

Qualitative Repräsentation von Bewegungsverläufen

Alexandra Musto

Institut für Informatik
der Technischen Universität München

Qualitative Repräsentation von Bewegungsverläufen

Alexandra Musto

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen
Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Gudrun Klinker, Ph.D.

Prüfer der Dissertation:

1. Univ.-Prof. Dr. Dr. h. c. Wilfried Brauer
2. Univ.-Prof. Christian Freksa, Ph.D.

Universität Hamburg

Die Dissertation wurde am 31.08.2000 bei der Technischen Universität München
eingereicht und durch die Fakultät für Informatik am 08.12.2000 angenommen.

Danksagung

Mein Doktorvater Prof. Dr. Dr. h. c. Wilfried Brauer hat es verstanden, mit einer Mischung aus Gewährung größtmöglicher Freiheit einerseits, und schneller und kenntnisreicher Unterstützung bei auftretenden Problemen andererseits, das angenehmste und produktivste Arbeitsklima zu schaffen, das ich mir überhaupt vorstellen kann. Dafür gilt ihm mein Dank.

Die Anregungen, die ich in Diskussionen mit meinem Kollegen Klaus Stein erhalten habe, kann ich gar nicht mehr zählen. Für die hervorragende Zusammenarbeit sei ihm gedankt!

Diese Arbeit entstand im Projekt „Qualitative Repräsentation von Bewegungsverläufen: Kognitive und psychophysische Grundlagen“ im DFG-Schwerpunktprogramm „Raumkognition“. Die Arbeit in einem interdisziplinären Projekt und Schwerpunktprogramm hat mir viele interessante Einblicke in die „anderen“ Disziplinen der Kognitionswissenschaft ermöglicht. Dafür möchte ich nicht nur Prof. Brauer, sondern auch seinen Co-Antragstellern Dr. Daniel Hernández und Dr. Kerstin Schill, an dieser Stelle danken.

Meinen Projektpartnern am IMP Dr. Andreas Eisenkolb und Florian Röhrbein sei für lehrreiche Diskussionen und Tips gedankt. Meinen Kollegen in der KI-Gruppe möchte ich meinen Dank aussprechen für die entspannte Arbeitsatmosphäre, in der diese Arbeit entstehen konnte.

Besonderer Dank gilt den Mitarbeitern des Bremer Projekts „Bildfolgenbasierte semilokale 3D-Landmarken zur Navigation in dynamischen Umgebungen“ Dr. Thomas Röfer, Axel Lankenau und Rolf Müller. Sie haben viele unserer Ideen aufgegriffen und in ihrem semiautonomen Rollstuhl dem Praxistest unterzogen, was meist mit einem erheblichen Arbeitsaufwand verbunden war. Ohne diesen Einsatz wäre die Arbeit erheblich praxisferner geblieben.

Meinem Zweitgutachter Prof. Christian Freksa, PhD, möchte ich für wertvolle Anregungen danken.

Weiterhin danke ich den Studenten Kolja Vogel, Nataliya Skrypnyuk und Korbinian Trumpp für ihren Einsatz bei der Implementierung des MM-TOOLS.

Meinem Ehemann Dr. Thomas Nüßler danke ich für fachliche und moralische Unterstützung sowie dafür, daß er sich dem mühevollen Geschäft des Korrekturlesens unterzogen hat.

Kurze Zusammenfassung der Arbeit

Die Repräsentation von Bewegungsverläufen ist eine wichtige Teilaufgabe in vielen informatischen Anwendungen: Überall, wo sich z.B. Roboter autonom durch eine offene Umgebung (nicht eine abgeschlossene Umgebung wie eine Fertigungsstraße) bewegen sollen, ist sowohl die Eigenbewegung des Roboters als auch die Bewegung anderer Subjekte wahrzunehmen, zu repräsentieren und zu verarbeiten – sei es für Bahnplanung, für Kollisionsvermeidung oder an der Schnittstelle für Instruktionen des Menschen.

Es sind bereits viele quantitative Repräsentationen bekannt, die solche Aufgaben lösen und überall dort nutzbringend eingesetzt werden, wo hinreichend genaue quantitative Meßdaten vorhanden sind. An der Bedienerschnittstelle kann es mit ihnen jedoch zu Problemen kommen: Menschen denken meist nicht in quantitativen Kategorien wie 52° , sondern in qualitativen wie „links vorne“. Gerade wenn naive Benutzer mit technischen Systemen interagieren sollen, wie z.B. mit einem semiautonomen Rollstuhl oder einem Serviceroboter, ist es wichtig, daß die Kommunikation in für den Menschen natürlichen Begriffen stattfinden kann. Besonders an der Benutzungsschnittstelle ist es also notwendig, daß technische Systeme qualitative Repräsentationen verstehen und weiterverarbeiten können.

Die vorliegende Arbeit entstand im Rahmen eines interdisziplinären Projekts in Zusammenarbeit mit dem Institut für medizinische Psychologie der LMU München¹, in dem es um die Modellierung von menschlichen kognitiven Leistungen in der Bewegungswahrnehmung geht. Ein Ziel der Arbeit war es, eine für den Menschen intuitive, qualitative Repräsentation von Bewegungsverläufen zu entwickeln, die – ganz oder teilweise – an solchen Schnittstellen zum Einsatz kommen kann. Durch die Einbettung dieser Arbeit in ein kognitionswissenschaftliches Schwerpunktprogramm war es möglich, neueste psychologische Erkenntnisse über die Wahrnehmung, Repräsentation und Verarbeitung von Bewegungsverläufen zu berücksichtigen, um eine für den Menschen möglichst intuitive Repräsentation zu erhalten.

Aus eigenen [63] und fremden [43, 44, 45] Experimenten wissen wir, daß der Gestalt-Aspekt bei der Bewegungswahrnehmung eine entscheidende Rolle spielt. Deswegen kommt in dem in dieser Arbeit vorgestellten Ansatz zur qualitativen Repräsentation von

¹„Qualitative Repräsentation von Bewegungsverläufen: Kognitive und psychophysische Grundlagen“, Projekt BR609/9-1 bis 9-3 im DFG-Schwerpunktprogramm „Raumkognition“

Bewegungsverläufen der Form eines Bewegungsverlaufs ebenfalls eine große Bedeutung zu.

Die vorliegende Arbeit stellt eine qualitative Repräsentation von Bewegungsverläufen punktförmiger Objekte im zweidimensionalen Raum vor. Die Repräsentation besteht aus zwei Schichten: einer relativ feingranularen vektoriiellen Repräsentation, den Qualitative Motion Vectors (QMV) und einer grobgranularen, abstrakteren Schicht, die die Form von Teilen der Trajektorie als repräsentationale Grundelemente benutzt (die SHAPE-Repräsentation). Die QMV-Repräsentation kann aus beobachteten Bewegungsverläufen erzeugt (und so z.B. der Ausgabe an einen Menschen dienen) oder aber direkt eingegeben werden. Die SHAPE-Repräsentation wird durch Vereinfachung der QMV-Repräsentation mittels Generalisierung, Segmentierung in basale SHAPES, und Klassifizierung dieser basalen SHAPES in komplexere, die einem vordefinierten SHAPE-Vokabular entnommen werden, erzeugt. Die verschiedenen Möglichkeiten, mit qualitativen Vektoren zu rechnen, und die verschiedenen Generalisierungs- und Klassifizierungsalgorithmen, die dazu entwickelt wurden, werden in der vorliegenden Arbeit vorgestellt. Überlegungen zu verschiedenen Referenzsystemen, in denen Bewegungsverläufe gemessen und repräsentiert werden können, sowie zu Vergleichbarkeit und Komplexität der so repräsentierten Bewegungsverläufe, runden die Betrachtungen ab.

Trotz des kognitionswissenschaftlichen Hintergrunds war die Anwendbarkeit der entwickelten Repräsentation und der zugehörigen Algorithmen in technischen Systemen ein wichtiger Punkt. In Zusammenarbeit mit dem Projekt „Bildfolgenbasierte semilokale 3D Landmarken zur Navigation in dynamischen Umgebungen“² im DFG-Schwerpunktprogramm „Raumkognition“ konnten einige unserer Entwicklungen bei der Navigation eines semiautonomen Rollstuhls zum Einsatz gebracht werden. Eine weitere Anwendung, die Erzeugung von Routenbeschreibungen aus Bewegungsdaten, wurde zwar implementiert, konnte aber noch nicht zum praktischen Einsatz gebracht werden.

²Prof. Dr. Bernd Krieg-Brückner und Dr. Thomas Röfer und Kollegen, Universität Bremen.

Inhaltsverzeichnis

Kurze Zusammenfassung der Arbeit	I
1 Einleitung	1
1.1 Repräsentationen von Bewegung	1
1.2 Qualitativ versus quantitativ	2
1.3 Repräsentation versus Schlußfolgern	4
1.4 Bewegung und Kognition	5
1.5 Ziele und Anwendungen	6
1.6 Implementierung	7
1.7 Über diese Arbeit	8
2 Raum und Bewegung	9
2.1 Konzeptualisierungen von Raum	9
2.2 Qualitative räumliche Repräsentationen	11
2.2.1 Qualitative Physik	12
2.2.2 Intervallkalküle	13
2.2.3 Mereotopologische Kalküle	13
2.2.4 Kalküle mit reicherer Raumstruktur	15
2.3 Bewegungsverläufe	18
2.4 Qualitative Repräsentation von Bewegungsverläufen	20
2.4.1 Qualitative Kinematik	21
2.4.2 Axiomatische Ansätze	22
2.5 Welche Art von Bewegungsverläufen repräsentieren wir hier?	26
3 Stufen der Abstraktion	27
3.1 Zwei-Schichten-Architektur	27
3.2 Qualitative Bewegungsvektoren	28

3.2.1	Zeit	30
3.2.2	Richtung	30
3.2.3	Distanz und Geschwindigkeit	31
3.2.4	QMV's	32
3.3	SHAPE-Repräsentation	34
3.4	Gemischte Granularitäten	35
4	Quantity Spaces	37
4.1	Diskrete Intervalle	37
4.1.1	Distanz	37
4.1.2	Richtung	39
4.1.3	Beispiele	40
4.1.4	Rechnen in diskreten Quantity Spaces	41
4.2	Fuzzy-Modellierung von Bewegungsverläufen	47
4.2.1	Fuzzy Numerical Vectors	48
4.2.2	Rechnen mit FNV's	51
4.2.3	Fuzzy Motion Vectors	52
4.2.4	Abbildung FNV in FMV	52
4.2.5	Abbildung FMV in QMV	55
4.2.6	Abbildung QMV in FNV	55
5	Konstruktion der qualitativen Repräsentation	57
5.1	Von numerischen Daten zu QMV's	57
5.2	Generalisierung	58
5.2.1	Σ -Generalisierung	58
5.2.2	ε -Generalisierung	60
5.2.3	Beispiele	64
5.3	Konstruktion der SHAPE-Repräsentation	64
5.3.1	Segmentierung	64
5.3.2	Klassifizierung	68
5.3.3	Weitere Attribute für SHAPES	75
5.3.4	Beispiele	77
6	Referenzsysteme	79
6.1	Egozentrisches versus allozentrisches Referenzsystem	81

6.2	Referenzsysteme in Messung versus Repräsentation	83
6.2.1	Probleme bei egozentrischer Messung	83
6.2.2	Mögliche Lösungen	85
6.2.3	Konsequenzen bei späteren Berechnungen	87
6.3	Lokale Referenzsysteme	89
7	Ähnlichkeit und Komplexität	91
7.1	Ähnlichkeit von Bewegungsverläufen	91
7.1.1	Ähnlichkeit von QMV-Sequenzen	91
7.1.2	Fuzzy-Ähnlichkeit	94
7.1.3	Ähnlichkeit von SHAPE-Sequenzen	95
7.2	Komplexität von Bewegungsverläufen	96
7.2.1	Beschreibungskomplexität von QMV-Sequenzen	96
7.2.2	Komplexität von SHAPE-Sequenzen	98
8	Anwendungen	103
8.1	Roboternavigation	103
8.1.1	Navigieren mittels aufgezeichneter Eigenbewegung	103
8.1.2	Navigieren mit groben, qualitativen Routenbeschreibungen	107
8.2	Von Bewegung zu Routenbeschreibungen	111
9	Schlußbemerkungen	117
9.1	Zusammenfassung	117
9.2	Ausblick	118

Abbildungsverzeichnis

1.1	Qualifizierungsprogramm der 1999er Kunstflug-Europameisterschaft . . .	2
2.1	Die RCC-8-Relationen	14
2.2	Freksasches Doppelkreuz	16
2.3	Experiment zur Reproduktion von Trajektorien. Abbildung aus [63] . . .	20
2.4	Dominanzraum für RCC-8	23
3.1	Qualitative Repräsentation von Bewegungsverläufen in zwei Schichten . .	28
3.2	Qualitative Repräsentation einer Zugfahrt	29
3.3	Qualitative Repräsentation von Bewegungsrichtung	31
3.4	Qualitative Repräsentation von Bewegungsweite	31
3.5	Beispielsequenz	33
3.6	Kunstflugfiguren: Humpty-Bump, Half Cuban Eight, Tail Slide, Eight- side-loop	34
3.7	Bewegungsverlauf mit Landmarken und Generalisierung	36
4.1	Links: Beispielsequenz für Diskretisierung in 8 gleich- oder verschieden große Richtungsbereiche. Rechts: Beispielsequenz für Diskretisierung in 4 oder 8 gleichgroße Richtungsbereiche	41
4.2	Absorptionsregel	44
4.3	Ähnliche Bewegungsverläufe werden unähnlich repräsentiert	47
4.4	Kombination von Richtungs- und Distanzintervallen	49
4.5	Abbildung $FNV \rightarrow FMV$	54
4.6	Fuzzy-Richtungen	56
5.1	Originalspur und Σ -Generalisierung mit einer Schrittweite von $30 \times \text{very-}$ close , symbolisch gerechnet (siehe Kapitel 4.1.4)	60
5.2	Schritte der ε -Generalisierung	61

5.3	Test der ε -Bedingung	62
5.4	Originalspur (fein) und ε -Generalisierung (dick) im MM-TOOL	65
5.5	Originalspur einer Roboterbewegung und durch inkrementelle ε -Generalisierung gefundene Ecken (Kästchen). Die Verbindung der Ecken durch gerade Linien wurde in der Darstellung weggelassen.	66
5.6	Segmentierung einer QMV-Sequenz	68
5.7	Modularer Aufbau der Klassifikation	69
5.8	Klassifikation mit Memory Buffer	74
6.1	Illustration grundlegender Konzepte	82
6.2	Allozentrische und egozentrische Diskretisierung der Richtungsdomäne	83
6.3	Messung im allozentrischen und egozentrischen Referenzsystem	84
6.4	Links: Nichterkennen einer langsamen Kurve bei egozentrischer Messung Rechts: Lösung	86
6.5	Oben links: Zwei Bewegungsspuren in einem Korridornetz. Oben rechts: Graph-Repräsentation mit Erhaltung der egozentrischen Richtungen. Un- ten links und unten rechts: Zwei mögliche Graph-Matchings.	88
6.6	Zusammenhang zwischen intrinsischem und allozentrischem Referenzsystem	89
6.7	Ungerichtetes Referenzsystem	90
7.1	Bewegungsspuren und ihre zugehörigen SHAPE-Sequenzen	101
8.1	Der Bremer Autonome Rollstuhl „Rolland“. Abbildung aus [62]	104
8.2	Drei Bewegungsspuren mit durch Generalisierung gefundenen Ecken. Ab- bildung aus [62].	106
8.3	Unterschiedliche Generalisierungen bei ähnlichen Bewegungsverläufen. Abbildung aus [62].	107
8.4	Plan des 2. Stocks des MZH-Gebäudes in der Bremer Universität und der Route A (gestrichelte Linie, ca. 65 m). Abbildung aus [60].	110
8.5	Pfad des Rollstuhls bei einem erfolgreichen Versuch, die Route A abzufah- ren. Die Dreiecke geben Position und Orientierung des Rollstuhls wieder in Intervallen von ungefähr .25 s (aus [60]).	112
9.1	Trajektorien mit orientierten Objekten	120

1 Einleitung

1.1 Repräsentationen von Bewegung

„Emil überlegte nicht lange, sondern rannte, was er konnte, davon. Über eine Wiese, an vielen Bäumen vorbei, durch einen Bach, dem Wolkenkratzer zu. Manchmal sah er sich um; der Zug donnerte hinter ihm her, ohne abzulassen.“ ([46], S. 55f)

„Und während alle Bilder aus seinem Kopfe wichen [...], richtete er sich mechanisch von seinem Lager auf. Ganz langsam [...] richtete er sich auf. Erst einen Arm, ...dann stützte er den Oberkörper auf, dann schob er ein Knie unter der Decke hervor, ...dann...: doch plötzlich eilte er mit bloßen Füßen, auf den Zehen zu Basini herüber und setzte sich auf den Rand des Bettes.“ ([61], S. 103)

In beiden Zitaten wird ein Bewegungsverlauf beschrieben – zwar auf recht unterschiedlichen Weise, aber doch beide Male mit den Mitteln der natürlichen Sprache.

Die folgende Beschreibung:

„Die Kür bestand aus einem doppelten Toe-Loop, gefolgt von einem Doppelaxel, einem Bogen rechts vorwärts einwärts, einem einfachen Lutz und einer abschließenden Pirouette.“

ist zwar auch natürlichsprachlich, aber schon stärker formalisiert. Um sich den zugehörigen Bewegungsverlauf vorstellen zu können, muß man wissen, wie die Eislaufiguren, deren Namen als Abkürzung für Teile des Bewegungsverlaufs benutzt werden, aussehen. Eine ähnlich formalisierte Beschreibung, die die natürliche Sprache gar nicht mehr benötigt, findet sich in [Abbildung 1.1](#).

Bewegungsverläufe und Repräsentationen davon begegnen uns überall in unserem täglichen Leben, ob es sich dabei um eine Wegbeschreibung handelt oder darum, die Bewegung der in der Fußgängerzone entgegenkommenden Passanten wahrzunehmen und vorauszuberechnen, um ausweichen zu können.

Auch in der Physik und in den Ingenieurwissenschaften spielt Bewegung und deren Repräsentation eine große Rolle: Exakte Repräsentationen und Berechnungen sind dazu nötig, die Flugbahn einer Apollo-Kapsel zu berechnen, die Umlaufbahn der Jupitermonde, oder die Trajektorie eines Roboterarms.

