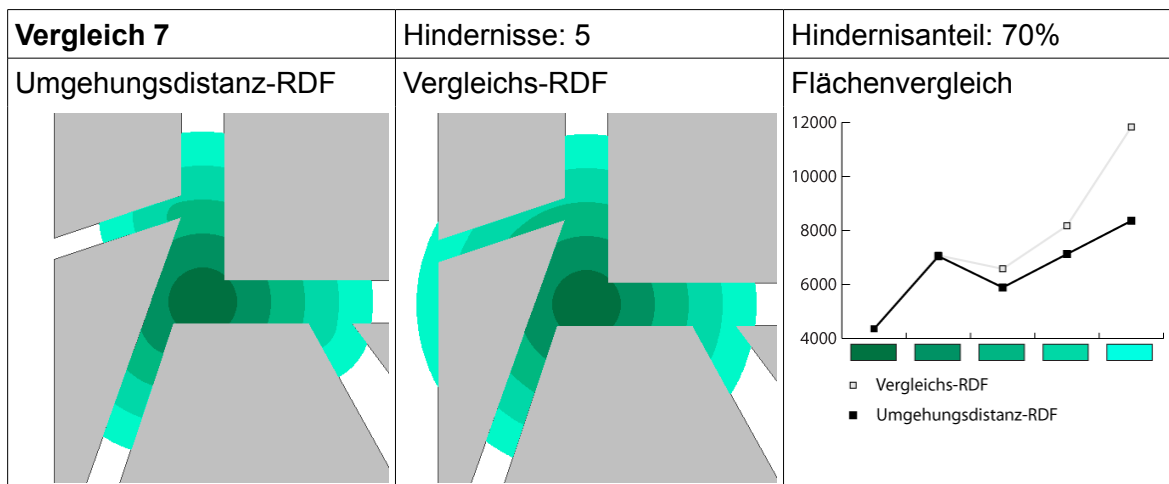


Abbildung 7.7: Vergleich 7 – In einem Netzwerk

Genauso wie lineare Objekt Hindernisse sein können, können sie auch die Bewegung erst ermöglichen. Das MO bewegt sich dann in einem Netzwerk. In Vergleich 7 bilden die Zwischenräume der Hindernisse ein Netzwerk. Deshalb ist auch der Anteil an Hindernissen mit 70% deutlich höher als in allen anderen Vergleichen. In den Teilen des Netzwerks, die gerade vom Mittelpunkt aus wegführen, sind die beiden RDF-Varianten identisch. Nur in Abzweigungen gibt es Unterschiede. So sind die Unterschiede in den ersten zwei Ringen vernachlässigbar. Unterschiede gibt es erst ab Ring 3, wo die erste Verzweigung auftritt, die nur teilweise vom Mittelpunkt aus sichtbar ist. Dass überhaupt Unterschiede auftreten, liegt an der Konfiguration der Hindernisse. Wenn alle Zwischenräume radial vom Mittelpunkt aus weggehen würden, gäbe es keine Differenz zwischen der U-RDF und der Vergleichs-RDF. Ob ein grosser Unterschied entstehen kann hängt also davon ab, wie verzweigt das Netzwerks ist. Die Zwischenräume sind in Vergleich 7 alle gerade. Gewundene Zwischenräume würden die Wahrscheinlichkeit für grosse Unterschiede erhöhen.

Die zufälligen Hindernis-Polygone und die Umgehungsdistanz-Buffer in den ersten fünf Vergleichen sind ähnlich gross. Die kleinsten Polygone sind etwa so gross wie der innerste Ring. Die grössten Polygone sind deutlich kleiner als alle Ringe zusammengezählt. Die längsten Segmente der Polygone sind ungefähr so lang wie die Maximaldistanz des äussersten Ringes. Die U-RDF ist in Form und Fläche meist deutlich von der Vergleichs-RDF verschieden. Wie sieht es aber aus, wenn die Polygone deutlich grösser oder deutlich kleiner sind als die Maximaldistanz der RDF?

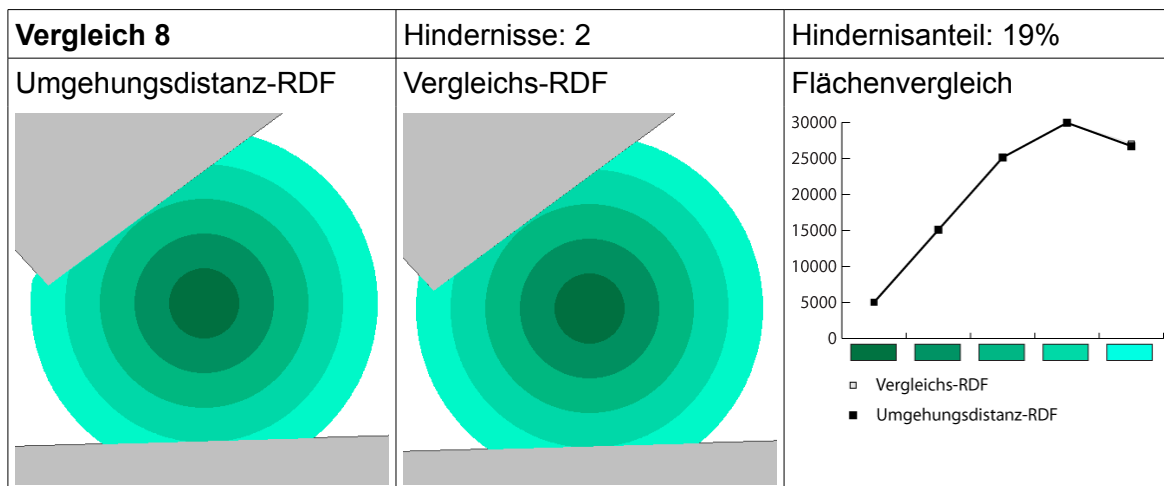


Abbildung 7.8: Vergleich 8 – Grosse Hindernisse

In Vergleich 8 wurden Polygone verwendet, deren Fläche ein Vielfaches derjenigen der RDF-Ringe ist. Auch die Randsegmente der Polygone sind deutlich länger als die Maximaldistanz der RDF. In Vergleich 8 beträgt der Unterschied zwischen den Flächen gerade mal 24 Pixel. Ein Unterschied kann nur zustande kommen, wenn ein Eckpunkt eines Polygons innerhalb des *buffer*s zu liegen kommt. Dabei muss eines der an den Eckpunkt grenzenden Segmente vom Mittelpunkt der RDF aus nicht sichtbar sein. In Vergleich 8 ist dies der Fall. Bei grossen Hindernis-Polygonen ist die Wahrscheinlichkeit gering, dass es einen erheblichen Unterschied zwischen den beiden RDFs gibt.

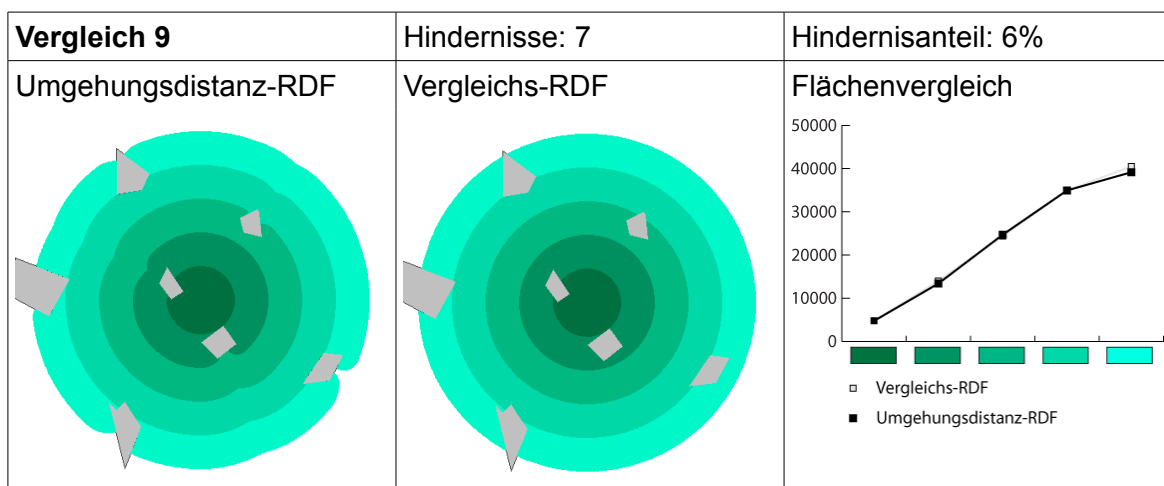


Abbildung 7.9: Vergleich 9 – Kleine Hindernisse

In Vergleich 9 wurden kleine Hindernis-Polygone verwendet. Die 7 Hindernisse haben einen Anteil von nur 6% an der maximal möglichen Gesamtfläche. Der Unterschied zwischen den Gesamtflächen der beiden Varianten der RDF macht gerade mal 1.08% aus. Nur beim zweitinnersten und beim äussersten Ring ist der Unterschied grösser als 2%. Der zweite Ring von aussen ist bei der U-RDF sogar über 1% grösser. Die kleinen Polygone

haben einen kleinen Einfluss, der Unterschied zwischen den beiden Varianten ist daher vernachlässigbar. Sowohl visuell als auch von der Fläche her ist die Differenz unbedeutend.



# 8 Diskussion

Im Modellbildungsprozess sind nun wieder die Systemtheorien im Fokus. Durch Auswertung der Resultate können Rückschlüsse auf die Theorien gezogen werden, die dem Modell zugrunde liegen. Die Systemtheorien sind die in Kapitel 3 vorgestellten Konzepte zum Kontext der Bewegung. Im Verlauf dieser Arbeit fanden zwei grosse Einschränkungen statt. Der Bewegungskontext wurde in drei Kontextklassen aufgeteilt, ein konzeptuelles Modell wurde aber nur für den Fortbewegungskontext erstellt. Spezifiziert und implementiert wurde danach nicht dieses Modell, sondern ein Algorithmus, der Teile dieses Modells verwendet.

Es ist klar, dass unter diesen Umständen nur sehr beschränkt von den Resultaten in Kapitel 7 auf die Konzepte der Systemtheorie in Kapitel 3 geschlossen werden kann. Dennoch kann durch die Diskussion des Algorithmus anhand dieser Konzepte deren Tauglichkeit aufgezeigt werden. Dies geschieht in einem ersten Teil dieses Kapitels. Danach wird das Modell aus Kapitel 4 diskutiert. Schliesslich wird auf den Algorithmus, seine Implementation und seine Resultate eingegangen.

## 8.1 Diskussion der Kontext-Begriffe

In Kapitel 3.1 wurde der Kontext-Begriff in Analyse- und Bewegungskontext aufgeteilt. Vom bewegungsökologischen Modell von Nathan et al. (2008) ausgehend, wurde der Bewegungskontext sodann in drei Komponenten aufgegliedert. Diese sind der Motivations-, der Navigations- und der Fortbewegungskontext. Zusätzlich wurde zwischen einem erkundenden und einem erklärenden Ansatz bei der Integration von Bewegungsdaten und ihrem Kontext unterschieden. Nur auf den Fortbewegungskontext wurde im Detail eingegangen. Bei einer Diskussion des Umgehungsdistanz-RDF Algorithmus werden dennoch alle diese Begriffe benötigt.

Das Modell für den Bewegungskontext erklärt Zusammenhänge zwischen der Bewegungskapazität und der physischen Umgebung des MO. Es ist daher eher dem erklärenden Ansatz zuzuordnen. Die RDF wurde entwickelt, um das Auffinden von Zusammenhängen zwischen dem MO und seiner Umgebung zu ermöglichen. Die RDF ist daher eine erkundende Methode. Der Umgehungsdistanz-RDF Algorithmus verknüpft somit das erklärende Modell mit einem erkundenden Verfahren. Der euklidische Distanzbegriff wird durch einen Distanzbegriff ersetzt, der auf die Bewegungskapazität des MO Rücksicht nimmt.

Bei der Diskussion des Algorithmus blieb das Endresultat der RDF stets im Hintergrund. Der Fokus lag auf dem Berechnen der für die RDF benötigten Geometrie. Das eigentliche Resultat der RDF ist aber nicht die Geometrie, es ist eine Verteilung eines Merkmales der Umgebung als eine Funktion der Distanz. Imfeld (2000) verwendete die RDF unter ande-

rem dazu, den Waldanteil der Umgebung von Trajektorien von Luchsen zu untersuchen. Die unterschiedliche Raumnutzung zweier Individuen über die Zeit (d.h. entlang ihrer Trajektorien) weist auf Differenzen im Motivationskontext hin. Da die beiden Individuen eine ähnliche Navigationskapazität und Bewegungskapazität haben, muss der Unterschied in ihrem räumlichen Verhalten auf den inneren Zustand zurückzuführen sein. Die Umgehungsdistanz-RDF kann benutzt werden, um solche und ähnliche Zusammenhänge zu finden. Mit der Umgehungsdistanz-RDF haben wir eine Methode, die verwendet werden kann, um Zusammenhänge zwischen Trajektorien und Bewegungskontext zu suchen. Muster zwischen Navigationskontext, Motivationskontext und der Bewegung können mit dieser Methode gefunden werden.

In Abschnitt 3.3 wurden die Klassifikationen von Andrienko und Andrienko (2007) und Dodge et al. (2008) auf die drei Bewegungskontext-Klassen abgebildet. Dabei wurde gezeigt, dass die Begriffe des Navigations-, Motivations- und Fortbewegungskontext helfen, Fragestellungen über den Bewegungskontext zu präzisieren. Bei Begriffen wie Wetter oder Müdigkeit ist eine Zuteilung zu einer der drei Arten des Bewegungskontexts schwierig. Dies ist der Fall, weil solche Begriffe unterschiedliche Phänomene bezeichnen können. Die Einteilung des Bewegungskontexts hilft uns in solchen Fällen, genauere Fragen zu stellen.

## 8.2 Weitere Integration der Umgehungsdistanz

Die Ersetzung des euklidischen Distanzbegriffs ist nur eine der in Abschnitt 4.5 diskutierten Anwendungsmöglichkeiten des Modells. Die Integration des Fortbewegungskontexts in die Umgehungsdistanz-RDF ist rudimentär. Um aufzuzeigen, wie vielseitig das Modell verwendet werden kann, sind hier weitere mögliche Anwendungen im Zusammenhang mit der RDF aufgelistet.

- Bewegungsarten haben für eine gegebene Zeiteinheit unterschiedliche Reichweiten. Die Maximaldistanz, und damit die Radien der Ringe der RDF, können mit wechselnder Bewegungsart variiert werden.
- Verschiedene Bewegungsarten haben unterschiedliche Hindernisse. Für ein MO mit mehreren Bewegungsarten können mehrere Datensätze mit den entsprechenden Hindernissen erstellt werden. Die aus der Fortbewegungsart resultierenden Hindernisse werden bei der Berechnung der Umgehungsdistanz-RDF verwendet.
- Anstelle der binären Unterscheidung Hindernis/Nicht-Hindernis könnte das Modell auch zur Erstellung von differenzierten Kostenoberflächen verwendet werden. Die Rasterlösung für die Umgehungsdistanz-RDF kann mit solchen

Kostenoberflächen umgehen. Eine Vektorlösung müsste erst noch erarbeitet werden.

### 8.3 Diskussion des Algorithmus

In diesem Kapitel wird auf den spezifizierten und implementierten Algorithmus eingegangen. Ein erster Abschnitt beschäftigt sich mit der Frage, ob die Implementierung den Spezifikationen genügt. Dabei wird auf Methoden der Verifikation eingegangen. Danach stellt sich die Frage, inwiefern die Gebrauchstauglichkeit des Algorithmus abgeschätzt werden kann. Es wird auf den Einfluss der Konfiguration der Hindernisse und den Aufwand, der mit dem Einsatz des Umgehungsdistanz-RDF Algorithmus verbunden ist, eingegangen. Dabei wird auch aufgezeigt, wann sich ein solcher Einsatz lohnt, und wann eher die normale RDF vorzuziehen ist. Abschliessend wird auf Probleme mit Geodaten eingegangen, die zu erwarten sind, wenn der Algorithmus eingesetzt wird.