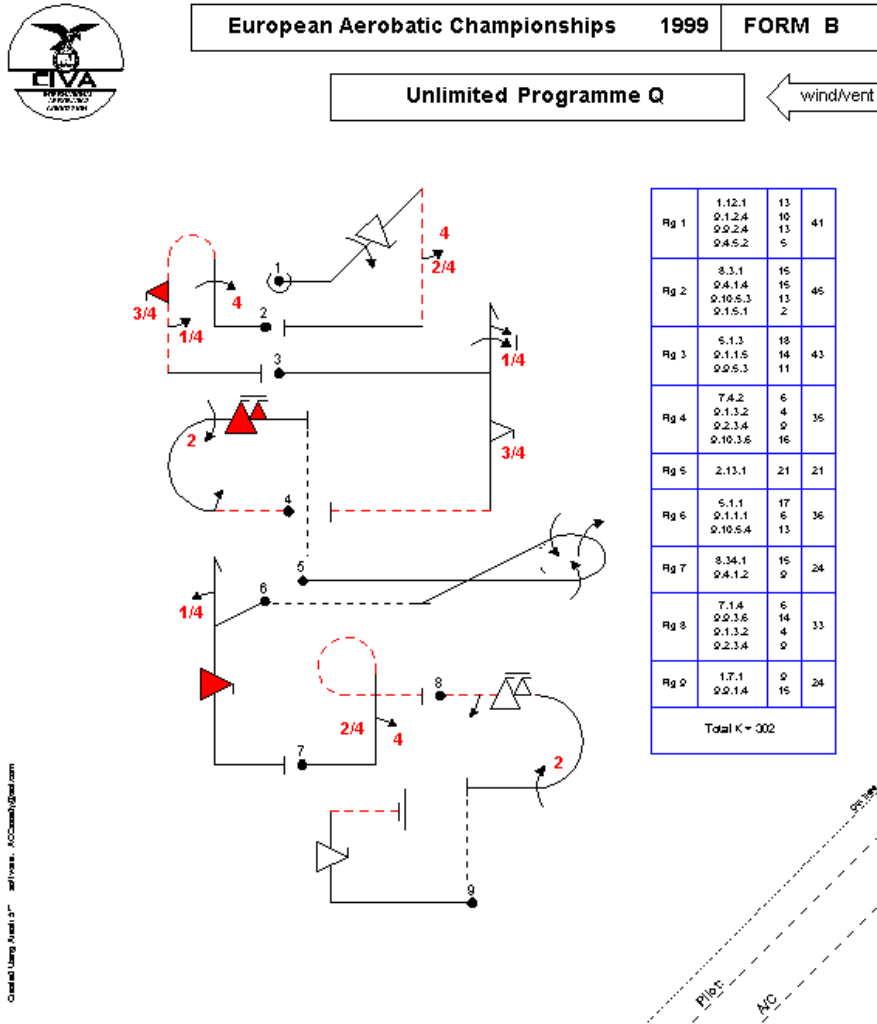


Abbildung 1.1: Qualifizierungsprogramm der 1999er Kunstflug-Europameisterschaft

1.2 Qualitativ versus quantitativ

Bei den letztgenannten Beispielen handelt es sich um quantitative, numerische Repräsentationen. Im Titel dieser Arbeit ist aber von *qualitativen* Repräsentationen die Rede. Was bedeutet qualitativ im Unterschied zu quantitativ? Bei beiden Arten der Repräsentation geht es darum, bestimmte *Größen* zu benennen: Wie lang ist eine Strecke, wie schwer ein Gewicht, wie groß ein Land? Beiden gemeinsam ist es auch, daß man Größen benennt, indem man sie zu anderen Größen in Relation setzt. Bei einer quantitativen

Repräsentation ist die Vergleichsgröße eine vordefinierte Einheit, z.B. ein Meter, der als Urmeter auch durchaus real und physikalisch vorliegen muß. Bei quantitativen Repräsentationen liegen Daten einer bestimmten Dimension (Länge, Fläche, ...) immer im Vergleich zu derselben Größe vor, also z.B. zum Meter¹. Bei qualitativen Repräsentationen muß dies nicht so sein. So kann z.B. die Länge von Haaren im Vergleich zu einem Streichholz angegeben werden, die Länge einer Wegstrecke in einer Stadt hingegen in Gebäudeblöcken, ohne daß dies zu Verwirrungen beim Hörer führen würde.

Ganz wesentlich bei qualitativen Repräsentationen ist, daß sie, im Gegensatz zu quantitativen Repräsentationen, einen Vergleich immer nur „so ungefähr“ machen. Wenn jemand erzählt, die beste Freundin habe sich die Haare streichholzkurz schneiden lassen, so erwartet niemand, daß die Haare exakt genauso lang sind wie ein (evtl. genormtes) Streichholz, sondern eben nur ungefähr so lang. Die qualitative Längenangabe bezeichnet ein Intervall, das sich bei „streichholzkurz“ etwa von 2 bis 5 cm erstreckt.

In [84] wird dargelegt, daß die räumlichen mentalen Modelle [37], die der Mensch zur Verfügung hat, nur über ganz grobe Relationen zwischen den einzelnen Elementen verfügen und meist keinerlei metrische Information enthalten. Typische von Menschen verwendete räumliche Ausdrücke sind z.B. „nahe bei“, „links von“, „östlich von“. Nicht nur, daß in räumlichen mentalen Modellen kaum metrische Information vorkommt, es gibt auch häufig systematische Verzerrungen, die die Einschätzung von Distanzen und Richtungen beeinflussen. So wird z.B. in [79] festgestellt, daß die Distanz von einem normalen Gebäude zu einer Landmarke² kleiner eingeschätzt wird als die Entfernung von der selben Landmarke zum selben Gebäude. Das führt zu asymmetrischen Distanzen.

Da in der menschlichen Kognition kaum metrische Informationen repräsentiert werden, und wenn, dann häufig sehr schlecht, ungenau oder mit systematischen Verzerrungen behaftet, ist es sinnvoll, in der Mensch-Maschine-Interaktion nicht zuviel Gewicht auf quantitative Repräsentationen zu legen, und stattdessen mehr qualitative Repräsentationen wie z.B. das oben genannte „streichholzkurz“ verwenden.

Bei qualitativen Repräsentationen ist die Auswahl einer geeigneten Vergleichsgröße ein wichtiger Faktor. Dient die qualitative Repräsentation dazu, jemandem eine Vorstellung von etwas zu geben, also zur Kommunikation mit einem Menschen, sollte die Vergleichsgröße nach kognitiven Faktoren ausgewählt werden.

So haben z.B. die meisten Bundesbürger eine grobe Vorstellung davon, wie groß die

¹ ... oder können, falls in Bezug zu unterschiedlichen Vergleichsmaßen vorliegend, mittels Formeln in die jeweils andere Einheit umgerechnet werden.

²Landmarke wird in [79] definiert als räumlicher Referenzpunkt, der benutzt wird, um andere, Nichtreferenzpunkte, zu lokalisieren. Im Experiment wurde die Auffälligkeit und Wichtigkeit von Örtlichkeiten einer den Versuchspersonen bekannten Umgebung indirekt über Kenntnis der jeweiligen Örtlichkeit, Anzahl der Besuche des Orts und Einschätzung der historischen und kulturellen Bedeutung gemessen. Es wurde dann angenommen, daß bei Paarungen von Orten mit hoher und niedriger Auffälligkeit der mit der hohen Auffälligkeit als Referenzpunkt benutzt wird. Die Kontrollgruppe erhielt Paare von Orten gleicher Auffälligkeit.

Bundesrepublik ist, könnten aber nicht sagen, ob eine Fläche von 357.042 km² größer, kleiner oder ungefähr gleichgroß ist. Der Quadratkilometer ist für die meisten Menschen und die meisten Zwecke eine ungeeignete Vergleichsgröße.

In einem Lernprogramm zur Geographie kann z.B. der Schüler die Information erhalten, daß Namibia 823.145 km² groß ist und 1,8 Mio Einwohner hat. Dies ist zwar korrekt, vermittelt aber den meisten Menschen keine Vorstellung davon, was das bedeutet. Wenn jedoch (zusätzlich oder ausschließlich) die Information gegeben wird, daß Namibia etwa so groß ist wie die BRD und Frankreich zusammen und etwa so viele Einwohner wie Wien hat, wird sich vermutlich jeder Leser aus dem deutschen Sprachraum vorstellen können, daß man es mit einem großen, leeren Land zu tun hat.

Die Vergleichsgrößen für qualitative Repräsentationen richten sich außerdem meist nach dem Kontext, in dem die Repräsentation sich befindet. In obigem Beispiel wird die Größe eines Landes mit der eines anderen Landes verglichen und nicht etwa mit dem Mittelmeer oder einem Fußballfeld. Aber auch Eigenheiten des Adressaten, an den sich eine qualitative Aussage richtet, sind wichtig.

So wird obiges Namibia-Beispiel einem amerikanischen Leser unter Umständen gar nichts sagen, weil er vielleicht weder eine Vorstellung davon hat, wie groß die Bundesrepublik ist, noch, wie viele Einwohner Wien hat. Für ihn müssen andere Vergleichsgrößen ausgewählt werden: Deswegen gibt das CIA World Factbook³ für jedes dort beschriebene Land nicht nur die km²-Zahlen an, sondern auch meist einen der amerikanischen Staaten als Vergleichsgebiet. So ist die Landfläche von Namibia dort beschrieben als „slightly more than half the size of Alaska“, die von Simbabwe als „slightly larger than Montana“.

Aber auch, wenn es sich nicht um Benutzerinteraktion handelt, können qualitative Repräsentationen sinnvoll sein. In vielen Anwendungen ist die Meßgenauigkeit so schlecht, daß eine quantitative Angabe nur eine gar nicht gegebene Genauigkeit vortäuschen würde, denn bei quantitativen Angaben wie z. B. 923mm nimmt man an, daß es sich auch um 923mm handelt, und nicht um „irgendeinen Wert zwischen 900 und 950mm“. Oft wird eine Größe auch gar nicht in der möglichen Meßgenauigkeit benötigt. So geht es in einer regelungstechnischen Aufgabe z. B. manchmal nur darum, zu überwachen, ob eine bestimmte Größe steigt oder fällt. Dann reicht als Vergleichsgröße für die Differenz zweier Werte die Null; von Interesse ist, ob sie größer, kleiner oder gleich Null ist.

1.3 Repräsentation versus Schlußfolgern

Zu Anfang wurden einige Beispiele von nicht-mathematischen Repräsentationen von Bewegungsverläufen genannt, die im täglichen Leben vorkommen. Aber kann man mit Repräsentationen dieser Art auch schlußfolgern, wie man mit quantitativen Repräsentationen rechnen kann? Menschen sind ja ganz offensichtlich in der Lage, aus solchen

³<http://www.odci.gov/cia/publications/factbook/>

grogen, qualitativen Angaben neues Wissen zu erschließen. So kann man z. B. aus der Beschreibung „Hans betrat das Zimmer, drehte sich nach links und ging auf das Fenster zu“ einige Aussagen inferieren, z. B. daß Hans jetzt im Zimmer ist, daß das Fenster links von der Tür ist, daß Hans' Position am Ende links von der Tür ist, usw.

In der Disziplin des qualitativen räumlichen Schließens (QRS) geht es darum, qualitative Raumrepräsentationen und Methoden, wie man mit ihnen schlußfolgern kann, in Kalkülen zu formalisieren. QRS ist mittlerweile ein eigenes Forschungsgebiet in der Künstlichen Intelligenz (KI). Einen Kurzüberblick über die vielfältigen Anstrengungen und Ansätze, die auf diesem Gebiet bisher erbracht wurden, gibt [13]; einen etwas älteren vergleichenden Überblick findet man in [32].

Allerdings existieren bisher wenige Anwendungen, bei denen es nicht nur um die qualitative Repräsentation von Raum, sondern um echtes Schlußfolgern mit qualitativen Repräsentationen geht. So muß man zwar überall, wo man z.B. vom Benutzer lediglich qualitative Eingaben erwarten kann, diese weiterverarbeiten können. Die üblicherweise angewandten Techniken der diskreten Constraint-Erfüllung haben allerdings den Nachteil der Ineffizienz (NP-Vollständigkeit). Deswegen ist es sinnvoll, überall, wo andere, effizientere Techniken bekannt sind, diese auch zu nutzen. Wo numerische Daten von vorneherein bekannt sind (also das System nicht auf qualitative Benutzereingaben angewiesen ist), ist es meist nachgerade unsinnig, nur mit qualitativen Größen zu rechnen. Hier müssen numerische und qualitative Repräsentationen nebeneinander existieren: die numerische, weil das Rechnen mit ihnen in den allermeisten Fällen effizienter ist⁴, und die qualitative, weil sie für die Benutzerinteraktion die dem Menschen adäquatere Repräsentation ist.

1.4 Bewegung und Kognition

Die vorliegende Arbeit entstand im Rahmen eines interdisziplinären Projekts in Zusammenarbeit mit dem Institut für medizinische Psychologie der LMU München⁵, in dem es um die Modellierung von menschlichen kognitiven Leistungen in der Bewegungswahrnehmung geht. Begleitet wurde dies von psychophysischen Experimenten zu Wahrnehmungs- und Diskriminationsschwellen in der Bewegungswahrnehmung im Millisekundenbereich und Experimenten zu Gedächtnis- und Reproduktionsleistung bei Wahrnehmung längerer Trajektorien.

Bewegung ist für die visuelle Wahrnehmung wichtig: Die Wahrnehmung völlig statischer Stimuli ist für den menschlichen Wahrnehmungsapparat gar nicht möglich. Der Tremor

⁴Eine Ausnahme sind z.B. nichtlineare Optimierungsprobleme, bei denen mit entsprechenden Heuristiken versehene Constraint erfüllungsalgorithmen mittlerweile effizienter sind als klassische Lösungen, siehe auch [72].

⁵„Qualitative Repräsentation von Bewegungsverläufen: Kognitive und psychophysische Grundlagen“, Projekt BR609/9-1 bis 9-3 im DFG-Schwerpunktprogramm „Raumkognition“

des Auges stellt sicher, daß auch bei Fixation eines unbewegten Gegenstandes immer eine minimale Bewegung vorhanden ist; würde man ein statisches Bild auf die Retina projizieren, würde man nach kurzer Zeit den Eindruck eines weißen Rauschens erhalten [71].

Das Verständnis dessen, was bei Bewegungswahrnehmung vor sich geht, ist wichtig für das Verständnis visueller Wahrnehmung überhaupt. Frühere Ergebnisse aus der psychologischen Forschung [43, 44, 45] weisen darauf hin, daß dem Gestalt-Aspekt bei der Bewegungswahrnehmung und -verarbeitung eine große Bedeutung zukommt: Nur anhand einer sich bewegenden Punktvolke können Versuchspersonen innerhalb kürzester Zeit erkennen, daß diese Punktvolke einen gehenden oder rennenden Menschen, ein tanzendes Paar, oder ähnliches darstellt. Auch eigene Experimente im o.g. Projekt legen den Schluß nahe, daß der Formaspekt bei der Bewegungswahrnehmung eine große Rolle spielt: Trajektorien, deren Spuren gewisse markante Formen aufweisen, konnten leichter erinnert und reproduziert werden (siehe Abschnitt 2.3 und [63]). Eine qualitative Repräsentation von Bewegungsverläufen, die dazu geeignet sein soll, dem Menschen Bewegungsverläufe möglichst natürlich zu kommunizieren, muß diesen Formaspekt also unbedingt berücksichtigen.

1.5 Ziele und Anwendungen

Trotz des kognitionswissenschaftlichen Hintergrunds ist die Anwendbarkeit der entwickelten Repräsentation und der zugehörigen Algorithmen in technischen Systemen in dieser Arbeit ein wichtiger Punkt. Überall, wo sich z.B. Roboter autonom durch eine offene⁶ Umgebung bewegen sollen, ist sowohl die Eigenbewegung des Roboters als auch die Bewegung anderer Subjekte wahrzunehmen, zu repräsentieren und zu verarbeiten – sei es für Bahnplanung, für Kollisionsvermeidung oder an der Schnittstelle für Instruktionen des Menschen. Andere Anwendungen, bei denen die Repräsentation von Bewegungsverläufen wichtig bis essentiell ist, sind z.B. die Verkehrsüberwachung, Verkehrstelematik oder automatische Indizierung von Bildfolgen in Multimediadatenbanken. An der Benutzungsschnittstelle sind, wie bereits dargelegt, *qualitative* Repräsentationen von besonderer Wichtigkeit.

Ein Ziel der Arbeit war es, eine für den Menschen intuitive, qualitative Repräsentation von Bewegungsverläufen zu entwickeln, die – ganz oder teilweise – an solchen Schnittstellen zum Einsatz kommen kann. Durch die Einbettung dieser Arbeit in ein kognitionswissenschaftliches Schwerpunktprogramm war es möglich, neueste psychologische Erkenntnisse über die Wahrnehmung, Repräsentation und Verarbeitung von Bewegungsverläufen berücksichtigen, um eine für den Menschen möglichst intuitive Repräsentation zu erhalten. Der Formaspekt sollte dabei aus den oben dargestellten Gründen eine entscheidende

⁶im Gegensatz zu abgeschlossen wie z. B. eine Fertigungsstraße

Rolle spielen.

Wechsel der Granularität oder des Referenzsystems kommen in natürlichsprachlichen Beschreibungen von Bewegung häufig vor (siehe z. B. [85]). Für eine möglichst intuitive Repräsentation von Bewegungsverläufen ist es wichtig, daß sie auf unterschiedlichen und vor allem gemischten Granularitäten bzw. Abstraktionsstufen und mit unterschiedlichen Referenzsystemen funktioniert. Das macht die Repräsentation auch flexibler und vielseitiger einsetzbar. In der zu entwickelnden qualitativen Repräsentation sollte deshalb der Behandlung verschiedener Granularitäten und Referenzsysteme ein Hauptaugenmerk gewidmet werden.

Die Repräsentation sollte aber nicht nur für Benutzungsschnittstellen geeignet sein, sondern auch für andere Aufgaben in technischen Systemen. Zum Test für ihre Praxistauglichkeit haben wir eine Anwendung für unsere Repräsentationen und Algorithmen in einem autonomen Roboter gesucht. In Zusammenarbeit mit dem Projekt „Bildfolgenbasierte semilokale 3D-Landmarken zur Navigation in dynamischen Umgebungen“⁷ im DFG-Schwerpunktprogramm „Raumkognition“ konnten einige unserer Entwicklungen bei der Navigation eines semiautonomen Rollstuhls zum Einsatz gebracht werden. Eine weitere Anwendung, die Erzeugung von Routenbeschreibungen aus Bewegungsdaten, wurde zwar implementiert, konnte aber noch nicht zum praktischen Einsatz gebracht werden. Mehr Information dazu findet sich in Kapitel 8.

Neben dem Anspruch, für die Verwendung in der Benutzerinteraktion geeignet zu sein, war ein weiteres Ziel bei der Entwicklung der qualitativen Repräsentation, gewisse Aspekte der Bewegungswahrnehmung und -verarbeitung modellieren zu können, z. B. bei der Richtungsdiskrimination, oder bei Gedächtnis- und Reproduktionsleistungen bei der Wiedergabe von Bewegungsverläufen. Die Repräsentation sollte deswegen modular und parametrisierbar gehalten werden, um neue psychologische Erkenntnisse an möglichst vielen Stellen auch nachträglich einfließen lassen zu können.

1.6 Implementierung

Außer der praktischen Anwendung im Navigationskontext wurden die meisten der vorgestellten Repräsentationen und Algorithmen in einer Testumgebung implementiert, dem MM-TOOL. Das MM-TOOL schafft eine Möglichkeit, Bewegungsverläufe zu Testzwecken mit der Maus einzugeben. Die so eingegebenen, realen Bewegungsverläufe können dann mit verschiedenen Algorithmen bearbeitet, in die qualitative Repräsentation umgewandelt und weiterverarbeitet werden. Dies erlaubt es, die entwickelten Repräsentationen und Algorithmen anhand realer Bewegungsverläufe zu testen und weiterzuentwickeln.