#### 8.3.1 Verifikation der Implementierung

Verifikation heisst, zu zeigen, dass ein Modell formal korrekt ist. Es muss aufgezeigt werden, dass das implementierte Modell den Modellspezifikationen entspricht (Sargent 1988). Der implementierte Algorithmus muss also mit den Spezifikationen aus Abschnitt 5.2 übereinstimmen. Bei dynamischer Verifikation wird das Modell mit verschiedenen Startbedingungen durchgerechnet und die Resultate werden auf ihre Plausibilität geprüft (Sargent 1988: 35). Für ein geometrisches Problem eignet sich dynamisches Testen gut. Das Resultat kann visualisiert werden und Fehler (z.B. falsche oder fehlende Teilpolygone der Lösung) fallen in einer Visualisierung schnell auf. Visualisierungen wurden sowohl zum Testen von Programmteilen, wie auch zum Testen des ganzen Algorithmus verwendet.

Teilaufgaben, wie das Finden der `BorderNodes`, die Berechnung der Schattenpunkte und die Erstellung der `NodePoint`-Struktur wurden ausgiebig getestet. Die Teilaufgaben wurden mit zufälligen achteckigen Polygonen ausgeführt. Das Resultat, z.B. die gefundenen `BorderNodes`, wurden dann zusammen mit den Hindernissen visualisiert. Für einen mit dem Algorithmus vertrauten Betrachter ist es leicht, in einer Visualisierung allfällige Fehler zu erkennen. Nach zahlreichen Iterationen wurden die Klassen, die die Teilaufgaben lösen, als korrekt befunden. Abbildungen 6.3 und 6.4 sind solchen Test-Visualisierungen von Teillösungen entnommen.

Die `Runner`-Klasse, die den Algorithmus steuert, ist ein besonders komplexer Bestandteil des Algorithmus. Deshalb wurde diese Klasse in mehreren Schritten getestet. Zuerst wurde `Runner` ohne Rekursion betrachtet. Das Erstellen von Kreisbögen und Dreiecken für den Umgehungsdistanz-Buffer geschieht in zwei getrennten Methoden. Diese beiden Methoden konnten getrennt getestet werden. Sobald diese zwei Schritte im Algorithmus als kor-

rekt befunden wurden, konnte das gesamte `Runner`-Objekt ohne die Rekursion getestet werden (Abbildung 6.5). Erst als Fehler auf dieser Ebene ausgeschlossen werden konnten, wurde der Algorithmus rekursiv getestet. Das resultierende Polygon der `Runner`-Klasse mit Rekursion ist der Umgehungsdistanz-Buffer. Dieser wurde zusammen mit den Hindernissen, dem Mittelpunkt und einem Maximaldistanz-Kreis dargestellt. Durch zahlreiche Tests mit unterschiedlichen Hindernis-Konfigurationen konnte gezeigt werden, dass der Algorithmus ein korrektes Resultat liefert.

Die Implementierung wurde also durch dynamisches Testen verifiziert. Die Resultate entsprechen der Spezifikation, die der konzeptuelle Algorithmus vorgibt.

### 8.3.2 Einfluss der Geometrie der Hindernisse

Die Vergleiche in Kapitel 7 haben einige Hinweise darauf gegeben, bei welcher Form von Hindernissen der Unterschied zur normalen RDF am grössten ist. Dabei spielt die Maximaldistanz der RDF eine zentrale Rolle. Wenn in diesem Abschnitt von 'klein' oder 'gross' gesprochen wird, so ist das immer in Relation zur Maximaldistanz.

Kleine Hindernisse haben einen geringen geometrischen Unterschied in den zwei RDF-Varianten zur Folge. Dies entspricht auch der Intuition. Der zur Umgehung eines kleinen Hindernisses nötige Weg ist kurz. Und selbst zwischen einer grossen Anzahl von kleinen Hindernissen lässt sich ein ziemlich direkter Weg zu einem Ziel finden. Wenn also die Hindernisse geringe Fläche und einen kleinen Durchmesser haben, wird es sich kaum lohnen, die normale RDF durch die Umgehungsdistanz-RDF zu ersetzen.

Hindernisse, die deutlich grösser sind als die Maximaldistanz, haben ebenfalls einen geringen Unterschied zur Folge. Insbesondere wenn die Segmente der Polygonränder grösser sind als der Durchmesser der RDF, sind Unterschiede unwahrscheinlich. Damit ein Unterschied entstehen kann, muss ein Eckpunkt im Bereich der RDF liegen. Von den zwei angrenzenden Liniensegmenten des Randes muss eines vom Mittelpunkt her unsichtbar sein. Wenn die Segmente der Polygone wesentlich grösser sind als die Maximaldistanz, ist eine solche Konfiguration unwahrscheinlich.

Auch aus den Winkeln der Polygone konnten Erkenntnisse gewonnen werden. Spitze Winkel und konkave Partien eines Polygons führen eher zu merklichen Unterschieden. Sind die Winkel vorwiegend stumpf und die Polygone konvex, sind sich die zwei RDF-Varianten eher ähnlich.

Lange, dünne Hindernisse haben einen beachtlichen Einfluss, wenn sie tangential zur RDF stehen. Dies liegt daran, dass sie den Raum lokal aufteilen. Wenn die normale RDF von einem Hindernis zweigeteilt wird, könnte man auch nur den Teil betrachten, in dem der Mittelpunkt liegt. Der Unterschied ist dann eher gering. Bei linearen Hindernis-Objekten gilt, dass wenn sie gewunden sind, der Einfluss auf die Form der Umgehungsdistanz-RDF

grösser ist als bei geraden Objekten. Dasselbe gilt bei netzwerkartigen Konfigurationen, bei denen sich das MO in linienähnlichen Flächen zwischen Hindernissen bewegt. Bei diesen kommt es, neben der Sinuosität dieser Flächen, auch auf die Anzahl der Verzweigungen im Netzwerk an. Ist das Netzwerk stark verzweigt, und liegen mehrere Abzweigungen innerhalb der Maximaldistanz, ist der Unterschied zwischen den zwei Varianten der RDF gross. Natürlich kommt für einen solchen Fall auch die Modellierung der MO in einem Netzwerk in Frage. Es ist weniger aufwändig Distanzen in einem Netzwerk entlang der Kanten zu berechnen als die Umgehungsdistanz-RDF zu ermitteln.

Vor dem Einsatz der Umgehungsdistanz-RDF muss sich der Analytiker Gedanken darüber machen, ob sich bei den gegebenen Hinderniskonfigurationen signifikante Unterschiede zur normalen RDF ergeben. Dies ist vor allem dann der Fall, wenn die Hindernisse eine Grösse ähnlich derjenigen der Maximaldistanz der RDF haben, oder wenn die Hindernisse komplizierte Ränder mit konkaven und konvexen Teilstücken haben. Wenn dies nicht der Fall ist, kann auf die aufwändigere Umgehungsdistanz-RDF verzichtet werden, und die normale RDF kann zum Einsatz kommen.

### 8.3.3 Effizienz

Einen Umgehungsdistanz-Buffer für eine Hindernis-Konfiguration zu berechnen, wie sie in den ersten fünf Vergleichen verwendet wurde, braucht viel Zeit. Auf dem verwendeten System (Pentium Core Duo 2.4 Ghz, 2GB RAM, Windows Vista, Java 6.0) dauert eine solche Berechnung meistens einige Sekunden. Bei besonders komplizierten Konfigurationen musste mehr als eine Minute auf die Lösung gewartet werden. Da Rechenzeiten auf Systemen mit unterschiedlichen Ressourcen stark variieren, wurde auf eine systematische Erhebung verzichtet. Der offensichtlich hohe Rechenaufwand ist aber Grund genug, um sich Gedanken über die Komplexität des implementierten Algorithmus zu machen.