Das MM-TOOL ist in Java geschrieben und modular aufgebaut. Alle qualitativen Repräsentationen werden über Interfaces angesprochen und können so im Betrieb flexibel

⁷Prof. Dr. Bernd Krieg-Brückner und Dr. Thomas Röfer und Kollegen, Universität Bremen.

ausgetauscht werden, um die Effekte der unterschiedlichen Repräsentationen zu testen. Die Algorithmen sind so programmiert, daß sie auf qualitative und quantitative Repräsentationen gleichermaßen anwendbar sind. Die in der Arbeit genannten Beispiele wurden, wo nicht anders erwähnt, mit dem MM-TOOL erzeugt und gerechnet.

1.7 Über diese Arbeit

Kapitel 2 gibt einen kurzen Überblick über qualitative Repräsentationen von Raum und Bewegung. In Kapitel 3 legen wir die grundlegenden Konzepte und die zweischichtige Architektur, die der Repräsentation zugrundeliegt, dar. Kapitel 4 beschäftigt sich mit der Abbildung numerischer Werte in die diskreten Intervalle, die die Grundlage der qualitativen Repräsentation sind, und dem Rechnen (oder Schließen) mit den qualitativen Repräsentationen. Außerdem wird hier eine Fuzzy-Modellierung von Bewegungsverläufen und deren Abbildung in die qualitative Repräsentation vorgestellt. Kapitel 5 zeigt lückenlos den Weg von numerischen Daten bis zur abstraktesten Stufe der zweischichtigen qualitativen Repräsentation, der SHAPE-Repräsentation, auf. Hier befindet sich die Beschreibung und Analyse der meisten Algorithmen, die im Rahmen dieser Arbeit entwickelt wurden. In Kapitel 6 werden die Probleme und Möglichkeiten, die sich aus der Verwendung verschiedener Referenzsysteme bei Messung und Repräsentation eines Bewegungsverlaufs ergeben, referiert. In Kapitel 7 werden kurz einige Gedanken zu Ähnlichkeit von zwei (qualitativ repräsentierten) Bewegungsverläufen und der Komplexität eines Bewegungsverlaufs dargelegt. Kapitel 8 schließlich stellt einige der Anwendungen, die aus dieser Arbeit hervorgegangen sind, vor. Die Arbeit schließt mit einer kurzen Zusammenfassung der vorgestellten Repräsentationen und Algorithmen und einem Ausblick auf mögliche Weiterentwicklungen und Ergänzungen in Kapitel 9.

2 Raum und Bewegung

2.1 Konzeptualisierungen von Raum

Konzeptualisierungen von Raum tauchten sehr früh in der Wissenschaftsgeschichte auf und wurden meist von der Frage angestoßen, was Bewegung sei, was es den Dingen erlaubt, sich zu ändern, und wie die Welt – der Raum – beschaffen sei, der eben solches ermöglicht. Schon die antiken griechischen Philosophen, insbesondere die in der pythagoräischen Tradition stehenden, beschäftigten sich damit [66].

Schon Heraklit (535-475 vor Christus) beschäftigte das Phänomen der Bewegung, Veränderung. Er ging dabei von folgender Frage aus: Wenn ein Ding sich verändert, wie kann man dann noch sagen, daß es dasselbe Ding sei? Wenn sich Dinge verändern, ist die Vorstellung von Identität unhaltbar, wenn die Identität bleibt, hat sich das Ding nicht wirklich verändert. Seine Schlußfolgerung: „alles fließt“, d.h. alles, was wirklich existiert, sind die Prozesse, die die Dinge ändern, hat einige Berühmtheit erlangt.

Parmenides dagegen kommt zu der umgekehrten Schlußfolgerung: Wenn Entgegengesetztes identisch ist, dann gibt es keine wirkliche Veränderung. Wenn alle Dinge eins sind, verändert sich durch die Prozesse der Welt ja nichts wirklich. Parmenides entwickelt daraufhin, ausgehend vom Phänomen der Veränderung und von Heraklits Theorie, eine Theorie über die Struktur von Materie, Raum und Zeit, die aus ein wenig basalen Annahmen deduziert ist und zeigen soll, daß Veränderung unmöglich ist. Zu Beginn steht die Annahme, daß nur das existiert, was ist, und daß, was nichts ist, nicht existieren kann. Somit kann der leere Raum nicht existieren, denn er ist nichts. Deswegen ist die Welt voll von Materie und hat keine Teile. Also ist Bewegung und Veränderung unmöglich, da weder Teile existieren noch ein leerer Raum, der es Dingen erlauben würde, sich zu bewegen.

Parmenides' so offen der Empirie widersprechende Theorie führte dann zur ersten atomistischen Theorie durch Demokrit, der die Falschheit von Parmenides' Voraussetzungen aus der Falschheit seiner Schlußfolgerungen deduzierte: Da Bewegung existiert, kann die Welt nicht aus einem unveränderlichen, soliden Block bestehen. Demzufolge muß die Welt aus Teilen bestehen, mit leerem Raum zwischen ihnen. Die kleinsten, unteilbaren Einheiten sind die Atome, die sich nicht verändern. Die Veränderung der Dinge hingegen besteht aus einer Neuordnung der Atome und damit aus Bewegung.

Demokrit setzte also der Vorstellung Parmenides', es sei nichts anderes als Materie, eine andere Vorstellung entgegen: nämlich die vom Raum als eigenständig existierender Entität, einer Art Container für die Materie, unabhängig und vor ihr existierend. Empirisch gestützt wird dies durch die Existenz von Bewegung.

Hier treten also schon bei den alten Griechen, 400 Jahre vor Christi Geburt, die zwei zentralen Konzeptualisierungen von Raum auf: nämlich Raum als relativer Größe, abhängig von Materie existierend, und Raum als absoluter Größe, unabhängig von Materie existierend, und das zentrale Problem, das es zu bewältigen gilt, ist das Problem der Veränderung und Bewegung.¹

Wenig später entwickelte Euklid eine systematische und umfassende mathematische Theorie, die das basale Werkzeug zur Beschreibung und Erklärung des physikalischen Raumes zur Verfügung stellte. Wegen ihres langanhaltenden Erfolges wurde die euklidische Geometrie lange nicht lediglich als mathematische, abstrakte Theorie gesehen, sondern vielmehr als theoretische Grundlage für die Beschreibung der physikalischen Welt — d.h. man nahm an, die Welt gehorche tatsächlich den euklidischen Gesetzen, und die Geometrie sei somit keine mathematische, sondern eine physikalische Theorie. Erst viel später, im 19. Jahrhundert, wurden auf teilweise anderen Axiomen aufbauende Geometrien entwickelt, mit denen man in anderen (z.B. gekrümmten) Räumen rechnen kann.

Die Vorstellungen vom Raum als lediglich relativ zu Materie existierend und vom Raum als Absolutum wurden allerdings auch später in moderneren wissenschaftlichen Theorien wieder aufgegriffen, vornehmlich durch Newton und Kant im Rahmen der Newtonschen Mechanik, sowie durch Leibniz als Reaktion auf einige der Implikationen daraus.

Die Newtonsche Mechanik versucht ebenfalls, das Phänomen der Bewegung von Körpern im Raum zu erklären. Newton nahm den Raum, wie in der euklidischen Geometrie beschrieben, als gegeben an und entwickelte Formeln, um die Bewegung von festen Körpern zu berechnen und vorauszusagen. Dabei führte er physikalische Größen wie Masse und Kraft ein.

Seine Definition des Inertialsystems als Bezugssystem, in dem keine Trägheitskräfte auftreten, in dem also ein kräftefreier Massenpunkt in Ruhe oder gleichförmiger Bewegung bleibt, und die Feststellung, daß in allen Inertialsystemen die Gleichungen der Mechanik die selbe Form haben, haben zur Konsequenz, daß es kein einzelnes, ausgezeichnetes Bezugssystem gibt und somit keine absoluten Positionen im Raum festgestellt werden können. Mit der Newtonschen Mechanik wird also die Vorstellung von einem absoluten Raum, in dem man absolute Positionen feststellen kann, verworfen, denn ungeachtet Newtons eigener, philosophischer Verteidigung des absoluten Raums widerspricht diese Vorstellung seiner Theorie. Interessant ist hierbei, daß ein räumliches Bezugssystem

¹Der Gegensatz absolut-relativ wird uns in dieser Arbeit später noch begegnen, wenn es um die Referenzsysteme geht, in denen man Bewegung betrachten kann.

durch dynamische Aspekte definiert wird wie die gegenseitigen Anziehungen der Körper. Immanuel Kant, ein weiterer prominenter Vertreter der Vorstellung eines absoluten Raumes, bringt als erster einen anderen, interessanten Aspekt in diese Diskussion ein. In seiner Kritik der reinen Vernunft [56] vertritt er die Auffassung, daß Raum und Zeit keine normalen Entitäten der physikalischen Welt sind, sondern Teil des menschlichen Geistes und lediglich Werkzeug zum Begreifen der wirklichen Welt – wie überhaupt der Mensch keine Naturgesetze erkennen kann, sondern sein Geist sie in die Natur hineinlegt. Kants Glaube an die absolute Wahrheit mancher wissenschaftlicher Theorien wie die euklidische Geometrie und die Newtonsche Mechanik verhinderten allerdings, daß er Raum und Zeit als ledigliche Abstraktionen des menschlichen Geistes auffassen konnte – womit er sich erstaunlich an die Ansichten mancher zeitgenössischer Kognitionswissenschaftler angenähert hätte. Wie aber kann etwas einerseits dem menschlichen Geist entspringen und auf der anderen Seite absolut wahr sein wie die Newtonsche Mechanik? Kant löst dieses Dilemma, indem er behauptet, der menschliche Geist sei ausgestattet mit einer Menge von a priori-Erkenntnissen, die notwendigerweise wahr seien. Wenn ein Mensch eine Ordnung in der Natur findet, so bildet er die Ordnung in seinem Geist, die einer absolut wahren a priori-Erkenntnis entspricht, lediglich auf die Natur ab. Die Kantsche Vorstellung von Raum und Zeit ist deswegen die einer absolut wahren a priori-Erkenntnis, die nicht durch die Erfahrung gebildet wird, sondern im Gegenteil benutzt wird, um Beobachtungen der Welt zu ordnen. Die Newtonsche Mechanik und die euklidische Geometrie sind Produkte dieser a priori-Erkenntnis und deswegen absolut wahr. Die Newtonsche Mechanik wurde erst sehr spät von der Einsteinschen Relativitätstheorie abgelöst. In dieser Theorie sind Raum und Zeit nicht zwei getrennte Entitäten, sondern untrennbar zur „Raumzeit“ nach Minkowski verwoben.

2.2 Qualitative räumliche Repräsentationen

Die Physik (Newtonsche oder Einsteinsche) und die Geometrie (euklidisch oder nichteuklidisch) liefern nun durchaus befriedigende und umfassende Theorien zur Beschreibung von Raum. Wozu qualitative Raumrepräsentationen? Diese sind, wie bereits in Kapitel 1 erwähnt, überall da interessant, wo die Raumkognition des Menschen in die informativische Aufgabenstellung mit einfließt. Das kann zum Beispiel an der Mensch-Maschine-Schnittstelle von Informationssystemen sein, die etwas mit dem Raum zu tun haben, z.B. Geoinformationssystemen oder Multimediatelefonen [6, 7, 8]. Wenn diese Schnittstelle intuitiv gehalten werden soll, müssen in dem System qualitative räumliche Anfragen formuliert werden können, wie z.B. „Gib mir alle Bilder zurück, in denen ein Auto in einer Garage steht“, oder „Zeige mir die Regionen, in denen sich ein geplantes Wasserschutzgebiet mit landwirtschaftlicher Nutzfläche überlappt“. Aber auch Agenten, die sich frei im Raum bewegen, sollten idealerweise über intuitive Schnittstellen zum Menschen

verfügen. Ein Serviceroboter für den Haushalt muß grobe räumliche Relationen ebenso verstehen wie ein autonomer Rollstuhl, der über Spracheingabe gelenkt wird. Nicht nur für die Eingabe des Menschen, auch für die Ausgabe an den Menschen sind qualitative räumliche Repräsentationen nötig, z.B. für Navigationssysteme, Museums- und Stadtführer [3, 83]. Ein ausführlicher Überblick über Techniken des qualitativen räumlichen Schließens, die sich hauptsächlich mit statischen Aspekten des Raums beschäftigen, wird in [13] gegeben. Aus Gründen der Vollständigkeit geben wir eine kürzere Übersicht mit teilweise anderem Schwerpunkt im Folgenden.

2.2.1 Qualitative Physik

Die Anfänge des qualitativen räumlichen Schließens sind in der naiven Physik zu finden. Hayes forderte 1979 in seinem „manifesto“ [41] eine axiomatische Theorie der naiven Physik, also eine Axiomatisierung des Wissens und der Vorstellungen, auf die Menschen zurückgreifen können, wenn sie über die Welt schlußfolgern. Hier werden erstmals die Measure Spaces eingeführt, die uns in Kapitel 4 noch als Quantity Spaces begegnen werden. Interessanterweise glaubt Hayes, Bewegung müsse in der naiven Physik als prä-Galileisch beschrieben werden, d.h. die Newtonschen Gesetze der Bewegung dürfen nicht gelten. Hayes unterscheidet zwei verschiedene mögliche Konzeptualisierungen von Bewegung: „Displacement Motion“, die über die ganze Bewegung hinweg eine bewegende Kraft erfordert und aufhört, wenn diese Kraft nicht mehr ausgeübt wird (Hayes nennt das „griechisch“) und „Trajectory Motion“, die den üblichen Newtonschen-Galileischen Gesetzen folgt. Prä-Galileische Physik beinhaltet einige falsche Vorstellungen; eben z.B. daß sich etwas nur bewegt, wenn eine Kraft ausgeübt wird, oder daß ein im Kreis geschleudertes Stein aus einer Schlinge radial wegfliegt und nicht tangential. Warum aber sollte man so etwas Kontrafaktisches modellieren wollen? Weil genau diese Vorstellungen so auch in den mentalen Modellen der Menschen vorkommen (siehe z.B. [57]) und es für die Benutzerinteraktion wichtig sein kann, diese teilweise falschen Vorstellungen zu modellieren, z.B. um zu helfen, mögliche Benutzerfehler vorwegzunehmen und damit zu vermeiden.

Aus der naiven Physik entwickelte sich die qualitative Simulation (z. B. QSIM [50]). Hier geht es darum, mit nicht vollständigem Wissen über physikalische Systeme zu schlußfolgern. Es wird ein qualitatives Modell des physikalischen Systems entwickelt, in dem dann bestimmte Vorgänge simuliert werden können (z. B. eine Uhr, siehe CLOCK [25]). Die qualitative Simulation wird heute vorwiegend in Diagnosesystemen eingesetzt, wo sich der regelbasierte Expertensystem-Ansatz als unpraktikabel erwiesen hat. Ein qualitatives Modell eines physikalischen Systems erlaubt es, mögliche Ursachen von Störungen im realen System aufzufinden.

2.2.2 Intervallkalküle

Etwas abstraktere Ansätze beschäftigen sich nicht mit ganzen physikalischen Systemen, sondern zunächst mit einzelnen Domänen des Raumes bzw. der Zeit. So entstanden verschiedene Intervallkalküle, die einerseits für das Schlußfolgern über Intervalle auf der Zeitachse benutzt werden können, aber auch für Schlußfolgern über einzelne Dimensionen von räumlichen Entitäten, die auf die einzelnen Raumachsen projiziert werden.

Diese Kalküle gehen zumeist auf Allen [1] zurück, der 13 Relationen zwischen zeitlichen Intervallen definiert und das Schlußfolgern mittels Kompositionstabellen² einführt: Wenn Intervall A in Relation r_1 zu B steht, und Intervall B in Relation r_2 zu C, in welcher Relation stehen dann A und C zueinander? Dies kann durch ein simples Nachschlagen in der Tabelle entschieden werden, wobei das Resultat sehr oft eine Disjunktion von möglichen Relationen ist, wenn das Ergebnis nicht eindeutig ist. Diese Kompositionstabelle ermöglicht das Propagieren von Constraints und so das Schließen mit diesen Intervallen, z.B. für den Einsatz in einem Planungssystem [2]. Diese Idee wurde oft aufgegriffen und weiterentwickelt, z.B. von Ligozat mit den generalisierten Intervallen [52, 53]. In [54] werden diese Relationen erstmals gemäß ihrer Nachbarschaftsverhältnisse in einer polygonalen Struktur angeordnet.

Diese Idee wird auch von Freksa [29] verfolgt: Wo Allen für vages Wissen mit Disjunktionen von Relationen arbeiten muß, also umso mehr spezifizieren muß, je weniger er über die Beziehungen zwischen zwei Intervallen weiß, nutzt Freksa den Umstand aus, daß es zwischen den Intervallrelationen konzeptuelle Nachbarschaften gibt. D.h., wenn zwei Intervalle beispielsweise in der Relation a zueinander stehen, und etwas ändert sich (ein Intervall bewegt sich, oder wächst), ist es nicht möglich, in die Relation d zu gelangen, ohne b und c zuerst zu durchlaufen. Es hat sich herausgestellt, daß die Disjunktionen, die man in Allens Kompositionstabelle erhält, ausnahmslos aus vollständigen konzeptuellen Nachbarschaften bestehen. Schlußfolgern mit Nachbarschaften ermöglicht nun „billigere“ Berechnung unsicheren Wissens und erlaubt außerdem, Probleme zunächst auf einer größeren Granularität zu lösen als beim Schlußfolgern mit Disjunktionen von Basisrelationen. Die Idee der konzeptuellen Nachbarschaft von Relationen beeinflusste weitere Arbeiten im qualitativen räumlichen Schließen, siehe z.B. [42].

2.2.3 Mereotopologische Kalküle

In der Tradition der Kalküle mit Relationen zwischen Intervallen stehen Kalküle mit Relationen zwischen räumlichen Regionen. Dabei werden oft Relationen verwendet, die in topologischen Räumen invariant sind. Diese Theorien gehen auf Clarke [11] zurück. In [73] werden diese Relationen axiomatisch definiert, und zwar basierend auf einer primitiven

²Allen verwendet das Wort „Transitivitätstabellen“; mittlerweile hat sich allerdings „Kompositionstabellen“ eingebürgert.

zweistelligen Relation $C(x, y)$: Wenn x, y räumliche Regionen sind, so gilt $C(x, y)$ genau dann, wenn sie mindestens einen Punkt gemeinsam haben. Mit Hilfe dieser Relation, die mittels zweier Axiome als reflexiv und symmetrisch definiert wird, werden verschiedene weitere Relationen definiert, von denen 8 paarweise disjunkt und exhaustiv sind: die RCC-8-Relationen; siehe Abbildung 2.1.

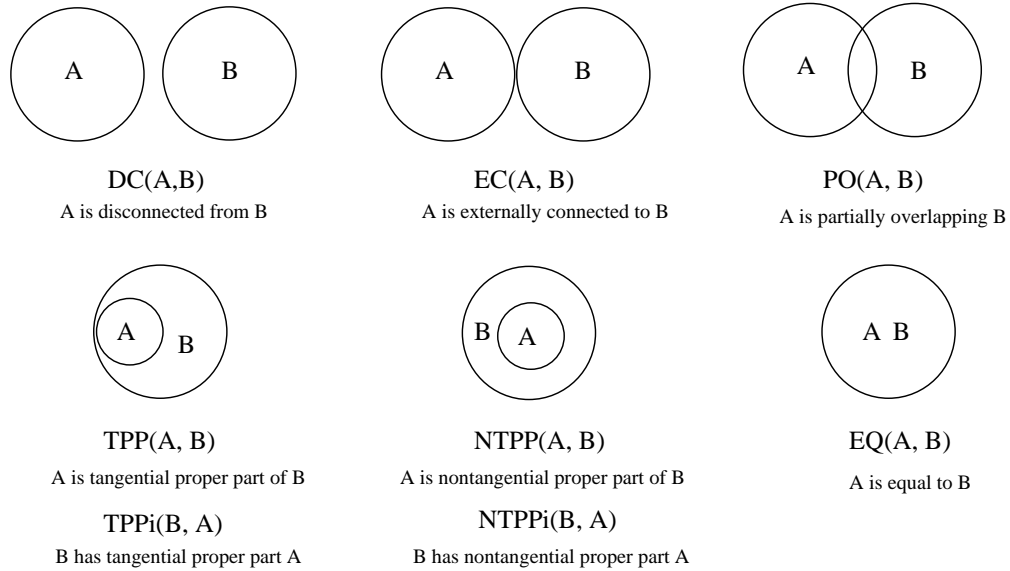


Abbildung 2.1: Die RCC-8-Relationen

In [4] wird ein ähnlicher Ansatz verfolgt, und überhaupt wurde in den letzten Jahren eine Reihe von ganz ähnlichen Kalkülen entwickelt, die sich oft nur in der Art unterscheiden, wie die basale Relation $C(x, y)$ definiert ist und wie Randpunkte behandelt werden. Eine systematische Einordnung und Übersicht dieser Kalküle wird in [15, 14] gegeben.

Aus der Geoinformatik kommt ein Ansatz [20], der Relationen zwischen räumlichen Regionen systematisch aus der reinen Topologie³ herleitet. Welche Relationen zwei Mengen in einem topologischen Raum zueinander einnehmen können, hängt von den Gegebenheiten zwischen den jeweiligen topologischen Entitäten Rand und Inneres dieser beiden Mengen ab. Es werden alle möglichen Schnittmengen, die sich daraus ergeben können, betrachtet: Rand/Rand, Rand/Inneres, Inneres/Rand, Inneres/Inneres. Jede dieser vier Schnittmengen (die dem Kalkül auch den Namen 4-intersection gaben) kann entweder leer oder nicht leer sein, womit sich insgesamt 16 mögliche Relationen zwischen zwei Mengen im topologischen Raum ergeben. Nun ist nicht jede Menge in jedem topologischen Raum sinnvollerweise als Region im geographischen Raum zu betrachten. Räumliche Re-

³Topologie im mathematischen Sinn

gionen werden deswegen folgendermaßen eingeschränkt definiert: Der topologische Raum X muß zusammenhängend sein. Eine räumliche Region A ist eine nichtleere echte Untermenge von X , wobei das Innere von A zusammenhängend ist und A identisch ist mit dem Abschluß des Inneren von A . Diese räumlichen Regionen sind homöomorph zu polygonalen Regionen im \mathbb{R}^2 . Lediglich 9 der 16 möglichen Relationen können zwischen zwei derart eingeschränkten Mengen gelten. Sie sind identisch zu den Relationen des RCC-8, wobei die neunte lediglich ein Spezialfall von *Overlap* ist, bei dem sich die Ränder der Regionen nicht schneiden und der im \mathbb{R}^2 nur realisierbar ist, wenn eine der Regionen ein Loch hat. Neuere Ansätze aus dieser Richtung betrachten nicht nur Inneres und Rand der Mengen, sondern auch deren Komplement, wodurch es möglich wird, die Mengen auch mit dem umgebenden Raum in Beziehung zu setzen. Daraus resultieren dann neun Schnittmengen (9-intersection) [18, 19] und damit $2^9 = 512$ mögliche Relationen, von denen sich allerdings wieder die meisten nicht mit räumlichen Regionen realisieren lassen.

2.2.4 Kalküle mit reicherer Raumstruktur

Repräsentationen, die rein topologische Relationen benutzen, haben den Nachteil, relativ strukturarm zu sein. So kann man nicht nur Aussagen wie „ X ist nahe bei Y “, was zumindest eine rudimentäre Repräsentation von Distanzen voraussetzt, dort nicht ausdrücken. Es ist auch nicht möglich, links und rechts zu unterscheiden. Dies würde die Einführung von Orientierungsrelationen voraussetzen. Freksa [30, 33] führt relative Orientierungen eines Punktes zu zwei anderen Punkten (z.B. Start- und Endpunkt eines Bewegungssegments) in seinem Doppelkreuzkalkül ein. Dazu wird ein „orientation grid“ benutzt, das durch einen Vektor definiert ist, der durch zwei Punkte a und b geht. Dieser Vektor wird nach vorn und hinten verlängert, und durch a und b wird je ein Lot gefällt: So erhält man das Doppelkreuz, das den Raum in 15 verschiedene qualitative Positionen unterteilt (siehe Abbildung 2.2).

Verschiedene Operationen sind auf diese Art der Repräsentation anwendbar. Seien ein Punkt c und ein Doppelkreuz, definiert durch a und b , gegeben. c befindet sich dann in einer der 15 Relationen zu a und b , geschrieben $c : ab$.

1. INV bezeichnet die Inversion des Vektors ab . Dies entspricht folgender Frage: Auf dem Weg von a nach b sah man eine Landmarke auf relativer Position c ; auf welcher relativen Position befindet sie sich auf dem Rückweg? Dies ist die einzige Operation mit exaktem Resultat, d.h. es kommt immer eine einzelne Relation heraus und nicht eine Disjunktion. Es gilt: $\text{INV}(\text{INV}(z : xy)) = z : xy$.
2. HM bezeichnet „homing“. Dies entspricht der Frage, wo sich der Startpunkt a befindet, wenn man seinen Weg von b nach c fortsetzt. Gegeben ist also $c : ab$, gesucht ist $a : bc$. Dies ergibt meist eine Disjunktion von möglichen Relationen, die aber alle konzeptuelle Nachbarschaften sind.

3. SC bezeichnet den „shortcut“. Gegeben $c : ab$, gesucht $b : ac$. Diese Operation ergibt alle Relationen, die auch schon im Ergebnis von HM vorkamen, nur in anderer Reihenfolge; hier besteht eine große konzeptuelle Ähnlichkeit.

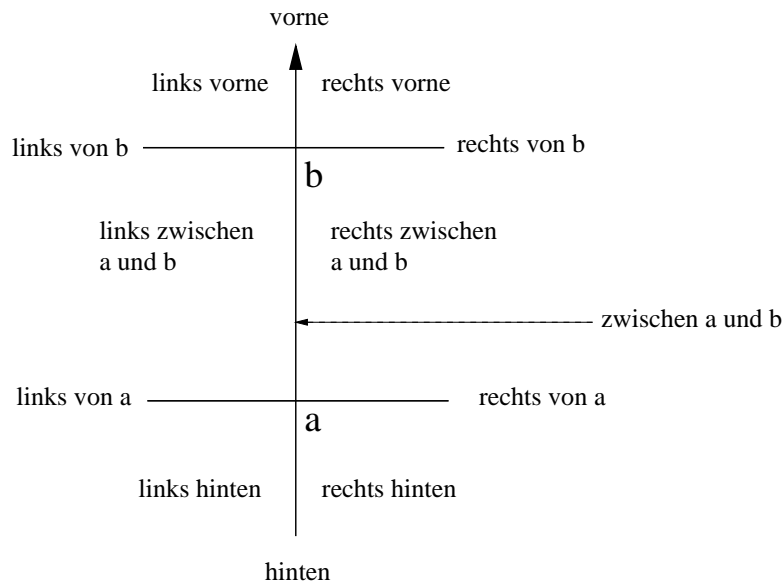


Abbildung 2.2: Freksasches Doppelkreuz

Wenn zwei Doppelkreuzrelationen vorhanden sind, kann man sie natürlich auch kombinieren. Dies entspricht der Operation, die Menschen durchführen, wenn sie auf einem (verschlungenen) Weg deduzieren, wo sich eine (verdeckte) Landmarke, die auf einem früheren Abschnitt des Weges sichtbar war, nun befindet. Etwas formaler ausgedrückt: Aus den Relationen $c : ab$ und $d : bc$ soll $d : ab$ deduziert werden. Auch diese Operation hat normalerweise eine Disjunktion von Relationen zum Ergebnis.

Mit den Operatoren INV, HM, SC und der Komposition kann man eine (Disjunktion von) Relation(en) für jede mögliche Kombination von Punkt/Referenzlinie deduzieren. Damit ist es möglich, die relative Position von Landmarken an jedem Punkt eines Weges zu deduzieren. Allerdings ist das Ergebnis recht ungenau, weil Distanzinformationen fehlen. So kommt oft eine größere Disjunktion von Richtungsrelationen, als mit Distanzinformationen nötig wäre, heraus.

Schlieder [81] versucht, den Raum strukturierter als durch bloße Topologie zu repräsentieren, ohne auf Orientierung zurückzugreifen. Stattdessen stützt er sich auf Ordnungsrelationen, die implizit auch schon im Freksaschen Ansatz enthalten sind: Ordnung und Orientierung hängen eng zusammen, z.B. induziert die Orientierung einer Linie, wie im Doppelkreuzkalkül, eine Orientierung von Segmenten auf dieser Linie, gegeben

durch die Ordnung der Endpunkte. Ein Segment $p_i p_j$ ist positiv (negativ) orientiert, wenn der Pfad von p_i nach p_j der positiven (negativen) Orientierung der Linie folgt: $[p_i p_j] = + \rightarrow p_i < p_j$.

Im Zweidimensionalen muß man eine Orientierung der Ebene einführen. Dann kann man ähnlich wie im Freksaschen Doppelkreuzkalkül die Orientierung von je drei Punkten zueinander feststellen: Die Orientierung eines Tripels $p_i p_j p_k$ kann positiv, negativ, oder 0 sein, je nachdem, ob die Anordnung der Punkte in der Ebene, wenn sie wie im Tripel angegeben durchlaufen werden, der positiven oder negativen Orientierung der Ebene folgt, oder, wie im letzten Fall, auf einer Geraden liegt.

Mit Hilfe dieser orientierten Tripel kann man nun einen Kalkül für Strecken entwickeln: Die Relation, in der zwei Strecken zueinander stehen, wird beschrieben durch alle orientierten Tripel ihrer Endpunkte. Daraus ergeben sich 14 Relationen, die natürlich wieder in einer konzeptuellen Nachbarschaft angeordnet werden können. Mit diesem Kalkül kann z. B. Bewegungsplanung durchgeführt werden, denn Bewegungen, die zu einer Kollision führen, können mit seiner Hilfe erkannt und ausgeschlossen werden.

Noch mehr Struktur bringt ein Ansatz von Hernández in den qualitativen Raum, der die qualitative Beschreibung mehrerer Domänen des Raums kombiniert, nämlich zunächst Orientierung und Topologie [42], und dann Orientierung und Distanz [12], womit qualitative Positionsangaben von zwei Objekten (Primärobjekt und Referenzobjekt) zueinander möglich werden. Die letztgenannte Art der Repräsentation beruht auf einer Diskretisierung dieser beiden Domänen in Intervalle, die bereits erwähnten Quantity Spaces. Während die Orientierungsdomäne von Natur aus begrenzt ist (nach 360° ist man wieder an der gleichen relativen Orientierung angelangt), und eine Diskretisierung dieser Domäne in eine endliche Zahl gleich oder verschieden großer Richtungsbereiche sich relativ natürlich ergibt (siehe dazu auch Kapitel 4), ist die Distanzdomäne potentiell unendlich. Wieviele Distanzintervalle man unterscheiden möchte und wie groß sie jeweils sein sollen, hängt sehr stark vom Kontext ab. In [12] werden die verschiedenen Möglichkeiten ausführlich diskutiert. Ferner wird hier auch auf das Problem der unterschiedlichen Referenzsysteme eingegangen. Für die Repräsentation einer qualitativen Position ist die Angabe von separaten Referenzsystemen für Orientierungen und Distanzen notwendig.

Die oben beschriebenen qualitativen Kalküle werden neben den rein ontologischen Modellierungsaufgaben natürlich auch zum Schließen benutzt. Dies geschieht vorwiegend mittels Constraintpropagierung, Constraint Erfüllung und Finden eines kanonischen Modells oder aller Modelle für ein gegebenes Constraintnetz. Wie in der Einleitung schon erwähnt, haben diese Algorithmen den Nachteil der NP-Vollständigkeit. Trotzdem wurden in den letzten Jahren einige Anstrengungen darauf verwendet, handhabbare Untermengen dieser Kalküle zu finden, z. B. für den Allenschen Intervallkalkül [65, 64] und RCC-8 [77, 76]. Für diese handhabbaren Untermengen ist der Nachweis der Pfadkonsistenz im Constraintnetz äquivalent zum Konsistenznachweis und kann in $O(n^3)$ (n ist die Anzahl der Knoten im Constraintnetz) entschieden werden. In [75] wird beschrieben,

wie man für ein gegebenes konsistentes RCC-8-Constraintnetz ein kanonisches Modell mit zusammenhängenden Regionen in Räumen mit einer Dimension ≥ 3 erzeugen kann. Diese und ähnliche Bemühungen haben dazu geführt, daß Constraintenerfüllung z. B. bei schwierigen Scheduling-Aufgaben teilweise effizienter ist als konventionelle Methoden, siehe auch [72]. Während allerdings temporale Relationen im Scheduling eine praktische Anwendung finden, sind die Anwendungen für qualitative räumliche Kalküle noch eher selten. Beispiele sind Layoutmanager und Dokumentenanalyse, die aber häufig mit Projektionen der Regionen auf die beiden Raumachsen und dort wieder mit Intervallrelationen arbeiten.

2.3 Bewegungsverläufe

Die Newtonsche Mechanik und die Relativitätstheorie als physikalische Theorien, die Raum und Bewegung beschreiben, wurden bereits erwähnt. Bewegungen werden in diesen Theorien mit Hilfe von Gleichungssystemen beschrieben, wobei Eigenschaften der sich bewegenden Körper wie Massen und ausgeübte Kräfte eine wichtige Rolle spielen. Die Beschreibung der Planetenbewegungen war bereits bei der Entwicklung der Newtonschen Mechanik die wichtigste Anwendung.

Die Mathematik kennt zur vollständigen Beschreibung eines Bewegungsverlaufs unabhängig von Massen und ausgeübten Kräften (den Ursachen für und Einflüssen auf die Bewegung), also zur rein deskriptiven Beschreibung, das Konzept der Trajektorie, die nicht zu verwechseln ist mit der Spur eines Bewegungsverlaufs. Eine Trajektorie ist eine differenzierbare Abbildung $\alpha : I \rightarrow \mathbb{R}^n$ eines offenen Intervalls $I = (a, b)$ der reellen Geraden \mathbb{R} in den \mathbb{R}^n , während ihre Spur $\alpha(I) \subset \mathbb{R}^n$ eine Menge von Punkten ist (vgl. [10]).

Wenn eine Bewegung einer Menge von Punkten beobachtet wird, z. B. eine Planetenbewegung, oder die Bewegung eines Rades, kann man jedem Punkt q aus dieser Menge einen Bewegungsvektor $w(q)$ zuordnen. Ist dieses Vektorfeld differenzierbar, so kann man ihm eindeutig eine Trajektorie zuordnen, die diese Bewegung beschreibt, d. h. es gibt eine Trajektorie $\alpha(t)$, $t \in I$ (I offenes Intervall in \mathbb{R}), so daß $\alpha'(t) = w(\alpha(t))$ ist. Das bedeutet, ein differenzierbares Vektorfeld w bestimmt ein System von Differentialgleichungen, dessen Lösung die Trajektorie von w ist.

Die Differentialgeometrie liefert die Werkzeuge, um lokale (Geschwindigkeit, Krümmung, Torsion, ...) und globale (Bogenlänge, Orientierung, geschlossen vs. offen, selbstüberschneidend, ...) Eigenschaften von Trajektorien zu analysieren. Die lokalen Eigenschaften sind infinitesimal und damit ungeeignet, die Gestalt einer Trajektorie wiederzugeben, wohingegen die globalen Maße dazu zu grob sind. Um den Gestalteindruck wiederzugeben, der bei der Beobachtung eines Bewegungsverlaufs entsteht, müssen andere Mittel der Repräsentation gefunden werden.

Auch in der psychologischen Forschung klafft eine erstaunliche Lücke zwischen der Erforschung von Wahrnehmung und Verarbeitung solch infinitesimaler, lokal-instanter Bewegungsmerkmale und der Wahrnehmung, Speicherung und Verarbeitung längerandauernder und komplexer Stimuli. Dies ist mit ein Grund für die Durchführung des Forschungsprojekts, aus dem diese Arbeit entstanden ist. Ein ausführlicher Überblick über den Stand der psychologischen Forschung wird in [21] gegeben.

Die ältesten und explizitesten Modelle der Bewegungsverarbeitung sind die Berechnungsmodelle, allen voran der Reichardt-Detektor [74], auf den im Prinzip alle lokalen visuellen Bewegungsdetektoren aufbauen, sofern sie nicht äquivalent dazu sind. Lokale Bewegungsdetektoren haben im einfachsten Fall zwei Eingänge an zwei direkt nebeneinandergelegenen Stellen, z. B. zwei benachbarten Facetten eines Insektenauges. Das Signal eines Eingangs wird eine kurze Zeit verzögert und anschließend mit dem Signal am anderen Eingang verglichen. Liegt das gleiche Signal an, wird dies als lokal-instantanes, gerichtetes Bewegungssignal interpretiert. Der Übergang von solchen lokalen Bewegungsdetektoren zu höheren Stufen in der Bewegungsverarbeitung, z. B. der Wahrnehmung einer Bewegungsgestalt, ist allerdings immer noch unklar. Daß Menschen eine solche Bewegungsgestalt wahrnehmen, also ganz offenbar die lokal-instantanen Merkmale über die Zeit *und* den Raum integrieren können, geht z. B. aus den Forschungen Johanssons hervor [43, 44, 45].

Ergebnisse innerhalb des Projekts, aus dem diese Arbeit hervorgegangen ist, deuten ebenfalls darauf hin, daß die Form von Teilen eines Bewegungsverlaufs die Wahrnehmungs- und Gedächtnisleistung beeinflusst und daß Menschen Bewegungsverläufe möglicherweise ganz allgemein nach Gesichtspunkten der Form strukturieren.

Abbildung 2.3 zeigt einige der Ergebnisse eines Experiments zu der Frage, wie gut Menschen am Bildschirm beobachtete Bewegungsverläufe reproduzieren können. Die Versuchspersonen (Vpn) sahen einen Ziel-Bewegungsverlauf am Bildschirm (erste Reihe). Unmittelbar nach dem Ende der Darbietung ertönte ein akustisches Signal, und die Vpn mußten den Bewegungsverlauf reproduzieren, indem sie mit dem Finger auf dem Touch Screen seine Spur nachzeichneten. Dargestellt ist die Reproduktionsleistung von 5 Vpn (Reihen) für 6 Ziel-Bewegungsverläufe (Spalten). Eine Diskussion des Experiments findet sich in [63].