In der theoretischen Informatik wird die O-Notation verwendet, um die *computational complexity* eines Algorithmus anzugeben (Papadimitriou 1995). Der Buchstabe  $n$  bezeichnet dabei die Grösse in der Funktion, die die Anzahl Rechenschritte bestimmt. Diese Grösse wird auch Problemgrösse genannt. Eine Funktion der Ordnung  $O(n)$  kann mit linearem Aufwand gelöst werden. Wenn  $n$  erhöht wird, nimmt der Aufwand in linearem Ausmass zu. Eine Funktion der Ordnung  $O(1)$  hat einen konstanten Aufwand,  $n$  hat keinen Einfluss auf die Anzahl Rechenschritte. Eine  $O(n^2)$  Funktion braucht einen quadratischen Aufwand zur Lösung, wenn  $n$  grösser wird, nimmt der Aufwand mit dem Quadrat zu. Weitere häufige Ordnungen sind logarithmisch  $O(\log n)$ ,  $n$ -log- $n$   $O(n \log n)$ , polynomial  $O(n^k)$ ,  $k > 1$  und exponentiell  $O(d^n)$ ,  $d > 1$ . Um herauszufinden wie gross der Aufwand und welches die Ordnung des Algorithmus ist, werden die einzelnen Komponenten des Algorithmus be-

trachtet. Diejenige Komponente mit dem grössten Aufwand bestimmt den Gesamtaufwand. So ist z.B.  $O(n + 3n + n^2) = O(n^2)$ .

Im Algorithmus kommt ein `Quadtrees` zum Einsatz. Dieser beschleunigt die Auswahl der relevanten Polygone. Der Aufwand für diese Auswahl ist dank dem `Quadtrees` kleiner als linear. Aus diesen Polygonen wird die Datenstruktur der `NodePoints` aufgebaut. Das Auswählen der relevanten Eckpunkte aus dieser Datenstruktur hat einen linearen Aufwand. Für jeden der  $n$  Punkte muss ein Abstand zum Mittelpunkt  $M$  berechnet werden. Dann wird überprüft, ob dieser Abstand zu gross ist, wenn nicht, wird der Punkt ausgewählt.

Die Problemgrösse im Kernalgorithmus ist die Anzahl der selektierten Eckpunkte, die wir als  $n$  bezeichnen. Wenn wir  $n$  Eckpunkte haben, müssen wir auch mindestens  $n$  Segmente haben, da jeder Eckpunkt zwischen zwei Segmenten liegt. Das Erstellen der `BorderNodes` ist von linearer Ordnung. Jedes der  $n$  Segmente der Polygone muss auf Überschneidung mit einem Kreis getestet werden. Das Berechnen der Sichtbarkeit für alle  $n$  `NodePoints` hat einen quadratischen Aufwand. Für jeden der  $n$  Punkte wird eine Linie erstellt, die auf Überschneidung mit  $n$  Segmenten geprüft wird. Dasselbe gilt für die Berechnung der Schattenpunkte. Auch hier wird für jeden Kandidaten eine Linie mit allen Segmenten getestet. Das Konstruieren eines Teilresultates durch ein `Runner`-Objekt hat lediglich einen linearen Aufwand. Zwar müssen zahlreiche Vergleiche gemacht werden, diese werden aber nicht zwischen allen  $n$  `NodePoints` gemacht. Die Vergleiche finden in einem Fall zwischen einem Eckpunkt und seinen Nachbarn statt. Hier ist der Aufwand  $O(2n) = O(n)$ . Im anderen Fall werden die Punkte sortiert und dann werden aufeinander folgende `NodePoints` verglichen. Auch hier ist der Aufwand linear.

Aus den Komponenten des Algorithmus, ohne Rekursion, geht also ein quadratischer Aufwand hervor. Das Berechnen der Sichtbarkeit aller Punkte von einem einzelnen Punkt aus hat die Ordnung  $O(n^2)$ . Da der Algorithmus rekursiv ist, muss die Sichtbarkeit für mehrere Mittelpunkte berechnet werden. Die Anzahl der Punkte, an denen eine Rekursion nötig ist, hängt wiederum von  $n$  ab. Je mehr Eckpunkte, desto mehr Punkte, die einen Schatten werfen können. Der implementierten Algorithmus hat die Ordnung  $O(n * n^2) = O(n^3)$ .

Ein Algorithmus der **Ordnung  $O(n^3)$**  ist bei grossem  $n$  sehr rechenintensiv. Grosse Datensätze haben riesige Mengen an Positionsdaten. Wenn für  $m$  *fixes* für jeden einzelnen *fix* eine Umgehungsdistanz-RDF berechnet werden muss, ist die Komplexität  $O(m * n^3)$ .

Noch nicht betrachtet wurde die Komplexität des Zusammenfügens der Dreiecke und Kreissegmente der Lösung. Dafür wird eine Funktion der JTS verwendet. Die JTS hat drei verschiedene Methoden um Polygone zu vereinen. Dies sind die `union`-Funktion, die in der `Polygon`-Klasse definiert ist, der *cascade union* Algorithmus und die `buffer(0)`-Lösung (Davis 2007). Um viele Polygone zusammenzufügen ist iteratives verwenden von

`union` ineffizient. Verwendet wurde die `buffer(0)`-Lösung. Dabei wird um eine Sammlung von Polygonen ein `buffer` mit einem Abstand von Null erstellt. Das Resultat ist dasselbe wie bei iterativem Verwenden der `union`-Funktion, die Polygone werden vereinigt. Auch die `buffer(0)`-Lösung ist sehr rechenintensiv, ist aber deutlich schneller als iteratives `union`. Das wesentlich effizientere `cascade union` konnte nicht verwendet werden, da es strengere Anforderungen an die Geometrie der Input-Polygone stellt.

Die Komplexität der Grössenordnung  $O(n^3)$  gilt für den implementierten Algorithmus. Der vorgestellte konzeptionelle Algorithmus muss nicht zwingend diese Komplexität haben. In manchen Bereichen der Implementierung wurde eine einfache Lösung der effizienteren Lösung vorgezogen. So kann z.B. die Sichtbarkeit, bei Verwendung spezieller Datenstrukturen, mit weniger als quadratischem Aufwand berechnet werden (De Berg et al. 2000). Wenn die aneinander grenzenden Teilstücke der Lösung nacheinander berechnet werden, kann auf die Verwendung einer `union`-Funktion verzichtet werden. Stücke des Randes des Umgehungsdistanz-Buffers können dann aneinander gehängt werden. Alle Stücke zusammen ergeben so den Rand dieses Polygons.

Zuversichtlich stimmt auch, dass es für das verwandte *shortest path* Problem Algorithmen mit geringerer *computational complexity* gibt. Der von De Berg et al. (2000) diskutierte Algorithmus hat die Ordnung  $O(n^2 \log n)$ . Der Algorithmus von Hershberger und Suri (1993) war die erste Lösung des *shortest path* Problems mit weniger als quadratischem Aufwand. Dieser Algorithmus verwendet eine *shortest path map* und hat eine Komplexität von  $O(n \log n)$ .

Zusätzlich ist anzumerken, dass  $O(n^3)$  eine obere Grenze für den implementierten Algorithmus ist. Es ist in der Praxis sehr unwahrscheinlich, dass alle Eckpunkte einen Schatten werfen. Die Anzahl der Rekursionsschritte mag zwar von  $n$  abhängen, wird aber im Normalfall nur ein Bruchteil von  $n$  sein. In der Hindernis-Konfiguration in Abbildung 6.10 gibt es für den 200-Pixel `buffer` 13 Punkte, an denen eine Rekursion nötig ist. `NodePoints`, inklusive `BorderNodes` hat es 39. Es ist in diesem Beispiel also nur für jeden dritten Punkt eine Rekursion nötig. Weil der Radius bei der Rekursion kleiner wird, wird auch die Problemgrösse mit fortschreitender Rekursionstiefe kleiner. Während beim ersten `Runner`-Objekt noch viele Punkte auf Sichtbarkeit getestet werden müssen, sind bei einem späteren `Runner`-Objekt vielleicht nur noch ein oder zwei Nachbarpunkte nahe genug. Es darf auch angenommen werden, dass es Stellen im Raum gibt, in deren Nähe keine Hindernisse vorhanden sind. Solch ein Fall kann dank dem `Quadtree` mit weniger als linearem Aufwand festgestellt werden. Die Konstruktion des `buffers` ist dann von konstanter Komplexität, es muss lediglich ein Kreis erstellt werden. In diesem Fall ist der Aufwand nicht grösser als derjenige der Vergleichs-RDF. Anstelle von  $O(n^3)$  wird der Aufwand für manche Konfigurationen also quadratisch oder gar linear ausfallen.