Ganz allgemein scheint es in diesem Experiment, daß manche Spuren besser reproduziert werden können als andere. Es scheint, daß in manchen Spuren Muster vorkommen, die klassifiziert und so besser erinnert werden können (wie z.B. die eckigen und runden Schleifen), wohingegen Spuren, die solche Muster nicht aufweisen, wie z.B. P13 oder P9, schlechter reproduziert werden, ja fast übergeneralisiert scheinen. Muster mit spitzen Ecken wie in P15 und P18 können anscheinend mit am besten reproduziert werden.

Der Formaspekt von Bewegungsverläufen scheint also nicht nur bei der Wahrnehmung von Punktwolken, die sofort strukturiert werden, eine Rolle zu spielen, sondern auch bei

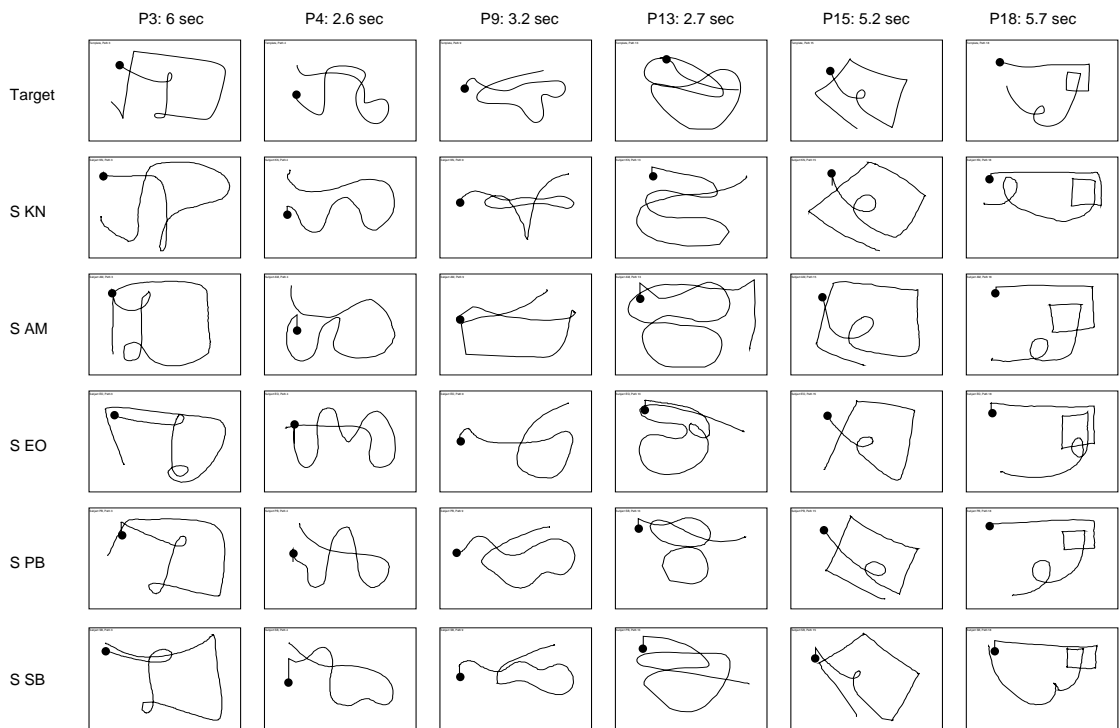


Abbildung 2.3: Experiment zur Reproduktion von Trajektorien. Abbildung aus [63]

der Wahrnehmung einzelner bewegter Punkte. Die Form des Bewegungsverlaufs scheint demnach geeignet, bei einer qualitativen, für den Menschen intuitiven Repräsentation von Bewegungsverläufen, eine entscheidende Rolle zu spielen.

2.4 Qualitative Repräsentation von Bewegungsverläufen

Was für qualitative Repräsentationen des Raumes gilt, gilt natürlich auch für die qualitative Repräsentation von Bewegungsverläufen: Systeme, die mit Bewegung im Raum zu tun haben und die über eine intuitive Benutzungsschnittstelle verfügen sollen, müssen sie verstehen. Dies gilt für Systeme, die sich selbst im Raum bewegen wie ein Service-roboter oder ein autonomer Rollstuhl („fahr langsamer“, „bieg links ab“) ebenso wie für Systeme, die Bewegungsverläufe wahrnehmen wie Verkehrsüberwachungssysteme, oder speichern wie Multimediadatenbanken („Gib mir alle Bildfolgen zurück, auf denen ein

Überholvorgang zu sehen ist“).

Die Literatur zur qualitativen Repräsentation von Bewegungsverläufen ist nicht ganz so reichhaltig wie die zu statischen räumlichen Repräsentationen. Ein kurzer Überblick wird im Folgenden gegeben.

2.4.1 Qualitative Kinematik

Kinematik ist die Theorie von den möglichen Bewegungen von Körpern, wobei die Kräfte, die in solche Bewegungen involviert sind, außer Acht gelassen werden; Kinematik ist also mehr mit der Geometrie der Bewegung befaßt als mit ihrer Dynamik.

Wie bei den qualitativen räumlichen Repräsentationen liegen die Anfänge der qualitativen Repräsentation von Bewegung in der qualitativen Physik. Die ersten Arbeiten zur qualitativen Kinematik waren stark von den Arbeiten der qualitativen Physik beeinflusst. Wie dort wollte man die klassischen Beschreibungen mittels Gleichungssystemen durch eine Repräsentation ersetzen, die auf Transitionen in diskreten Zustandsräumen basiert. In [24] wird eine qualitative Kinematik vorgestellt, mit der die Bewegungen eines hüpfenden Balls modelliert werden kann. Es werden dabei nur Bewegungen durch den freien Raum und instantane Kollisionen mit Oberflächen berücksichtigt. Ganz in der Tradition der qualitativen Physik wird versucht, dies als Zustandsübergangsdiagramm zu modellieren: Ein „place vocabulary“ (PV) definiert Teilgebiete des Raums, die bezüglich einer bestimmten Eigenschaft homogen sind. Dieses „place vocabulary“ ist allerdings eingebettet in eine eher quantitative, analoge Repräsentation, das „metric diagram“ (MD). Sie repräsentiert die Geometrie, die numerische und symbolische Information kombiniert. In einer späteren Publikation zur qualitativen Kinematik [23] behaupten Forbus und Co-Autoren sogar, eine rein qualitative universelle Kinematik sei nicht möglich, da ohne metrische Informationen wie die relative Größe von Objekten oder relative Positionen der Teile einer inhomogenen Oberfläche bestimmte Bewegungsprobleme gar nicht gelöst werden können. Diese Behauptung wurde weithin bekannt unter der Bezeichnung „poverty conjecture“, da sie behauptet, qualitative Repräsentationen können nicht reich genug sein, um Raum universell zu repräsentieren. Dies ist möglicherweise sogar richtig, wenn man lediglich die Idee der Transitionsdiagramme aus der qualitativen Physik zur Raumrepräsentation zuläßt. Diese Art der Repräsentation ist nützlich und erfolgreich, wenn es darum geht, Bewegungen in mechanischen Systemen zu modellieren, bei denen der Freiheitsgrad der einzelnen Teile stark eingeschränkt ist – z.B. in Getrieben. Diese Art der Mechanismen wird in [22] „kinematische Kette“ genannt und dort mit dem bereits erwähnten „place vocabulary“ modelliert. Die „places“, also Zustände, befinden sich hier allerdings nicht im freien 2- oder 3D-Raum, sondern im Konfigurationsraum des Mechanismus. Eine andere Hypothese aus [23] erfreute sich in der QRS-Gemeinde größerer Zustimmung als die „poverty conjecture“, nämlich die „connectivity hypothesis“. Sie formuliert die Idee, daß Connectivity, also Zusammenhang, die Schlüsselprimitive

für qualitative Beschreibungen von räumlichen Konfigurationen ist. Zusammenhang als primitive Relation ist ja die Basis für die mereotopologischen Kalküle, die in Abschnitt 2.2.3 vorgestellt wurden. Eine Publikation von Gotts mit dem Titel „How far can we C“ untersucht, welche repräsentationale Ausdruckskraft mit dieser Primitive erreicht werden kann [38].

2.4.2 Axiomatische Ansätze

Auch auf dem Gebiet der Bewegung gibt es natürlich Ansätze, die nicht gleich alle Aspekte für einen gewissen Fall der Bewegung modellieren wollen, sondern sich auf gewisse Aspekte der Bewegung und/oder reduzierte Raumrepräsentationen wie die topologische stützen.

Galton [35] stützt seine qualitative Theorie der Bewegung auf eine Theorie der Zeit, die Augenblicke (instants) als grundlegende Menge T von temporalen loci benutzt. T ist ausgestattet mit der Relation $<$ (zeitliche Aufeinanderfolge). Ein zeitliches Intervall ist definiert durch die beiden Augenblicke, an denen es beginnt und endet, also ein geordnetes Paar von Augenblicken $\langle t, u \rangle$ mit $t < u$. Galton definiert drei Relationen der zeitlichen Aufeinanderfolge, von denen 'meets' ($|$) die wichtigste ist: $\langle t, u \rangle | \langle v, w \rangle$ gdw $u = v$. Als räumliche Relationen werden die RCC-8-Relationen verwendet.

Das fundamentale Konzept in Galtons Analyse von raumzeitlicher Veränderung ist der „Fluent“, eine Entität, die zu unterschiedlichen Zeiten unterschiedliche Werte annehmen, also sich verändern kann. Wenn f ein Fluent ist und a ein Wert, dann ist $f = a$ ein boolescher Fluent oder *Zustand*. Über Zustände und Zeitpunkte bzw. Intervalle können nun Prädikate definiert werden wie Holds-at(S, t) (ein Zustand ist gültig für einen bestimmten Zeitpunkt t) oder Holds(S, i) (ein Zustand ist gültig in einem Intervall i).

pos(a) bezeichnet die Position eines Körpers a . Da sich die Position über die Zeit ändern kann, ist sie ein Fluent. Eine Bewegung ist ein Ereignis, das passiert, kein Zustand, deswegen wird ein weiteres Prädikat Occurs(e, i) eingeführt, das bedeutet, daß ein Ereignis des Typs e über ein zeitliches Intervall i hin passiert.

Bewegung wird nun mit Hilfe der Bedingungen ihres Auftretens analysiert; z.B. die Bewegung eines Körpers a von Position r_1 nach Position r_2 während eines Intervalls i . Dies kann folgendermaßen definiert werden:

$$\begin{aligned} \text{Occurs}(\text{move}(a, r_1, r_2), i) =_{def} \\ & \text{Holds-at}(\text{pos}(a) = r_1, \text{Beg}(i)) \wedge \text{Holds-at}(\text{pos}(a) = r_2, \text{End}(i)) \wedge \\ & \text{Holds}(\text{pos}(a) \neq r_1, i) \wedge \text{Holds}(\text{pos}(a) \neq r_2, i). \end{aligned}$$

Die Behandlung der Kontinuität ist für jede Theorie der Bewegung ein wichtiges Thema, wofür Galton, aufbauend auf seiner Theorie der Zustände, Dominanzräume über Zuständen vorschlägt:

Seien S' und S Zustände. S' dominiert S ($S' \succ S$), wenn für diese beiden Zustände

mit die folgende Situation vorkommen kann: $\text{Holds}(S, i) \wedge \text{Lim}(t, i) \wedge \text{Holds-at}(S', t)$ (das heißt: S gilt im Intervall i , t ist ein Endpunkt des Intervalls i , und S' gilt am Moment t , der das Intervall begrenzt).

S' wird von S dominiert ($S' \prec S$), wenn nur $\text{Holds}(S', i) \wedge \text{Lim}(t, i) \wedge \text{Holds-at}(S, t)$ vorkommen kann.

Mit Hilfe dieser Dominanzräume können z.B. die konzeptuellen Nachbarschaften von RCC-8 rekonstruiert werden. Diese konzeptuellen Nachbarschaften definieren sich wie in der Arbeit von Freksa [29] dadurch, was passiert, wenn sich bei zwei Regionen, die in einer bestimmten RCC-8-Relation zueinander stehen, etwas ändert; z.B. eine Region sich bewegt, oder wächst, etc. Bei diesem Prozeß kann sich die Relation, in der die beiden Regionen zueinander stehen, nicht beliebig ändern, sondern nur immer in einen konzeptuellen Nachbarn umgewandelt werden. Die Abbildung 2.4 gibt den Dominanzraum für den RCC-8-Kalkül wieder.

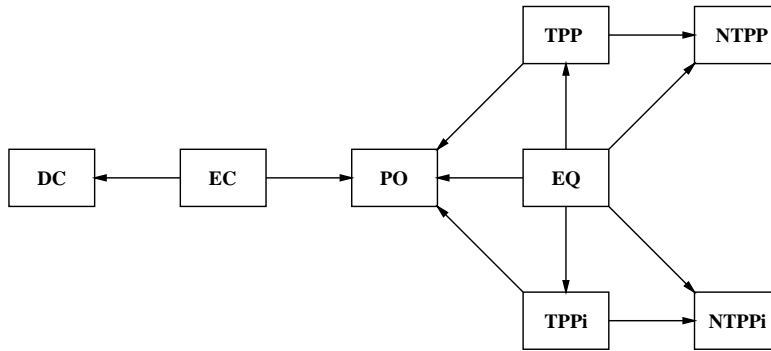


Abbildung 2.4: Dominanzraum für RCC-8

In [36] widmet sich Galton nochmals ausführlich dem Problem der kontinuierlichen Veränderung bei räumlichen Regionen. Dort greift er das ε - δ -Paradigma zur Definition kontinuierlicher Funktionen auf den reellen Zahlen aus der Mathematik auf. Damit untersucht er graduelle Unterschiede in der Kontinuität einer räumlichen Veränderung und definiert verschiedene Formen der Kontinuität, basierend auf verschiedenen Maßen der Separiertheit von räumlichen Regionen: Die *Boundary-Separation* zwischen zwei Regionen R_1 und R_2 ($R_1 \nabla_{\partial} R_2$) ist definiert als die kleinste Zahl d , so daß jeder Punkt im Rand ∂R_1 von R_1 sich innerhalb einer Distanz d zu einem beliebigen Punkt zum Rand ∂R_2 von R_2 befindet. Die *Size-Separation* zwischen zwei Regionen R_1 und R_2 ($R_1 \nabla_{size} R_2$) ist definiert als die Fläche (in zwei Dimensionen) oder das Volumen (in drei Dimensionen) der symmetrischen Differenz von R_1 und R_2 ($(R_1 \setminus R_2) \cup (R_2 \setminus R_1)$). Die *Interior-Separation* zwischen zwei Regionen R_1 und R_2 ($R_1 \nabla_{int} R_2$) ist definiert als die kleinste Zahl d , so daß jeder Punkt in jeder Region innerhalb einer Distanz d zu einem beliebigen Punkt

in der anderen Region liegt. *Boundary-Separation* und *Interior-Separation* drücken den Grad Getrenntheit der Regionen als Distanz aus, wogegen *Size-Separation* dies als Fläche oder Volumen beschreibt.

Aufbauend auf diesen drei Maßen der Separiertheit können nun unterschiedliche Arten der Kontinuität definiert werden, wobei Galton das ε - δ -Paradigma zu Hilfe nimmt:

Die räumliche „Geschichte“ einer Entität E ist \star -kontinuierlich (für $\star \in \{\partial, size, int\}$) über ein Zeitintervall (t_1, t_2) , wenn

1. $pos(E, t)$ definiert ist für alle $t \in (t_1, t_2)$ und
2. für jedes $\varepsilon > 0$ ein $\delta > 0$ existiert, so daß für alle $t, t' \in (t_1, t_2)$ gilt: wenn $|t - t'| < \delta$, dann $pos(E, t) \nabla_{\star} pos(E, t') < \varepsilon$.

Mit Hilfe dieser verschiedenen Arten der Kontinuität können nun Ereignisse räumlicher Veränderung untersucht werden. So kann z.B. die allmähliche Ablösung einer Halbinsel vom Festland, bei der sich eine Wasserstraße wie ein Keil durch die Landzunge treibt, als ∂ -, *size*- und *int*-kontinuierlich betrachtet werden. Dagegen ist der Fall, daß sich auf einen Schlag zwischen zwei Zeitpunkten t_0 und t_1 ein schmaler Kanal über die ganze Breite der Region bildet, der immer breiter wird, *size*- und *int*-kontinuierlich, aber nicht ∂ -kontinuierlich, da für jedes $\delta > 0$ zum Zeitpunkt $t_0 + \delta$ die *Boundary-Separation* der ursprünglichen Region von der neuen Konstellation mit den zwei Regionen mindestens die halbe Breite der ursprünglichen Region beträgt (wenn der Kanal sich genau in der Mitte dieser Region gebildet hat). Dies stimmt mit der Intuition überein, daß eine Veränderung wie im ersten Fall irgendwie kontinuierlicher ist als die im zweiten Fall. Dies ist eine Intuition, die mit Hilfe der verschiedenen Kontinuitätsbegriffe ausgedrückt werden kann.

Muller [59, 58] stützt sich in seiner qualitativen Theorie der Bewegung auf die mereotopologische Theorie aus [4]. Allerdings werden hier Entitäten und Relationen nicht als rein räumlich, sondern spatiotemporal interpretiert. Die primitive Relation, auf der aufbauend die anderen definiert werden, ist hier wieder $C(x, y)$. Wegen des spatiotemporalen Charakters der Theorie muß allerdings noch eine zeitliche Primitive eingeführt werden, die die Ordnungsstruktur der Zeitdomäne wiedergibt, nämlich „temporal connection.“ Ausgehend von diesen beiden Basisrelationen wird axiomatisch ein spatiotemporaler Kalkül definiert. Durch die Interpretation der n -dimensionalen Entitäten als $n - 1$ -dimensionale räumliche Objekte mit einer Ausdehnung in der Zeit entsteht ein wirklich integrierter Kalkül der Bewegung, der sich nicht auf eine Domäne stützt und die andere lediglich von außen „anklebt“.

Mit Hilfe des Kalküls können kontinuierliche spatiotemporale Objekte definiert werden, denn auch diese Theorie befaßt sich mit der Frage der Kontinuität von Bewegung. Damit lassen sich wie schon bei Galton die konzeptuellen Nachbarschaften im RCC-8-Kalkül rekonstruieren, oder aber Bewegungsklassen identifizieren, mit denen man z.B. Bewe-

gungsverbren eine formale Semantik geben kann. Auch logisches Schließen über den so definierten Konzepten ist mit dem Kalkül möglich.

Im Gegensatz zu den oben referierten, eher ontologisch ausgelegten Ansätzen sind die Arbeiten von Del Bimbo und Kollegen [7, 8] von einer praktischen Anwendung her motiviert, nämlich dem Problem des Retrieval in Multimediadatenbanken. Zu seiner Lösung wurde eine Sprache zur Repräsentation des Inhalts von Bildsequenzen entwickelt, die auf der temporalen (Modal-) Logik basiert und deswegen Spatio-Temporale Logik (STL) genannt wird.

Die Spatiale Logik, also der Teil der Spatio-Temporalen Logik, der mit statischen Beschreibungen zu tun hat, benutzt das bekannte Konzept, die 2-D-Information aus Bildern (bzw. Frames in Bildfolgen) auf die jeweiligen Achsen zu projizieren, auf denen dann mit der Temporallogik operiert werden kann.

Sei p der Name eines Objekts in der Szene, S ein Frame in einer Bildfolge, r ein Punkt in der Szene, und E_i eine Achse des Koordinatensystems. Die grundlegende Frame-Aussage

$$(S, r, E_i) \models p,$$

bedeutet, daß es einen Punkt in p gibt, dessen Projektion auf die Achse E_i mit der Projektion von r auf die Achse E_i zusammenfällt.

Der Modaloperator unt_s (s steht für spatial) wird folgendermaßen benutzt:

$$(S, r, E_i) \models \phi_1 \text{UNT}_{s+} \phi_2$$

bedeutet, daß es einen Punkt r' gibt, der erreicht wird, wenn man sich vom Punkt r aus in positiver Richtung der Achse E_i bewegt, so daß ϕ_2 am Punkt r' gilt und ϕ_1 an allen Punkten entlang des Wegs von r nach r' .

Mit Hilfe dieser Konstrukte kann die Position eines Objekts relativ zu einem Punkt der Szene spezifiziert werden. Um auch die Spezifikation relativ zu einem anderen Objekt zu ermöglichen, werden Kontextdeklaratoren der Form $(S, q, E_i) \models \phi$ eingeführt. Dies bedeutet, daß ϕ für alle Punkte des Objekts q gilt.

Diese statischen Frame-Aussagen, die die Positionen der Objekte der Szene in einem (statischen) Frame beschreiben, können nun mit Operatoren der temporalen Logik kombiniert werden, um so Sequenzaussagen zu bilden. Die basale Sequenzaussage lautet

$$(\sigma, j) \models \Phi,$$

wobei Φ eine Frame-Aussage ist. Dies bedeutet, daß Φ im j -ten Frame der Sequenz σ gilt.

Der temporale Operator unt_t wird folgendermaßen benutzt:

$$(\sigma, j) \models \theta_1 \text{UNT}_t \theta_2$$

bedeutet, daß es einen Index $k \geq j$ gibt, so daß θ_2 im k -ten Frame der Sequenz σ gilt und θ_1 mindestens vom j -ten bis zum k -ten Frame der Sequenz σ gilt.

Sequenzaussagen können unterschiedlich detailliert gemacht werden. Interessanterweise deckt die feinste Granularitätsstufe genau die 13 Allenschen Intervallrelationen ab [1], während die gröberen Granularitäten als Disjunktionen dieser Intervallrelationen ausgedrückt werden können.

Mit Hilfe dieser Sequenzaussagen können nun Angaben über das spatiotemporale Geschehen in Bildfolgen gemacht und so deren Inhalt beschrieben werden. Dies macht den Einsatz als Datenbankindex möglich. Eine Implementierung dieser Indizierungstechnik erlaubt query by example: Der Benutzer kann virtuelle Sequenzen definieren, indem er sich bewegendes Icons in einen virtuellen 3D-Raum plaziert. Ein spatiotemporaler Parser interpretiert die Beispielsequenz und übersetzt sie in eine Sequenzaussage, die mit denen der Sequenzen in der Datenbank verglichen wird. Passende Sequenzen werden dem Benutzer dann vorgespielt.

Weil STL nur topologische Relationen in Raum und Zeit wiedergibt, schlagen Del Bimbo und Vicario in [5] eine Erweiterung namens XSTL vor, die eine „qualitative Metrik“ in Raum und Zeit enthält. Durch eine Kombination von räumlichen und zeitlichen metrischen Relationen werden Aussagen über die relative Geschwindigkeit der Objekte in einer Sequenz möglich.

2.5 Welche Art von Bewegungsverläufen repräsentieren wir hier?

In den oben referierten Arbeiten geht es meist um Aussagen, die über Bewegungsverläufe gemacht werden können – ob die Bewegung kontinuierlich ist, oder ob bestimmte Raumregionen während der Bewegung eingenommen wurden, usw. Uns interessiert hingegen eine Beschreibung des tatsächlichen Verlaufs der Bewegung, also der Trajektorie und der Form der Spur, so daß die axiomatischen Ansätze für uns weniger geeignet sind. Wir brauchen eine deskriptivere Beschreibung. Die rein topologischen Raumrepräsentationen sind zu strukturarm, um für diese Aufgabe nützlich zu sein, weswegen wir uns auf ein Modell mit qualitativer Orientierung und qualitativen Distanzen stützen, wie es in [12] eingeführt wurde.

Diese Arbeit beschäftigt sich mit der Repräsentation von Bewegung punktförmiger Objekte im zweidimensionalen Raum. Die Erweiterung auf 3D ist allerdings kanonisch und wird jeweils angesprochen werden.

3 Stufen der Abstraktion

Ein und derselbe Bewegungsverlauf kann qualitativ auf verschiedenen Stufen der Abstraktion und Genauigkeit und damit in verschiedenen Granularitäten beschrieben werden, wobei die Beschreibungen jeweils verschiedene Zwecke erfüllen sollen. Die Wahl der richtigen Granularität hängt sicher auch von der Anwendung ab. Ein Kunstflieger, der sein Programm zusammenstellt, muß dieses sicherlich nicht auf der Ebene exakter Trajektorienverläufe tun. Hierzu genügt die Angabe einer Sequenz der zu fliegenden Figuren nicht nur völlig, sondern eine genaue Angabe der Flugbahn wäre im Gegenteil für einen Kampfrichter nur verwirrend und würde die Kommunikation erschweren. Dagegen weiß jeder der an so einem Wettbewerb Beteiligten, was er sich unter einer gerissenen Rolle vorzustellen hat und wie die Trajektorie aussieht, wenn sie sauber ausgeführt ist. Auf der anderen Seite wäre z.B. für die NASA bei der Berechnung der Flugbahn einer Mondrakete eine Beschreibung auf dieser hohen Abstraktionsebene sicher nicht ausreichend, da es hier nicht nur darauf ankommt, wie die Flugbahn aussieht, sondern vielmehr entscheidend ist, wann sich die Rakete an welchem Ort befindet und welcher Impuls nötig ist, um sie zum richtigen Zeitpunkt dort hinzubringen.

3.1 Zwei-Schichten-Architektur

Um diesen Sachverhalt zu reflektieren, schlagen wir für die qualitative Repräsentation von Bewegungsverläufen eine Architektur in Schichten vor, wobei in den unteren Schichten die Genauigkeit größer und die Granularität somit feiner ist (die allerunterste Schicht wäre demnach eine exakte mathematisch-numerische Repräsentation), wohingegen auf oberen Schichten der Grad der Abstraktion höher ist und bedeutungsvollere Konzepte den grobgranularen Darstellungen zugrunde liegen (wie in obigem Kunstflugbeispiel). Was in Bezug auf einen Bewegungsverlauf ein bedeutungsvolles Konzept ist, hängt natürlich stark von der Anwendung ab. Im Kunstflugbeispiel wären das Figuren wie Looping, Rolle oder Trudeln, bestimmt durch die Form von Teilen der Trajektorie.

Konkret wurde eine Zweischichtrepräsentation, wie in Abbildung 3.1 dargestellt, realisiert. Auf der unteren Schicht wird hierbei der Bewegungsverlauf durch eine diskrete Trajektoriendarstellung mittels qualitativer Bewegungsvektoren (*Qualitative Motion Vectors*, QMVs) repräsentiert. Auf der oberen Schicht wird vom Orts-/Zeitverhalten

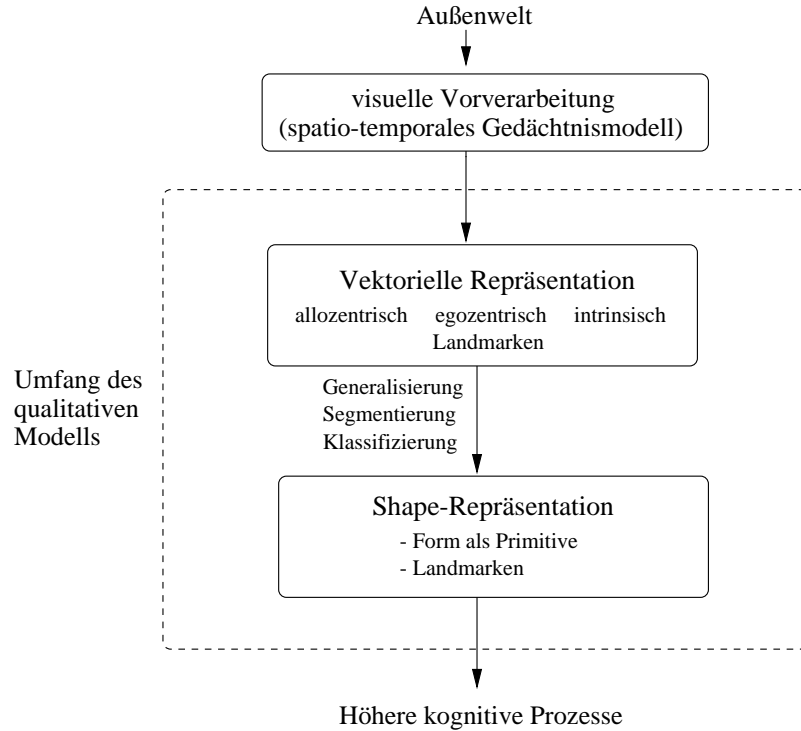


Abbildung 3.1: Qualitative Repräsentation von Bewegungsverläufen in zwei Schichten

des sich bewegenden Objekts stärker abstrahiert und die Formen von Teilen des Bewegungsverlaufs werden – wie z.B. bei Kunstflugfiguren – als Primitiva der Repräsentation verwendet. Die Form der Trajektorie ist ja anwendungsunabhängig als ein herausragendes qualitatives Konzept identifiziert worden (siehe Abschnitt 2.3). Die SHAPE-Repräsentation kann aus der QMV-Repräsentation (die ja genauer und feingranularer ist) automatisch erzeugt werden durch Generalisierung, Segmentierung und Klassifizierung der QMV-Sequenz.

Dieses Kapitel stellt die Architektur der Repräsentation im Überblick vor und gibt ein kurzes Beispiel, ohne auf Einzelheiten und Probleme wie Referenzsysteme, Quantity Spaces, usw. einzugehen. Dies wird in den folgenden Kapiteln geschehen.

3.2 Qualitative Bewegungsvektoren

Wie können die Einzelheiten einer Trajektorie *qualitativ* dargestellt werden? Eine Bewegung kann verstanden werden als Veränderung der Position eines Objekts über die Zeit.

Eine Möglichkeit, die Aufgabe zu lösen, wäre also, qualitative Objektpositionen mit qualitativer Zeitangabe zu modellieren. Dies mag für manche Aufgaben eine gute Wahl sein (z.B. wenn man eine Reiseroute anhand von Etappen grob beschreiben möchte, wie in Abbildung 3.2 dargestellt), ist aber u.a. für unser Kunstflugbeispiel völlig ungeeignet. Wie unterscheiden sich im Luftraum bestimmte Positionen qualitativ voneinander? Ist es überhaupt sinnvoll, die Bewegung eines Kunstflugzeugs durch die Angabe verschiedener absoluter Positionen zu beschreiben? Sicherlich nicht. Wichtig sind hier die *relativen* Positionen, also die Angabe von Richtungsänderungen und Distanzen relativ zur letzten Position. Dies ist der Ansatz, den wir auch mit unserer qualitativen Bewegungsbeschreibung verfolgen.



Abbildung 3.2: Qualitative Repräsentation einer Zugfahrt

Wenn wir also nicht absolute Positionen und Zeiten repräsentieren wollen, sondern lediglich die Veränderung der Position eines Objekts über die Zeit, genügt es, die jeweilige Richtung der Bewegung, die Distanz der Bewegung zwischen zwei Richtungsänderungen sowie die Geschwindigkeit, mit der diese Bewegung erfolgte, zu repräsentieren. Da sich in einer natürlichen Bewegung alle diese Parameter kontinuierlich und fließend ändern, ist es für eine qualitative Repräsentation wichtig, einige Entscheidungen über die Art und Weise der Diskretisierung des Bewegungsverlaufs zu treffen. Diese werden in den

folgenden Abschnitten für die Domänen Zeit, Richtung und Distanz beschrieben.

3.2.1 Zeit

Da viele technische Systeme eine diskrete Zeitaufösung haben (= Taktzyklen), liegt es vom technischen Standpunkt her erst einmal nahe, einen Bewegungsverlauf auch in diskreten Zyklen zu beobachten. Auch in der Kognitionsforschung gibt es Argumente für eine diskrete Abtastung, siehe z.B. [69]. Diese Vorgehensweise hat den weiteren Vorteil, daß Zeit implizit repräsentiert werden kann und Distanz äquivalent zu Geschwindigkeit wird: Je größer die Distanz, die ein Objekt in einem Abtastzyklus zurückgelegt hat, desto höher die Geschwindigkeit, mit der es sich bewegt hat.¹ Wenn man nicht interpoliert, was zwischen den Meßzeitpunkten passiert ist, muß man annehmen, daß sich das Objekt in der Zwischenzeit linear bewegt hat. Der örtliche Verlauf der Bewegung (die Spur) wird somit durch einen Polygonzug beschrieben. Fügt man an den Abtastpunkten Markierungen ein, so beschreiben diese zeitlich äquidistante Punkte. Eine wahrgenommene Richtungsänderung fällt immer mit einer dieser Markierungen zusammen, aber nicht an jeder Markierung muß eine Richtungsänderung stattfinden. Die örtliche Distanz zwischen den Markierungen codiert dabei die Geschwindigkeit, mit der sich das Objekt in der Zwischenzeit bewegt hat. Große Distanzen entsprechen hohen Geschwindigkeiten, kleine niedrigen.

3.2.2 Richtung

Ausgehend von diesem Polygonzug mit den temporal äquidistanten Markierungen, wollen wir nun eine qualitative Repräsentation finden, die den örtlichen Verlauf der Spur nicht in absoluten Positionsangaben beschreibt. Wie schon erwähnt, ist hier eine Repräsentation geeignet, die örtliche Veränderung relativ zur letzten Position codiert, und zwar in den Domänen Richtung und Entfernung. Die Richtung des Bewegungsverlaufs zwischen zwei Markierungen des Polygonzugs p_i und p_{i+1} ist nun einfach die Orientierung von Punkt p_{i+1} relativ zu Punkt p_i . Somit können hier die aus dem qualitativen räumlichen Schließen gut bekannten Orientierungsrelationen zur Repräsentation der Richtung verwendet werden (siehe z.B. [42]). Ausgehend von Punkt p_i wird der Raum in Sektoren aufgeteilt, die mit natürlichsprachlichen Bezeichnern wie „Norden“ oder „Vorwärts“ versehen werden; die Bezeichnung des Sektors, in dem der Punkt p_{i+1} liegt, ist dann die qualitative Bewegungsrichtung (siehe auch Abbildung 3.3).

Die Erweiterung auf den dreidimensionalen Raum, um auch unser Kunstflugbeispiel mo-

¹Hier kann es natürlich, wenn die Abtastrate schlecht gewählt ist, zu „Entartungen“ kommen: Wenn ein Objekt sich sehr schnell hin und her bewegt, alle 30 Sekunden am selben Platz vorbeikommt und die Abtastrate genau eine Abtastung alle 30 Sekunden beträgt, sieht das natürlich wie ein Stillstand aus.

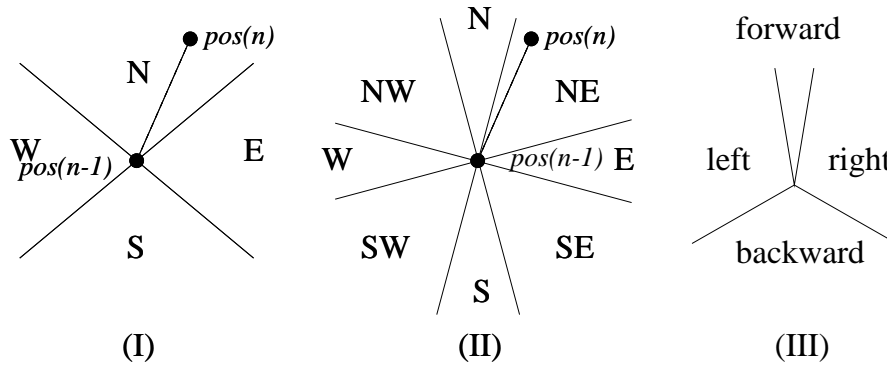


Abbildung 3.3: Qualitative Repräsentation von Bewegungsrichtung

dellieren zu können, ist hier kanonisch. Ausgehend von Punkt p_i wird der Raum in Volumensektoren aufgeteilt, die dann mit natürlichsprachlichen Bezeichnern wie „links oben“ oder „Nordwest-hoch“ versehen werden. Wie im zweidimensionalen Fall ist die Bezeichnung des Sektors, in dem der Punkt p_{i+1} liegt, dann die qualitative Bewegungsrichtung.

3.2.3 Distanz und Geschwindigkeit

Ausgehend von dem Polygonzug, verhält es sich mit der Distanz ganz genau wie mit der Richtung. Die Weite einer Bewegung zwischen zwei Markierungen des Polygonzugs p_i und p_{i+1} kann einfach codiert werden durch die aus dem qualitativen räumlichen Schließen bekannten Distanzrelationen, siehe z.B. [12] und Abbildung 3.4.

Bei der Entfernung als eindimensionaler Größe gibt es zwischen dem zwei- und dreidimensionalen Fall keine Unterschiede.

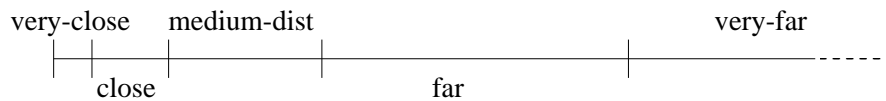


Abbildung 3.4: Qualitative Repräsentation von Bewegungsweite

Da die Bewegungsgeschwindigkeit von der Weite der Bewegung in einem Abtastzyklus abgeleitet wird, ist es sinnvoll, auch nur soviele Geschwindigkeiten zu unterscheiden, wie man Entfernungen unterscheiden kann, da feiner sowieso nicht differenziert werden kann. Die explizite qualitative Repräsentation von Geschwindigkeit in einem Segment des Polygonzugs kann deswegen durch einfache Abbildung der entsprechenden Distanzrelation auf eine qualitative Bezeichnung für Geschwindigkeit geschehen; also z.B. sinnvollerweise

durch Abbildung von „sehr nah“ auf „sehr langsam“, „nah“ auf „langsam“, . . . „sehr weit“ auf „sehr schnell“.

3.2.4 QMVs

Ein qualitativer Bewegungsvektor (QMV) v_i nun ist ein Vektor, der die Positionsveränderung eines Objekts zwischen zwei Markierungen p_i und p_{i+1} des die Trajektorie repräsentierenden Polygonzugs beschreibt, und zwar mittels qualitativer Komponenten für Richtung und Entfernung. Da die zeitliche Entfernung δ der Markierung des Polygonzugs bekannt ist, ist die Geschwindigkeit hierin implizit codiert, kann aber auch durch einfache Abbildung der Distanzrelationen in qualitative Geschwindigkeiten explizit angegeben werden.