Die Problemgrösse  $n$  hängt primär von zwei Faktoren ab. Dies sind die Maximaldistanz und die Verteilung der Eckpunkte im Raum. Bei einer hohen Dichte der Eckpunkte und einem grossen Radius wird  $n$  auch gross sein. Wenn aber nur wenige Eckpunkte innerhalb des Maximaldistanz-Radius zu liegen kommen, ist  $n$  klein und der Algorithmus schnell. Wie gross die Maximaldistanz sein soll, ist eine Frage des Analysekontexts. Um die Anzahl der Eckpunkte zu reduzieren, können die Hindernis-Polygone mit einem Linienvereinfachungs-Operator () vorgängig zur RDF stark generalisiert werden.

### 8.3.4 Die Rasterlösung

Die verwendete *Cost Distance*-Operation berechnet Distanzen nicht euklidisch. Rasterzellen werden entweder waagrecht, senkrecht oder um  $45^\circ$  geneigt traversiert. Dies hat eine achteckige Form des resultierenden *buffers* zur Folge. Ohne Hindernisse haben nur die acht Ecken eines regelmässigen Achtecks die Maximaldistanz  $d$  zum Mittelpunkt  $M$ . Die Formel für die Fläche des Achtecks ist:

$$A = \frac{8 * r^2 * \sin(45^\circ)}{2}$$

Die Formel für die Fläche eines Kreises ist:

$$A = r^2 * \pi$$

In beiden Formeln ist die Variable  $r^2$  enthalten. Der Unterschied kommt also allein durch die Konstanten zustande. Der konstante Teil der Formel beim Achteck ist 2.828, beim Kreis ist er  $\pi$ . Im hindernisfreien Fall ist das Achteck 9.97% kleiner als ein Kreis. Wenn Hindernisse vorhanden sind variiert dieser Unterschied, 10% ist aber ein guter Richtwert. Der Unterschied in der Fläche der Umgehungsdistanz-Buffer gegenüber der analytisch korrekten Vektorlösung ist gross, zu gross um Vektor- und Rasterfall sinnvoll vergleichen zu können.

Eine brauchbare Rasterlösung müsste die Euklidische Distanz zwischen Zellmittelpunkten verwenden. Eine solche Lösung wird hier kurz skizziert. Raster-GIS haben Methoden für die Berechnung der Sichtbarkeit in einem Geländemodell. Wenn die Hindernisse und der freier Raum entsprechend modelliert werden, können solche Methoden für das Umgehungsdistanz-Problem verwendet werden. Es könnte dann den sichtbaren Zellen die euklidische Distanz zugeordnet werden. Rekursionspunkte wären Zellen, die eine sichtbare, eine nicht sichtbare und eine Hindernis-Nachbarzelle haben. In die vom Rekursionspunkt sichtbaren Zellen könnte die Distanz zum Rekursionspunkt, addiert zur bereits zurückgelegten Distanz, geschrieben werden. Die Ergebnisraster der einzelnen Rekursionsschritte könnten dann zusammengefügt werden, indem immer der kleinste Distanz-Wert übernommen wird. Eine solche Lösung wäre mit grösserem Aufwand verbunden als die getestete Rasterlösung, wäre aber der Vektorlösung sehr viel ähnlicher.

### 8.3.5 Analyseaufwand

Nicht nur der Rechenaufwand, auch der Aufwand bei der Analyse ist hoch. Die Erstellung der RDF und die Visualisierung einer Verteilung innerhalb der Distanzklassen lässt sich automatisieren. Das visuelle Auffinden von Mustern ist aber vorderhand weitgehend dem Anwender überlassen. Die Visualisierungen der RDF werden erstellt, um die natürliche Gabe des Betrachters, Muster zu erkennen, zu unterstützen. In der Praxis interessiert sich der Anwender eher für einen Verlauf mit der Zeit als für einzelne RDF-plots. Der *temporal RDF plot* von Imfeld (2000) ist ein geeignetes Mittel, um den Verlauf einer RDF-Funktion während der Bewegung eines MOs zu visualisieren. So können die einzelnen RDF-Diagramme einer Trajektorie zusammengefasst als Funktion der Zeit betrachtet werden. Methoden um mehrere *temporal RDF plots* effizient zu visualisieren wurden noch nicht vorgeschlagen. Wenn viele MO auf diese Art analysiert werden, muss der Betrachter eine grosse Anzahl an Diagrammen betrachten. Zusammenhänge unter solchen Umständen zu finden, stellt hohe Anforderungen an die kognitiven Fähigkeiten des Analytikers. Methoden die erlauben, eine Vorauswahl zu treffen, wären wünschenswert. Methoden, die uns helfen, interessante Trajektorien, oder Teile davon, zu identifizieren, könnten den kognitiven Aufwand und den Rechenaufwand verringern.

### 8.3.6 Probleme von Bewegungsdaten

Die Diskussion von Problemen, die sich bei der Verwendung von Bewegungsdaten ergeben, drängt sich an dieser Stelle auf. Für die Modellierung, die Spezifizierung und die Entwicklung des Algorithmus spielen diese Probleme keine grosse Rolle. Wenn der Algorithmus angewandt werden sollte, müssen diese Schwierigkeiten beachtet werden. Laube et al. (2005) diskutieren solche Probleme im Rahmen ihrer REMO Analysemethode für Bewegungsmuster. Hier wird abgeschätzt, welche Folgen die von Laube et al. (2005) diskutierten Probleme für die Umgehungsdistanz-RDF haben könnten.

- *Uncertain and missing fixes*. Fehler in den Bewegungsdaten können für die Umgehungsdistanz-RDF fatal sein. Nur schon ein Messfehler von wenigen Metern kann bedeuten, dass ein *fix* auf der anderen Seite eines linearen Hindernisses ist. Lineare Hindernisse haben die Tendenz, den lokalen Raum aufzuteilen. Die aus dem fehlerhaften *fix* resultierenden Umgehungsdistanz-Buffer und die korrekten Buffer sind disjunkt, kein Punkt im Raum ist in beiden Buffern. Der betrachtete Raum ist also wegen einem kleinen Fehler komplett verschieden. Fehlende *fixes* sind weniger problematisch. Ihr Fehlen ist dann relevant, wenn deswegen ein Muster nicht erkennbar ist.

- *Interpolation issues.* Das REMO *framework* ist auf regelmässige Zeitabstände zwischen *fixes* angewiesen. Für die RDF ist dies nicht der Fall. Für einen *temporal RDF plot* ist eine Regelmässigkeit in den verwendeten *fixes* aber wünschenswert. Die Interpolation in Bewegungsdaten ist ein Forschungsfeld mit vielen offenen Fragen (Laube et al. 2005).
- *Fix sampling rate vs. analysis granularity.* Die Granularität der Analyse sollte derjenigen der Daten entsprechen. Probleme mit der räumlichen und zeitlichen Auflösung der Daten wurden bereits in Abschnitt 4.4 als eine grosse Herausforderung identifiziert.
- *Aggregation.* Temporale Aggregation der Bewegungsdaten kann nötig sein, um regelmässige RDF Funktionen für einen *temporal RDF plot* zu bekommen. Auch für die Vergleichbarkeit mehrerer *plots* untereinander ist eine Regelmässigkeit wünschenswert. Temporale Aggregationen können zu Problemen ähnlich des *modifiable area unit problem* (MAUP, Openshaw 1984) führen. Deshalb müssen Effekte solcher Aggregationen im Detail untersucht werden (Laube et al. 2005).
- *Classification.* Klassifikationsprobleme treten bei REMO im Zusammenhang mit der Einteilung beschreibender Variablen der Bewegung in Klassen auf. Bei der RDF wird die Distanz zum *fix* klassifiziert. Das Verwenden von zu vielen oder zu wenigen Klassen kann das Auffinden von Mustern erschweren oder verunmöglichen.