Auf dieser Stufe der Abstraktion wird also ein Bewegungsverlauf repräsentiert durch eine Sequenz von QMVs der Art

$$\langle \text{Dist}_1 \text{Dir}_1 \rangle^{i_1} \langle \text{Dist}_2 \text{Dir}_2 \rangle^{i_2} \dots \langle \text{Dist}_n \text{Dir}_n \rangle^{i_n},$$

wobei die Dist_i für qualitative Distanzrelationen stehen, die Dir_i für qualitative Richtungsrelationen, und die $i_i \in \mathbb{N}$ Zähler sind, die angeben, für wieviele Zeiteinheiten δ sich Geschwindigkeit (also Distanz) und Richtung der Bewegung nicht geändert haben. Die Zähler sind lediglich eine abkürzende Schreibweise, um kürzere Vektorsequenzen zu erzielen.

Verschiedene Möglichkeiten, die qualitativen Kategorien von Richtung und Entfernung zu definieren, werden im Kapitel 4 noch ausführlich diskutiert werden.

Das Beispiel zeigt eine QMV-Sequenz, die mit der Mauseingabe des MM-TOOLS² (dargestellt in Abbildung 3.5) generiert wurde:

```
<0 0>1 <close east>5 <close north>2 <close west>3 <close south>1
<medium-dist south>1 <medium-dist east>1 <far east>1 <0 0>1
<far east>2 <close east>1
```

Diese Spur beschreibt also eine Bewegung, bei der das Objekt zunächst für eine Zeiteinheit stillsteht, sich dann für 5 Zeiteinheiten jeweils eine kurze Distanz Richtung „Osten“ bewegt, dann für zwei Zeiteinheiten jeweils eine kurze Distanz Richtung „Norden“, dann für drei Zeiteinheiten jeweils eine kurze Distanz Richtung Westen, usw.

Repräsentiert man die Geschwindigkeiten explizit, erhält man folgende Sequenz:

```
<0 0 0>1 <close east slow>5 <close north slow>2 <close west slow>3
<close south slow>1 <medium-dist south medium-vel>1
<medium-dist east medium-vel>1 <far east fast>1 <0 0 0>1
<far east fast>2 <close east slow>1
```

Wenn man mehrere gleiche Vektoren zusammenfassen will (wie z.B. hier <clo-

²Es handelt sich hier um eine ältere Implementierung des MM-TOOLS in Scheme/Tk

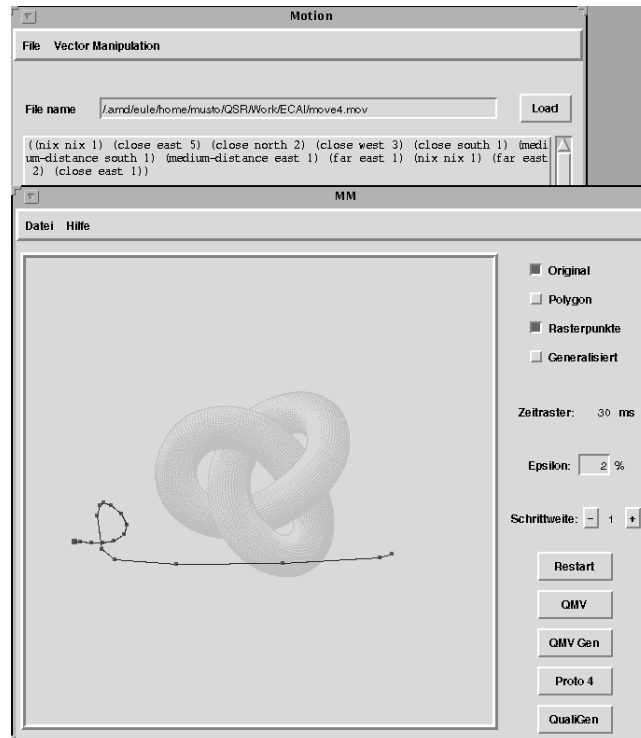


Abbildung 3.5: Beispielsequenz

se east slow>⁵, was ja für 5 Vektoren steht, muß man natürlich die Distanzen addieren (5 mal close kann ja durchaus eine größere Distanz ergeben, z.B. medium-dist), während die ursprünglichen Geschwindigkeiten selbstverständlich gleich bleiben, also z.B.

```
<0 0 0>1 <medium-dist forward slow> <close left slow> <close left slow>
<close left slow> <medium-dist forward medium-vel> <medium-
dist left medium-vel> <far forward fast> <0 0 0>1 <far forward fast>
<close forward slow>
```

Die Richtungsrelationen, die hier absolut in north, east, south und west benannt sind, können auch relativ ausgedrückt werden; und zwar immer relativ zur Richtung der vorangegangenen Bewegung wie bei „rechts abbiegen“, siehe auch Abbildung 3.3 (III). Start-Richtung ist dabei immer forward. Unsere Beispielsequenz sieht dann folgendermaßen aus:

```
<0 0 0>1 <medium-dist forward slow> <close left slow> <close left slow>
<close left slow> <medium-dist forward medium-vel> <medium-
dist left medium-vel> <far forward fast> <0 0 0>1 <far forward fast>
```

<close forward slow>

Die QMV-Repräsentation kann natürlich mit verschiedenen Granularitäten arbeiten; dies hängt lediglich von den gewählten Quantity Spaces und der gewählten zeitlichen Granularität ab. So kann z.B. die Richtungsdomäne in 4 oder 8 qualitative Richtungen aufgeteilt werden (wie in Abbildung 3.3 zu sehen) und die Entfernungsdomäne in eine unterschiedliche Anzahl qualitativer Entfernungen.

3.3 Shape-Repräsentation

Schon im Grundsatz grobgranularer, weil auf einer höheren Abstraktionsebene angesiedelt, ist die Beschreibung eines Bewegungsverlaufs durch Mittel, wie sie der Kunstflieger oder Eisläufer benutzt, nämlich durch eine Abfolge von vordefinierten Figuren. Diese Art der Beschreibung gruppiert Teile einer Trajektorie zu qualitativ bedeutsamen Einheiten und strukturiert damit den Bewegungsverlauf. Ganz allgemein gesagt wird ein Bewegungsverlauf unterteilt in Stücke, die eine gewisse (vordefinierte) *Form* (SHAPE) beschreiben.

Für eine qualitative Repräsentation auf dieser Stufe der Abstraktion müssen wir natürlich ein Vokabular von MOTION SHAPES definieren, das die Formen enthält, die im gewählten Kontext eine bestimmte Bedeutung haben. In unserem Kunstflugbeispiel wäre das ein Vokabular wie $\{Rolle, Looping, Avalanche, Cuban\ Eight, Humpty\ Bump, \dots\}$. In Abbildung 3.6 sind ein paar dieser Figuren in der internationalen Notation Aresti³ dargestellt.

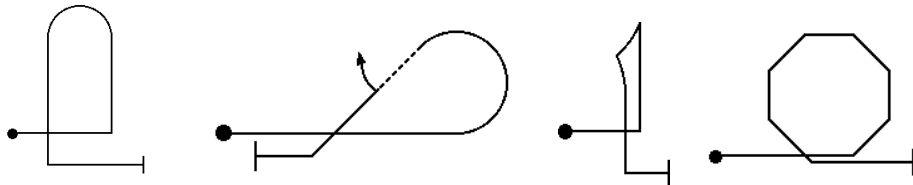


Abbildung 3.6: Kunstflugfiguren: Humpty-Bump, Half Cuban Eight, Tail Slide, Eight-side-loop

Allgemeiner und im Zweidimensionalen kann dies ein Vokabular wie $\{straight-line, left-turn, right-turn, u-turn-left, loop-left, \dots\}$ sein. Auf dieser Stufe der Abstraktion ist ein Bewegungsverlauf somit eine Sequenz von MOTION SHAPES, die mit gewissen Attributen wie Größe, Ähnlichkeit zum vordefinierten Prototypen, Ausführungsgeschwindigkeit,

³Benannt nach dem Kunstflug-Pionier Jose Aresti. Eine kurze Einführung in die Bedeutung dieser Notation findet sich auf http://acro.harvard.edu/IAC/acro_figures.html; Einführende Literatur wird in http://acro.harvard.edu/IAC/faq_aerobatics.html#books angegeben.

Varianz in der Kinetik, usw. versehen werden können:

```
<Shape1 attrib11 ... attrib1n> <Shape2 attrib21 ... attrib2n> ...
<Shapem attribm1 ... attribmn>.
```

Wenn im Vokabular Formen wie $\{left\text{-turn}, right\text{-turn}, u\text{-turn-left}, \dots\}$ enthalten sind, ist eine relationale Beschreibung von Richtungswechseln zwischen den einzelnen MOTION SHAPES nicht nötig, da jeder Richtungswechsel wiederum durch einen eigenen MOTION SHAPE repräsentiert werden kann.

Die QMV-Sequenz aus unserem Beispiel kann damit dargestellt werden als:

```
<straight-line small slow> <loop-left small slow> <straight-line large
very-fast>.
```

Man sieht sofort, daß diese Darstellung nicht nur kürzer und kompakter ist, sondern auch sofort eingängiger und leichter kommunizierbar. Während die QMV-Sequenz zwar genauer war, ist die Beschreibung mittels MOTION SHAPES besser geeignet, unsere Vorstellung zu evozieren.

3.4 Gemischte Granularitäten

Manchmal werden Beschreibungen benötigt, bei denen die Granularitäten gemischt werden können. Wenn man z.B. auf einer Konferenz in einer fremden Stadt bei einem abendlichen Spaziergang einen schönen Weihnachtsmarkt auf dem Domplatz entdeckt, und am nächsten Tag dem Kollegen den Weg zu der Bude beschreiben möchte, an der man diese wunderbaren Weihnachtsengel gekauft hat, wird man für die Wegbeschreibung sicherlich unterschiedliche Granularitäten benutzen. Die Wegbeschreibung wird sich auf den zurückgelegten Bewegungsverlauf stützen (wie in Abbildung 3.7 dargestellt), der aber auf unterschiedlichen Abstraktionsebenen wiedergegeben wird:

Die Feinstruktur (hier gestrichelt) des tatsächlich zurückgelegten Weges ist für eine Routenbeschreibung uninteressant und wird deswegen generell weggelassen. Die Wegbeschreibung bis zum Domplatz ist sicher grobgranular und wird sich auf Landmarken und Richtungsänderungen beschränken (wobei Elemente der SHAPE-Repräsentation, wie z.B. die S-Kurve in Abbildung 3.7, als besondere Landmarken benutzt werden können). Auf dem Platz selber muß die Beschreibung dann feingranularer werden und vielleicht sogar geschätzte Weglängen enthalten, damit die eine unter lauter gleich aussehenden Buden gefunden werden kann.

Für unsere Zweischicht-Repräsentation bedeutet das, daß wir eventuell mitten im Bewegungsverlauf die Granularität wechseln wollen und ein Stück des Bewegungsverlaufs als Abfolge von SHAPES beschreiben, und ein anderes Stück als QMV-Sequenz. Das wiederum heißt, daß unsere MOTION SHAPES als QMV-Sequenzen ausdrückbar sein müssen und umgekehrt. In der zweiten Richtung ist das selbstverständlich, da die SHA-

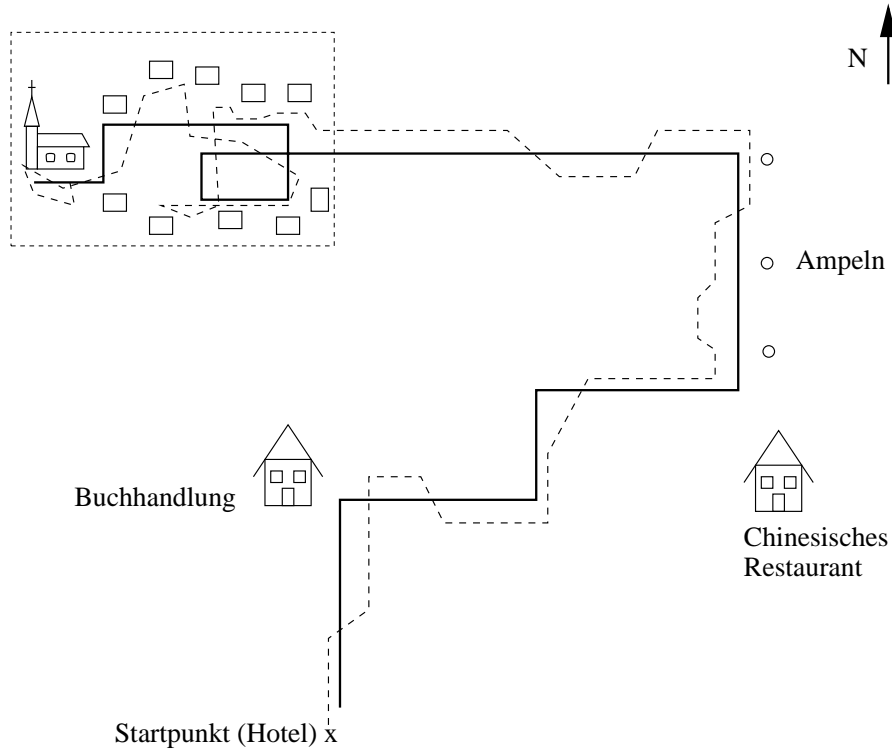


Abbildung 3.7: Bewegungsverlauf mit Landmarken und Generalisierung

PE-Repräsentation schließlich aus der QMV-Repräsentation erzeugt wird, wie wir im Kapitel 5 sehen werden. In die andere Richtung heißt das, daß wir beim Konstruktionsprozeß die Subsequenz der QMV-Sequenz, die zu jedem SHAPE korrespondiert, speichern müssen. Falls die SHAPE-Sequenz nicht aus einer QMV-Sequenz erzeugt wurde, muß auf prototypische Repräsentationen zurückgegriffen werden. Da bei der Konstruktion der SHAPE-Repräsentation aus einer QMV-Sequenz die QMVs normalerweise „auseinandergeschnitten“ werden — ein **left-turn** besteht aus zwei QMVs, von denen jeder, wenn sie sich in der Mitte einer Sequenz befinden, wiederum zu einem weiteren SHAPE beiträgt —, geht hier die zeitliche „Taktung“ möglicherweise verloren. Sinnvoll ist hier also eine Angabe der QMVs in der Form `<Dist Dir Speed>`.

Eine gemischte Repräsentation unserer Beispielsequenz könnte z.B. so aussehen:

```
<straight-line small slow> <loop-left small slow>
<medium-dist south medium-vel> <medium-dist east medium-vel>
<far east fast> <0 0 0>1 <far east fast> <close east slow>.
```

4 Quantity Spaces

Eine der Fragen, die wir im Kurzüberblick über die qualitative Repräsentation im letzten Kapitel beiseitegelassen haben, ist die Konstruktion einer konkreten Abbildung der „wirklichen Welt“ auf die qualitativen Intervalle (die Quantity Spaces). Wir haben lediglich kurz angesprochen, daß wir die Orientierungs- und Distanzrelationen wie z.B. in [42, 12] eingeführt, benutzen können. Es gibt jedoch viele verschiedene Möglichkeiten, wie die Intervalle, in die die numerischen Werte abgebildet werden, definiert werden können. Drei Möglichkeiten für die Diskretisierung der Richtungsdomäne sind schon in Abbildung 3.3 illustriert; diese betreffen aber alle nur ein bestimmtes Paradigma, nämlich die nicht-überlappende, exhaustive Zerlegung.

Von den vielen denkbaren Zerlegungsparadigmen werden wir im folgenden zwei Möglichkeiten diskutieren, nämlich die bereits angesprochene nicht-überlappende Zerlegung, und eine Zerlegung in Fuzzy-Intervalle. Nicht-exhaustive Zerlegungen werden wir nicht berücksichtigen, da Zerlegungen, bei denen Teile des Raumes zu keinem Intervall gehören, problematisch sind. Die Fuzzy-Modellierung von Bewegungsverläufen in Abschnitt 4.2 stützt sich letztendlich wieder auf eine nicht-überlappende Zerlegung.

4.1 Diskrete Intervalle

Wie in jedem qualitativen Modell wird die Wahl der geeigneten Diskretisierung vom Kontext der Anwendung abhängen. Da eine der wichtigsten Motivationen für die Einführung qualitativer Modellierungen die Erleichterung der Mensch-Maschine-Interaktion ist, wollen wir uns bei unseren Versuchen zunächst an der menschlichen Kognition orientieren.

4.1.1 Distanz

Ein Quantity Space für Distanzen (wie in Abbildung 3.4 zu sehen) muß in Bezug auf eine vordefinierte Größe gewählt werden, die natürlich kontextabhängig ausgesucht werden muß. Wenn es zum Beispiel darum geht, Bewegungsabläufe zu modellieren, die sich im Gesichtsfeld eines Benutzers abspielen (z.B. auf dem Computerbildschirm), ist die Gesamtheit des zur Verfügung stehenden Raumes einer natürlichen Grenze unterworfen

(sowohl Gesichtsfeld als auch Bildschirmoberfläche sind begrenzt). Bewegungen auf einem Fußballfeld oder in einem Eislaufstadion können sich ebenfalls nicht beliebig weit ausdehnen. Die Größe eines derartig beschränkten Feldes kann somit benutzt werden, um einen Referenzwert zu definieren – dies kann z.B. die Länge des Fußballfeldes sein oder die Bildschirmdiagonale. Jede gemessene Distanz wird dann mit dem Referenzwert verglichen und, abhängig vom Ergebnis dieses Vergleichs, unter eine gewisse linguistische Variable subsumiert (siehe auch [55]).