# 9 Fazit

Die Ziele dieser Arbeit waren (1) einen Überblick über die Forschung zum Thema Kontext der Bewegung zu verschaffen, (2) Konzepte für ein besseres Verständnis des Kontext-Begriffs zu erarbeiten, (3) ein Modell zu entwerfen für die Integration von Kontext in die Analyse von Bewegungsdaten und (4) auf dem Modell aufbauend eine Anwendung für die Analyse des Kontexts der Bewegung zu implementieren. In diesem Kapitel wird diskutiert, inwiefern diese Ziele erreicht wurden. Zuerst werden die Ergebnisse dieser Arbeit zusammengefasst. Dann werden die daraus gewonnenen Erkenntnisse aufgelistet. Abschließend wird ein Ausblick gewagt.

Hierbei gelten gewisse Einschränkungen. Die Umgehungsdistanz-RDF wurde nur vorgestellt und nicht eingesetzt. Probleme, die bei der Verwendung von echten Bewegungsdaten und bei der Modellierung des Bewegungskontexts auftreten, konnten deshalb nur hypothetisch behandelt werden. Die Nützlichkeit der Umgehungsdistanz-RDF konnte nicht an einem konkreten Anwendungsbeispiel demonstriert werden. Auch das Modell für den Fortbewegungskontext wurde nur sehr eingeschränkt angewandt. Es dient als Grundlage für die Berechnung des Umgehungsdistanz-Buffers. Da der Algorithmus aber mit zufälligen Polygonen getestet wurde, kam das Modell nicht zum Einsatz. Deshalb bleibt die Frage offen, wie das Modell im Zusammenhang mit einem konkreten MO mit Inhalten gefüllt wird.

## 9.1 Ergebnisse

- Ein Überblick über das Forschungsfeld der Analyse von MO wurde gegeben. Dabei wurden Arbeiten besonders ausführlich diskutiert, die sich mit dem Umfeld in dem Bewegung stattfindet auseinandersetzen.
- Die Position der Geographie im interdisziplinären MO-Forschungsfeld wurde diskutiert. Die Geographie hat ihre Stärke im Erkennen von räumlichen Zusammenhängen. Sie ist aber auch eine Schnittstelle zwischen anwendungsorientierter und methodisch orientierter MO-Forschung. Die Geographie kann Methoden entwickeln, mit denen sich Relationen zwischen MO und ihrer räumlichen Umgebung finden lassen.
- Der Kontext-Begriff wurde für die MO-Forschung in mehrere Kontextklassen aufgeteilt. Es wurde zwischen Analysekontext und Bewegungskontext unterschieden. Der Bewegungskontext wurde in Motivations-, Navigations- und Fortbewegungskontext unterteilt. Im Verlauf der Arbeit konnte mehrfach gezeigt werden, dass diese Begriffe nützlich sind. Insbesondere zwingen die-

se Begriffe den Forscher dazu, Fragestellungen betreffend des Kontexts präzise zu formulieren.

- Der Fortbewegungskontext wurde modelliert. Dieses Modell zeigt die Relationen zwischen dem MO, seinen Fortbewegungsarten und der Oberfläche auf. Eigenschaften der Bewegung, wie z.B. die Maximalgeschwindigkeit konnten als ein Resultat der mechanischen Interaktion von MO und der Oberfläche, auf der die Bewegung stattfindet, identifiziert werden. Das Modell zeigt auch die Verbindungen zwischen dem modellierten Fortbewegungskontext und Trajektoriendaten auf.
- Es konnten mehrere Anwendungsmöglichkeiten für das Modell des Fortbewegungskontexts identifiziert werden. Eine davon ist das Ersetzen des euklidischen Distanzbegriffs durch die Umgehungsdistanz, die Hindernisse für die Bewegung berücksichtigt.
- Die Umgehungsdistanz wurde definiert als kürzester Pfad zwischen zwei Punkten, der nicht durch das Innere von Hindernissen geht. Die Umgehungsdistanz ist näher an der Perspektive des MOs als die euklidische Distanz, da sie Rücksicht auf den Fortbewegungskontext des Individuums nimmt. Ein Algorithmus zur Berechnung eines Umgehungsdistanz-Buffers wurde vorgestellt. Dieser kann ein Polygon berechnen, dessen Fläche innerhalb einer bestimmten Umgehungsdistanz zu einem Ausgangspunkt liegt.
- Der Algorithmus wurde in Java unter Verwendung der JTS-Programm-bibliothek implementiert. Die Implementierung wurde durch dynamisches Testen verifiziert. Für die Verifizierung wurden zahlreiche Visualisierungen von Teilproblemen und der Gesamtlösung verwendet.
- Der Umgehungsdistanz-Buffer Algorithmus wurde verwendet, um die Distanzklassen für eine eigene Version der RDF (im Gegensatz zu Imfeld 2000) zu erstellen. Dabei wurden aus dem Fortbewegungskontext-Modell hergeleitete Hindernisse in die Berechnung der Distanz mit einbezogen.
- Unterschiede zwischen den beiden RDF-Varianten wurden untersucht und es wurde evaluiert, wann welche der beiden Varianten einzusetzen ist. Folgende Faktoren haben einen grossen Unterschied zwischen den beiden RDF-Varianten zur Folge:
  - (1) Linienähnliche Hindernisse, die den Raum lokal aufteilen.
  - (2) Spitze Winkel und konkave Partien der Hindernis-Polygone.

- (3) Grösse der Hindernisse ähnlich derjenigen des Buffers.
- Die Komplexität des Umgehungsdistanz-Algorithmus wurde abgeschätzt. Mit  $O(n^3)$  ist der rechnerische Aufwand des Algorithmus hoch. Es konnte aber gezeigt werden, dass  $O(n^3)$  eine obere Grenze ist. In vielen Fällen in der Praxis ist der Aufwand geringer. Wenn die Dichte der Eckpunkte nicht sehr hoch ist, kann der Algorithmus problemlos eingesetzt werden.

## 9.2 Erkenntnisse

Ziel (1) war, einen Überblick über die Forschung zum Thema Kontext der Bewegung zu verschaffen. Dabei wurde offensichtlich, dass wenig Theorie über den Kontext, in dem die Bewegung stattfindet, vorhanden ist. Dieser Mangel ist aber bereits erkannt und einige Forscher haben begonnen sich mit dem Thema Kontext auseinanderzusetzen. Es kann zwischen anwendungsorientierter Forschung und methodisch orientierter Forschung unterschieden werden. Anwendungsorientierte Forschung (z.B. die Ökologie oder die Verkehrsplanung) setzt sich mit konkreten MO auseinander und Wissen über den Kontext der Bewegung dieses MO ist vorhanden. Allgemeine Theorien über den Kontext sind aber wenige vorhanden. Auf der anderen Seite entwickelt die methodisch orientierte Forschung Theorien, Modelle und Werkzeuge zur Analyse von Bewegungsdaten. Der Bewegungskontext spielt im Normalfall keine Rolle.

Ziel (2) war, Konzepte für ein besseres Verständnis des Kontext-Begriffs zu erarbeiten. In der Literatur, die sich mit dem Kontext der Bewegung auseinandersetzt, werden verschiedene Begriffe verwendet. Der Begriff *influencing factors* (Andrienko und Andrienko 2007) zum Beispiel ist aber auch nicht wesentlich klarer als der Begriff Kontext. Die Unterscheidung zwischen dem Analysekontext und dem Bewegungskontext beseitigt eine erste Quelle potentieller Missverständnisse. Dank dem bewegungsökologischen Modell von Nathan et al. (2008) gelang es, den Bewegungskontext weiter aufzuteilen. Die Begriffe Navigations-, Motivations- und Fortbewegungskontext haben sich in dieser Arbeit bewährt. Sie helfen Aussagen und Fragestellungen zu präzisieren und sind daher einem allgemeinen Kontext-Begriff überlegen.

Die Modellierung des Fortbewegungskontexts hat zur Erkenntnis geführt, dass dieser einfacher zu handhaben ist, als der Navigations- oder Motivationskontext. Der Fortbewegungskontext ist räumlich lokal und zeitlich auf einen Augenblick beschränkt. Zusammenhänge zwischen Bewegungsarten und Bewegungsdaten sind offensichtlich. Der Fortbewegungskontext eignet sich daher hervorragend als Startpunkt für die Integration von Kontext in die Analyse von Trajektorien.

Aus der Perspektive eines MOs ist die Entfernung zu einem Objekt so gross, wie der kürzeste Weg ist, auf dem das MO zu diesem Objekt gelangen kann. Da MOs von Hindernis-

sen in ihrer Bewegung eingeschränkt werden, ist dieser kürzeste Weg nicht äquivalent zur euklidischen Distanz des MOs zum Objekt. Deshalb ist es wichtig, dass die Umgehungsdistanz bei der Analyse von Bewegungsdaten verwendet wird, da sie in vielen Fällen aussagekräftiger ist als die euklidische Distanz. Da die Verwendung der Umgehungsdistanz mit einem grossen Aufwand verbunden ist, muss sich der Analytiker überlegen ob sich dieser zusätzliche Aufwand lohnt. In Abschnitt 8.3.2 wurde gezeigt, dass es geometrische Konfigurationen gibt, wo dies der Fall sein wird. Wenn Hindernisse den Raum lokal aufteilen berücksichtigt dies die U-RDF.

### 9.3 Ausblick

Es sollte ein Anwendungsgebiet gesucht werden, bei dem die Umgehungsdistanz-RDF mit grossem Nutzen eingesetzt werden kann. Wenn der Algorithmus in so einem Gebiet zum Einsatz kommt, könnten einige der offenen Fragen dieser Arbeit beantwortet werden.

Bewegungsdaten werden heute in grossen Mengen mit GPS und anderen Ortungstechnologien gesammelt. Die Entwicklung der Technologie ist mindestens so rasant wie diejenige des MO-Forschungsgebietes. Nicht nur Ortungstechnologien, auch andere Sensortechnologien werden laufend verbessert. In Zukunft wird es möglich sein Informationen über den Kontext zusammen mit den Bewegungsdaten zu erheben. Die Auswertung solcher Daten erfordert neue Methoden. Die theoretischen Grundlagen über den Bewegungskontext und Strategien zu dessen Erforschung bilden die Grundlagen, auf denen neue Methoden entstehen können. Die Zeit ist reif dafür, dass sich das MO-Forschungsfeld und damit auch die GIScience intensiv mit solchen Grundlagen auseinandersetzt.

# Literatur

- Alvares, L. O., Bogorny, V., Kuijpers, B., de Macedo J. A. F., Moelans, B., Palma, A. T., 2007. Towards semantic trajectory knowledge discovery. Technical report, Hasselt University.
- Alvares, L. O., Bogorny, V., Kuijpers, B., de Macedo J. A. F., Moelans, B., Vaisman, A., 2007. A model for enriching trajectories with semantic geographical information. *In: ACM GIS*. New York, NY: ACM Press, 162-169.
- Andrienko, G., Andrienko, N., Kopanakis, I., Ligtenberg, A., Wrobel, S., 2008a. Visual analytics methods for movement data. *In: F. Giannotti und D. Pedreschi, eds. Mobility, data mining and privacy*. Heidelberg, Berlin: Springer-Verlag, 375-410.
- Andrienko, G., Andrienko, N., Wrobel, S., 2007. Visual Analytics Tools for Analysis of movement data. *SIGKDD Explorations*, 9 (2), 38-46.
- Andrienko, N., Andrienko, G., 2007. Designing visual analytics methods for massive collections of movement data. *Cartographia*, 42 (2), 117-138.
- Andrienko, N., Andrienko, G., Pelekis, N., Spaccapietra, S., 2008b. Basic concepts of movement data. *In: F. Giannotti und D. Pedreschi, eds. Mobility, data mining and privacy*. Heidelberg, Berlin: Springer-Verlag, 15-38.
- Benkert, M., Gudmundsson, J., Hübner, F., Wolle, T., 2008. Reporting flock patterns. *Computational Geometry*, 41 (3), 111-125.
- Berg, de, M., Van Kreveld, M., Overmars, M., Schwarzkopf, O., 2000. *Computational geometry – algorithms and applications*. Second, revised edition. Heidelberg, Berlin: Springer-Verlag.
- Blok, C., 2000. Monitoring change: Characteristics of dynamic geo-spatial phenomena for visual exploration. *Lecture Notes in Computer Science*, 1849, 16-30.
- Bogorny, V., Wachowicz, M., 2008. A framework for context-aware trajectory data mining. *In: L. Cao, P. S. Yu, C. Zhang, H. Zhang, eds. Data mining for business applications*. Heidelberg, Berlin: Springer-Verlag, 225-239.
- Brinkhoff, T., 2002. A Framework for generating network-based moving objects. *Geoinformatica*, 6 (2), 153-180.
- Buchin, K., Buchin, M., Gudmundsson, J., 2008. Detecting single file movement. *Proceedings of the 16<sup>th</sup> ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*.
- Chen, G., Kotz, D. A., 2000. *Survey of context-aware mobile computing research*. Technical Report, Dept. of Computer Science, Dartmouth College.
- Chen, P., P.-S., 1976. The entity-relationship model – Toward a unified view of data. *ACM Transactions on Database Systems*, 1 (1), 9-36.
- Clancey, W. J., 1991. Review of *The invention of memory* by Israel Rosenfield. *Artificial Intelligence*, 50, 241-284.
- Davis, M., 2007. *Fast polygon merging in JTS using cascade union*. Erhältlich unter: <http://lin-earth-inking.blogspot.com/2007/11/fast-polygon-merging-in-jts-using.html> [Zugriff: 13.08.09]
- Dawkins R., 1987. *The blind watchmaker*. New York, London: W. W. Norton & Company Ltd.
- Dijkstra, E. W., 1959: A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269-271.

- Dodge, S., Weibel, R., Lautenschütz, A.-K., 2008. Towards a taxonomy of movement patterns. *Information Visualization*, 7, 240-252.
- Duden, 1996. *Duden – Rechtschreibung der deutschen Sprache, 21. Auflage*. Mannheim, Leipzig, Wien, Zürich: Dudenverlag.
- Dykes, J. A., Mountain D. M., 2003. Seeking structure in records of spatio-temporal behaviour: Visualization Issues, Efforts and Applications. *Computational Statistics & Data Analysis*, 43, 581-603.
- Egenhofer M. J., Herring J. R., 1991. *Categorizing binary topological relationships between regions, lines and points in geographic databases*. Technical Report, Department of Surveying Engineering, University of Maine.
- Feyerabend, P., 1983: *Wider den Methodenzwang*. Frankfurt am Main: Suhrkamp Verlag.
- Fayyad, U. M., Piatetsky-Shapiro G., Smyth, P. (1996). From data mining to knowledge discovery: An overview. *AI Magazine*, 17 (3), 37-54.
- Frank, A. U., 2001. Socio-economic units: Their life and motion. In: A. U. Frank, J. Raper, J.-P. Cheylan, eds. *Life and Motion of Socio-Economic Units*. London: Taylor & Francis, 21-34.
- Frentzos, E., Pelekis, N., Ntoutsi, I., Theodoridis, Y., 2008. Trajectory database systems. In: F. Giannotti und D. Pedreschi, eds. *Mobility, data mining and privacy*. Heidelberg, Berlin: Springer-Verlag, 151-187.
- Galton, A., 1995. Towards a qualitative theory of movement. *Lecture Notes in Computer Science*, 988, 377-396.
- Giannotti, F., Pedreschi, D., 2008. Mobility, data mining and privacy: A vision of convergence. In: F. Giannotti und D. Pedreschi, eds. *Mobility, data mining and privacy*. Heidelberg, Berlin: Springer-Verlag, 1-11.
- Gómez, L., Kuijpers, B., Vaisman, A., 2009. Querying and mining databases using places of interest. *New Trends in Data Warehousing and Data Analysis*, 3, 1-26.
- Güting, R. H., de Almeida, V., T., Ding, Z., 2006. Modeling and querying moving objects in networks. *The VLDB Journal*, 15 (2), 165-190.
- Gudmundsson, J., Laube, P., Wölle, T., 2008. Movement patterns in spatio temporal data. In: S. Shekhar, H. Xiong, eds. *Encyclopedia of GIS*. Heidelberg, Berlin: Springer-Verlag.
- Hershberger, J., Suri, S., 1993. Efficient computation of euclidean shortest paths in the plane. *Proceedings of the 34<sup>th</sup> Annual Symposium on Foundations of Computer Science*, 508-517.
- Hägerstrand, T., 1970. What about people in regional science? *Papers of the Regional Science Association*, 24, 7-21.
- Holyoak, M., Casagrandi, R., Nathan, R., Revilla, E., Spiegel, O., 2008. Trends and missing parts in the study of movement ecology. *PNAS*, 105 (49), 19060-19065.
- Huisman, O., Forer, P., 2005. The complexities of everyday life: Balancing practical and realistic approaches to modelling probable presence in space-time. *SIRC 2005 – The 17<sup>th</sup> Annual Colloquium of Spatial Information Research Centre*. University of Ontario.
- Imfeld, 2000. *Time, points and space – Towards a better analysis of wildlife data in GIS*. PhD Thesis. Universität Zürich.
- Kwan, M.-P., Lee, J., 2004. Geovisualization of human activity patterns using 3D GIS: A time-geographic approach. In: M. Goodchild, D. Janelle, eds. *Spatially integrated social science*. New York, NY: Oxford University Press.

- Lambrinos, D., Möller, R., Labhart, T., Pfeifer, R., Wehner, R. 2000. A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems*, 30, 39-64.
- Laube, P., 2005. *Analysing point motion – Spatio-temporal data mining of geospatial lifelines*. PhD Thesis, Universität Zürich.
- Laube, P., Dennis, T., Forer, P., Walker, M., 2007. Movement beyond the snapshot – Dynamic analysis of geospatial lifelines. *Computers, Environment and Urban Systems*, 31, 481-501.
- Laube, P., Duckham, M., Wolle, T., 2008. Decentralized movement pattern detection amongst mobile geosensor nodes. *Lecture Notes in Computer Science*, 5266, 199-216.
- Laube, P., Imfeld, S., 2002. Analyzing relative motion within groups of trackable moving point objects. In: M.J. Egenhofer, D.M. Mark, eds. *GIScience 2002, LNCS 2478*. Heidelberg, Berlin: Springer-Verlag. 132-144.
- Laube, P., Imfeld, S., Weibel, R., 2005. Discovering relative motion patterns in groups of moving point objects. *International Journal of Geographical Information Science*, 19 (6), 639-668.
- Laube, P., Purves, R. S., 2006. An approach to evaluating motion pattern detection techniques in spatio-temporal data. *Computers, Environment and Urban Systems*, 30, 347-374.
- Laurini R., Thompson D., 1992. *Fundamentals of spatial information systems*. London: Academic Press Ltd.
- Miller, H. J., 1991. Modelling accessibility using space-time prism concepts within geographical information systems. *International Journal of Geographical Information Systems*, 5 (3), 447-453.
- Miller, H. J., Han, J., 2001. Geographic data mining and knowledge discovery: An overview. In: H. J. Miller, J. Han, eds. *Geographic Data Mining and Knowledge Discovery*. London: Taylor & Francis, 3-32.
- Mountain, D. M., 2005. *Exploring mobile trajectories: An investigation of individual spatial behaviour and geographic filters for information retrieval*. PhD Thesis, City University London.
- Mouza, du, C., Rigaux, P., 2004. Multi-scale classification of moving object trajectories. *Proceedings of the 16<sup>th</sup> International Conference on Scientific and Statistical Database Management*, 307-324.
- Mouza, du, C., Rigaux, P., 2005. Mobility patterns. *Geoinformatica*, 9 (4), 297-319.
- Müller, M., Wehner, R., 1988. Path integration in desert ants, *Cataglyphis fortis*. *Proc. Natl.Acad. Sci. U.S.A.*, 85, 5287-5290.
- Nathan, R., Getz, W. M., Revilla, E., Holyoak, M., Kadmon, R., Saltz, D., Smouse, P. E., 2008. A movement ecology paradigm for unifying organismal movement research. *PNAS*, 105 (49), 19052-19059.
- Noyon, V., Claramunt, C., Devogele, T., 2007. A relative representation of trajectories in geographic spaces. *Geoinformatica*, 11, 479-496.
- Openshaw, S., 1984. *The modifiable area unit problem*. Norwich: Geo Books.
- Ottmann, T., Widmayer, P., 2002. *Algorithmen und Datenstrukturen*, 4. Auflage. Heidelberg, Berlin: Spektrum Akademischer Verlag.
- Papadimitriou, C. H., 1995. *Computational Complexity*. Boston, MA: Addison-Wesley.
- Pfeifer, R., Bongard, J., 2006. *How the body shapes the way we think: A new view of intelligence*. Cambridge, MA: MIT Press.

- Pfeifer, R., Scheier, C., 1999. *Understanding intelligence*. Cambridge, MA: MIT Press.
- Sargent, R. G., 1988. A tutorial on validation and verification of simulation models. In: M. Abrams, P. Haigh, J. Comfort, eds. *Proceedings of the 1988 Winter Simulation Conference*, 33-39.
- Sargent, R. G., 2005. Verification and validation of simulation models. In: M. E. Kuhl, N. M. Steiger, F. B. Armstrong, J. A. Joines, eds. *Proceedings of the 2005 Winter Simulation Conference*, 103-143.
- Sarjakoski, L. T., Nivala, A.-M., 2005. Adaption to context – A way to improve usability of mobile maps. In: L. Meng, T. Reichenbacher, A. Tipf, eds. *Map-based mobile services*, 107-123.
- Schmid, S., 2008. *Automated constraint-based evaluation of cartographic generalization solutions*. Master Thesis, Universität Zürich.
- Simon, H. A., 1969. *The science of the artificial*. Cambridge, MA: MIT Press.
- Spaccapietra S., Parent, C., Damiani, M. L., de Macedo, J. A., Porto, F., Vangenot, C., 2008. A conceptual view on trajectories. *Data & Knowledge Engineering*, 65, 126-146.
- Stachowiak, H., 1973. *Allgemeine Modelltheorie*. Wien: Springer-Verlag.
- Tomlin, C. D., 1990. *Geographic information systems and cartographic modeling*. Englewood Cliffs, NJ: Prentice Hall.
- Turchin, P., 1998. *Quantitative analysis of movement: Measuring and modelling population redistribution in animals and plants*. Sunderland, MA: Sinauer Publishers.
- Van de Weghe, N., Kuijpers, B., Bogaert, P., de Maeyer, P., 2005. A qualitative trajectory calculus and the composition of its relations. *Lecture Notes in Computer Science*, 3799, 60-76.
- Van de Weghe, N., Cohn, A. G., de Tré, G., de Maeyer, P., 2006. A qualitative trajectory calculus as a basis for representing moving objects in geographical information systems. *Control and Cybernetics*, 35 (1), 97-117.
- Visvalingam M, Whyatt, J.D., 1993. Line generalisation by repeated elimination of points. *Cartographic Journal*, 30 (1), 46-51.
- Weber, J., Kwan, M.-P., 2003. Evaluating the effects of geographic contexts of individual accessibility: A multilevel approach. *Urban Geography*, 2003, 24 (8), 647-671.
- Wehner, R., Michel, B., Antonsen, P., 1996. Visual navigation in insects: Coupling of egocentric and geocentric information. *Journal of Experimental Biology*, 199, 129-140.
- Wehner, R., Rüber, F., 1979. Visual spatial memory in desert ants, *Cataglyphis bicolor* (Hymenoptera: Formicidae). *Experientia*, 35, 1569-1571.
- Wehner, R., Srinivasan, M. V., 1981. Searching behaviour of desert ants, genus *Cataglyphis* (Formicidae, Hymenoptera). *Journal of Comparative Physiology*, 142, 315-338.
- Wolfson, O., Xu, B., Chamberlain, S., Jiang, L., 1998. Moving object databases: Issues and solutions. *Proceedings of the Tenth International Conference on Scientific and Statistical Database Management*, 111-122.
- Zaho, J., Forer, P., Harvey, A. S., 2008. Activities, ringmaps and geovisualization of large human movement fields. *Information Visualization*, 7 (3-4), 198-209.

Persönliche Erklärung:

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und die den verwendeten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Jürg Eggenberger