Beispiel 4.1

Wir wollen einen Bewegungsverlauf modellieren, der sich auf einem Bildschirmfenster abspielt, das 15 cm im Diagonaldurchmesser hat. Wir wählen uns dazu einen Referenzwert aus, der ca. die halbe Länge mißt, also $d_{const} = 7$ cm. Die Relation r eines gemessenen Wertes d_m und d_{const} ist $r = \frac{d_m}{d_{const}}$, z.B. $\frac{3}{7}$. Nun können wir in einer vordefinierten Tabelle nachsehen und erhalten eine linguistische Variable für unseren gemessenen Wert. So eine Tabelle kann z. B. folgendermaßen aussehen:

zero	=	$[0, 0]$
very-close	=	$(0, \frac{1}{3}]$
close	=	$(\frac{1}{3}, \frac{2}{3}]$
medium-dist	=	$(\frac{2}{3}, 1]$
far	=	$(1, \frac{3}{2}]$
very-far	=	$(\frac{3}{2}, \max]$

Möglicherweise gibt es zusätzliche Bedingungen, die bei der Instanzierung dieser Quantity Spaces eingehalten werden müssen. So kann man z. B. die Bedingung formulieren, daß für jedes Intervall $(t_a, t_b] \neq \mathbf{zero}$ der Anfangspunkt für das nächste Intervall $t_b = n \times t_a$ für ein festes $n \in \mathbb{N}, \mathbb{Q}$ oder \mathbb{R} sein soll. Dann müssen wir eine andere Instanzierung finden, die diese Bedingung erfüllt. Dazu definiert man zunächst einen „virtuellen“ Startpunkt im ersten Intervall (weil man 0 nicht nehmen kann) wie z. B. 0.1, und erhält dann z. B. für $n = 2$ folgende Tabelle:

zero	=	$[0, 0]$
very-close	=	$(0, 0.2]$
close	=	$(0.2, 0.4]$
medium-dist	=	$(0.4, 0.8]$
far	=	$(0.8, 1.6]$
very-far	=	$(1.6, \max]$

Die Granularität der Diskretisierung in 5 Distanzbereiche ist natürlich nicht fest, sondern hier willkürlich festgelegt. Man könnte genauso gut 3, 6 oder 10 Distanzen unterscheiden (oder eine beliebige andere Zahl, wobei natürlich eine sehr feine Diskretisierung fast

wieder einer quantitativen Darstellung nahekommt). Ein Indiz, daß eine feinere Unterscheidung als in 5 Distanzbereiche möglicherweise keinen Sinn macht, ist die Tatsache, daß hier schon die natürlichsprachlichen Worte fehlen, um die Intervalle zu bezeichnen. Welche Unterteilung in der Bewegungswahrnehmung sinnvoll ist, also z. B. für die Wahrnehmung eines sich bewegenden Punktes auf dem Bildschirm, dafür gibt es unseres Wissens nach noch keine Hinweise aus der Psychophysik.

Eine andere Möglichkeit, einen Referenzwert zu definieren, ist, die Ausdehnung des bewegten Objekts bzw. Körpers und die Reichweite der Extremitäten zu benutzen. Dann wäre evtl. eine Unterteilung im Grüsserschen Sinne sinnvoll. Grüsser [40]¹ unterscheidet vier Arten von extrapersonalem Raum auf der Grundlage der Handlungen, die man in ihnen jeweils vornehmen kann. Der „grasping space“ oder Greifraum ist der Raum um die Person, in dem sie Objekte direkt manipulieren kann. Der „near-distance action space“ oder nahe Handlungsbereich ist der Raum, in dem eine Person sich leicht bewegen und agieren kann. Grüsser definiert seine Größe als die unmittelbare Umgebung der Person, in der sie mit verbundenen Augen umhergehen kann, ohne sich über ihre Position unsicher zu werden. Der „far-distance action space“ oder entfernte Handlungsbereich ist der Raum, der so weit von der Person entfernt ist, daß Distanzen im Wesentlichen durch visuelle und auditive Reize erfahren werden. Zuletzt nennt Grüsser den „visual background“ oder visuellen Hintergrund, der als Grenze des extrapersonalen Raums erfahren wird und in dem keine Tiefe mehr wahrgenommen wird. Für diese Unterscheidung gibt es nicht nur kognitive, sondern auch neuronale Entsprechungen. Es könnte also durchaus sinnvoll sein, für Eigenbewegungen den Raum in drei Bereiche² zu diskretisieren und mit dem Greifraum, dem nahen und entfernten Handlungsbereich zu identifizieren.

4.1.2 Richtung

Die Bewegungsrichtung eines Objekts in einem bestimmten Abtastzyklus n kann definiert werden als die Orientierung der Position des Objekts zum Abtastzeitpunkt n relativ zur Position des Objekts zum Abtastzeitpunkt $n - 1$. Das bedeutet, daß wir, wie bereits erwähnt, hier die bekannten Orientierungsrelationen aus der Literatur zum qualitativen räumlichen Schließen benutzen können, wie z. B. in [42, 12]. Es muß aber jeweils noch die Richtung **zero** vorhanden sein, da Vektoren der Länge **zero** natürlich auch keine eindeutige Richtung haben können; so kann Stillstand modelliert werden.

Um die menschliche Kognition zu modellieren, kann die Diskretisierung in 4, 8 oder 16 gleich große Intervalle allerdings unangebracht erscheinen. Menschen können Richtungen nahe der vertikalen und horizontalen Achse bedeutend feiner unterscheiden als „schräge“ Richtungen (der sogenannte „oblique effect“, siehe z.B. [39]). In [89] wurde außerdem

¹zitiert nach [88]

²Der visuelle Hintergrund ist zu weit weg, als daß er für die Repräsentation von Eigenbewegung eine Rolle spielen könnte.

ermittelt, daß Menschen die Ausdrücke für kanonische Relationen wie „rechts“, „links“, „oben“, „unten“ nur in einem sehr schmalen Bereich benutzen, und zwar bei maximal 15° Abweichung von der vertikalen und horizontalen Achse. Für alle anderen Bereiche werden zusammengesetzte Ausdrücke verwendet, wie „links oben“, „rechts unten“, usw. Eine Diskretisierung, die diesen Effekt bzw. Sprachgebrauch widerspiegelt, ist in Abbildung 3.3 (II) zu sehen.

Abbildung 3.3 (III) zeigt eine asymmetrische Diskretisierung. Dies kann nützlich sein, wenn man eine Richtung (hier: vorwärts) feiner unterscheiden will als andere Richtungen. Allerdings ist das Rechnen mit asymmetrischen Diskretisierungen schwieriger, so daß wir nur symmetrische Quantity Spaces in der Richtung berücksichtigen.

Die unterschiedliche Benennung der Richtungen in den verschiedenen Abbildungen rührt von der Verwendung verschiedener Referenzsysteme her; im einen Fall (vorwärts, rückwärts, rechts, links, . . .) verwenden wir ein Referenzsystem relativ zur letzten Bewegung (die erste ist dann immer „vorwärts“), der sich auch mit dem Bewegungsverlauf ständig fortbewegt und dreht. Im anderen Fall (Nord, Süd, Ost, West, . . .) handelt es sich um einen absoluten Referenzrahmen, der fest bleibt. Die genauen Implikationen dieser Unterscheidung werden wir in Kapitel 6 diskutieren.

4.1.3 Beispiele

Abbildung 4.1 zeigt zwei Bewegungssequenzen, die mit dem MM-TOOL erzeugt wurden. Die Bewegung wird diskret alle 30 ms³ abgetastet und liegt dann als Vektorsequenz vor, die weiterverarbeitet werden kann. Die runden Punkte sind die Abtastpunkte, die jeweils 30ms auseinanderliegen. Die etwas dickere Linie ist eine Generalisierung.

Die Generalisierung der Spur auf der linken Seite ergibt mit einer Diskretisierung in 8 gleichgroße Richtungsbereiche folgende QMV-Sequenz:

```
<medium-dist east>1 <far north>1 <far east>1,
```

und mit einer Diskretisierung in 8 Richtungsbereiche, wobei die Intervalle an den Haupt-himmelsrichtungen 30° groß sind, die an den „schiefen“ Himmelsrichtungen 60° groß sind (siehe Abbildung 3.3 (II)), folgende QMV-Sequenz:

```
<medium-dist east>1 <far northeast>1 <far east>1.
```

Die Generalisierung der Bewegungssequenz in Abbildung 4.1 (rechte Seite) ergibt mit einer Diskretisierung in 4 gleichgroße Richtungsbereiche folgende QMV-Sequenz:

```
<medium-dist east>1 <far north>1 <far east>1,
```

und mit einer Diskretisierung in 8 gleichgroße Richtungsbereiche folgende QMV-Sequenz:

```
<medium-dist east>1 <far northeast>1 <far east>1.
```

³Dieser Parameter ist einstellbar.

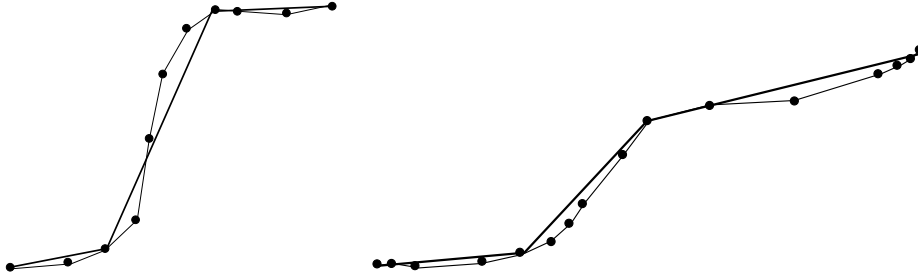


Abbildung 4.1: Links: Beispielsequenz für Diskretisierung in 8 gleich- oder verschieden große Richtungsbereiche. Rechts: Beispielsequenz für Diskretisierung in 4 oder 8 gleichgroße Richtungsbereiche

4.1.4 Rechnen in diskreten Quantity Spaces

Um Algorithmen auf QMVs zu realisieren, insbesondere um eine Repräsentation auf einer höheren Abstraktionsebene aus QMVs erzeugen zu können, braucht man Möglichkeiten, um mit QMVs rechnen zu können. Die Art und Weise, wie dies zu realisieren ist, hängt natürlich von der Art der unterliegenden Quantity Spaces ab. Im Folgenden besprechen wir verschiedene Möglichkeiten des Rechnens mit QMVs in diskreten Quantity Spaces. Wir gehen davon aus, daß die Richtungsdomäne in 4 oder 8 gleichgroße Bereiche unterteilt ist wie in Abbildung 3.3 (I).

In [12] wird eine Lösung für das Problem der Komposition von kombinierter Richtungs/Entfernungsinformation gegeben. Sie basiert auf Intervallarithmetik und ergibt korrekte, aber vage Ergebnisse, nämlich eine Disjunktion aller möglichen Ergebnisse. Diese Art des Schließens funktioniert zwar und ist korrekt, würde jedoch für unsere Anwendung zu einer kombinatorischen Explosion führen. In den Generalisierungsalgorithmen, die in Abschnitt 5.2 vorgestellt werden, müssen möglicherweise sehr viele kurze QMVs miteinander kombiniert werden. Addition von mehreren hundert QMVs zu einem einzigen ist dort keine Seltenheit. Zu der berechnungstechnischen Schwierigkeit, zu der die kombinatorische Explosion führt, kommt noch, daß die vagen Ergebnisse in so einem Fall zwar korrekt, aber völlig unbrauchbar sind, denn am Ende würde oft eine Disjunktion aller Möglichkeiten stehen.

Deswegen entwickeln wir eine Methode, die zwar nicht völlig korrekt ist, aber eindeutige Resultate ergibt. Wir akzeptieren, bei der Berechnung einen gewissen Fehler zu machen, um keine vagen Resultate zu erhalten. Völlig korrekte Resultate sind für unsere Anwendung (Extrahierung der Bewegungsgestalt für die SHAPE-Repräsentation) nicht so wichtig, da es nicht darum geht, korrekte Aussagen über einen möglichen Ort zu inferieren, sondern die Information über die Form der Grobstruktur des Bewegungsverlaufs möglichst zu erhalten.

Um die Resultate eindeutig zu machen, gehen wir von einigen Annahmen aus: Um „glatt“ mit den Distanzen rechnen zu können, müssen wir hier eine Diskretisierung wählen, die es uns erlaubt, größere Distanzen als Vielfache einer kleineren Distanz auszudrücken. Wir definieren also die Intervalle wie folgt: Wenn t_i die Obergrenze eines Distanzintervalls ist, ist t_{i+1} , die Obergrenze des nächsten Intervalls, $n \cdot t_i$.

Dann ist es möglich, jede qualitative Distanz als Vielfaches von **very-close** auszudrücken:

$$\begin{aligned}\text{very-far} &= n \cdot \text{far} \\ &= n^2 \cdot \text{medium-distance} \\ &= n^3 \cdot \text{close} \\ &= n^4 \cdot \text{very-close}\end{aligned}$$

Damit machen wir auch die vereinfachende Annahme, daß **very-far** eine Obergrenze hat. Durch diese Annahmen können wir nun QMVs derselben Richtung addieren (z.B. für $n = 4$): $\langle \text{far north} \rangle^3 + \langle \text{close north} \rangle^5 = \langle \text{close north} \rangle(3 \cdot 4^2 + 5) = \langle \text{close north} \rangle^{53}$.

Weitere vereinfachende Annahmen machen wir bezüglich der Richtung der QMVs, nämlich, daß wir QMVs entgegengesetzter Richtung subtrahieren können, z. B. $\langle \text{far east} \rangle + \langle \text{far west} \rangle = \langle \text{far east} \rangle - \langle \text{far east} \rangle = \langle 0 \ 0 \rangle$.

Damit erhalten wir ein Äquivalenzaxiom: Für alle Distanzen X und entgegengesetzte Richtungen A und B ist

$$\langle X \ A \rangle + \langle X \ B \rangle = \langle X \ A \rangle - \langle X \ A \rangle = \langle 0 \ 0 \rangle$$

Außerdem nehmen wir an, daß die Orientierungen $\{\text{west}, \text{east}\}$ und $\{\text{north}, \text{south}\}$ voneinander unabhängig sind, so daß wir sie jeweils unabhängig addieren oder subtrahieren können.

Beide der im Folgenden beschriebenen Methoden, mit QMVs zu rechnen, basieren auf diesen vereinfachenden Annahmen. Die Berechnungen werden dadurch inkorrekt in dem Sinne, daß nicht jede Instanzierung der qualitativen Variablen dieses Resultat ergeben würde. Zum Beispiel sind die beiden (quantitativen) Vektoren $(80, 90^\circ)^T$ und $(130, -90^\circ)^T$ Instanzierungen von $\langle \text{far north} \rangle$ und $\langle \text{far south} \rangle$, aber $(50, -90^\circ)^T = (80, 90^\circ)^T + (130, -90^\circ)^T$ ist keine Instanzierung von $\langle 0 \ 0 \rangle = \langle \text{far north} \rangle + \langle \text{far south} \rangle$. Das kann natürlich zu Verschiebungen und Verzerrungen der errechneten Summe von zwei QMVs im Vergleich zu der tatsächlichen Summe der instanzierenden Vektoren führen. Experimente mit dem MM-TOOL zeigten allerdings, daß diese Verschiebungen und Verzerrungen für unsere Anwendung nicht böseartig sind. Uns geht es ja darum, die Feinstruktur eines Bewegungsverlaufs zu glätten, um die Grobstruktur, aus der letztendlich die SHAPE-Repräsentation erzeugt werden soll, besser sichtbar zu

machen. Dies funktioniert mit unserer Rechenmethode hinreichend gut, während eine Methode wie in [12], die immer ein korrektes, aber vages Resultat erzeugt, dies wegen der angesprochenen Probleme nicht leisten könnte.

Symbolisches Rechnen

Eine Möglichkeit des Rechnens mit QMVs unter den oben gemachten Annahmen ist nun, einfach rein symbolisch zu manipulieren. Dazu muß man sich nur einige wenige Regeln überlegen. Wir besprechen hier gleich den schwierigeren Fall des Rechnens mit relativen Richtungen. Der Einfachheit halber betrachten wir eine Diskretisierung in vier Richtungen.

Da die Richtungen hier immer relativ zur letzten Bewegungsrichtung ausgedrückt sind, kann man immer nur zwei aufeinanderfolgende QMVs zusammenfassen. Die Domäne der Richtungen ist eine Gruppe mit **forward** als neutralem Element (invers zu sich selbst), **backward** ebenfalls invers zu sich selbst, und **left** und **right** invers zueinander. Die Komposition von Richtungen allein ist jedoch uninteressant, da bei der Kombination von Vektoren das Zusammenspiel von Richtung und Entfernung für das Ergebnis entscheidend ist.

Wir können hier verschiedene Fälle unterscheiden, wobei meistens lediglich die Richtung des zweiten QMV relevant ist, da sie ja relativ zur Richtung des ersten ausgedrückt wird. Die Aufzählung unten beschreibt die Additionsregeln für symbolisches Rechnen mit QMVs mit 4 verschiedenen Richtungen. Das Prinzip für 8 Richtungen ist analog.

1. $\langle \text{dist1 dir1 speed1} \rangle + \langle \text{dist2 forward speed2} \rangle = \langle \text{dist1+dist2 dir1 average-speed} \rangle$
2. $\langle \text{dist1 dir1 speed1} \rangle + \langle \text{dist2 backward speed2} \rangle = \langle |\text{dist1-dist2}| \text{ erg-dir erg-speed} \rangle$.
Wenn $\text{dist1} - \text{dist2} > 0$, $\text{erg-dir} = \text{dist1}$; sonst $\text{erg-dir} = -\text{dist1}$.
 $\text{erg-speed} = |\text{dist1-dist2}|$ geteilt durch die Summe der Zeiteinheiten.
3. Die Richtung des zweiten Vektors ist **left** oder **right**. Wenn wir nur in 4 Richtungsgebiete diskretisiert haben, können wir keine „Zwischenrichtung“ berechnen. Deswegen müssen wir eine Absorptionsregel benutzen — das heißt, der (entfernungsmäßig) größere Vektor „saugt“ den kleineren auf. Das erscheint plausibel, wenn wir uns vergegenwärtigen, daß in dem relativen Bezugssystem die „Diskretisierungsscheibe“ jedesmal neu ausgerichtet wird, so daß „vorwärts“ immer genau in der Mitte des Intervalls **forward** liegt.

Die resultierende Richtung muß natürlich wieder relativ zur Richtung des letzten Vektors ausgedrückt werden. Wir müssen deswegen beachten, daß sich die „absolute“ Richtung bei einer Komposition so ändern kann, daß die relative Richtung

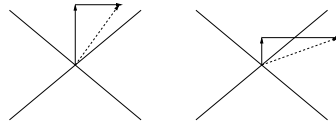


Abbildung 4.2: Absorptionsregel

praktisch umgedreht wird. Es gibt da die folgenden Möglichkeiten (mit $\text{dir} \in \{\text{left}, \text{right}\}$):

- $\langle \text{dist1 dir speed1} \rangle + \langle \text{dist2 dir speed2} \rangle =$
 $\langle \text{dist2 backward speed2} \rangle$, wenn $\text{dist2} > \text{dist1}$;
 $\langle \text{dist1 dir speed1} \rangle$ sonst.
- $\langle \text{dist1 dir speed1} \rangle + \langle \text{dist2 -dir speed2} \rangle =$
 $\langle \text{dist2 forward speed2} \rangle$, wenn $\text{dist2} > \text{dist1}$;
 $\langle \text{dist1 dir speed1} \rangle$ sonst.
- $\langle \text{dist1 forward speed1} \rangle + \langle \text{dist2 dir speed2} \rangle =$
 $\langle \text{dist1 forward speed1} \rangle$, wenn $\text{dist1} > \text{dist2}$;
 $\langle \text{dist2 dir speed2} \rangle$ sonst.
- $\langle \text{dist1 backward speed1} \rangle + \langle \text{dist2 dir speed2} \rangle =$
 $\langle \text{dist2 -dir speed2} \rangle =$ wenn $\text{dist2} > \text{dist1}$;
 $\langle \text{dist1 backward speed1} \rangle$ sonst.

Die Absorptionsregel ist natürlich problematisch, weil sie eine systematische Verschiebung des Rests der QMV-Sequenz bewirkt. Wenn man 8 Richtungen benutzt, ist der Fehler kleiner, da mehr Richtungen korrekt kombiniert werden können (z. B. auch **forward** und **left** zu **halfleft**) und die Verschiebung im Falle einer Absorption auch nicht so groß ist. Trotzdem muß die Absorptionsregel auch hier angewandt werden, wenn es beispielsweise gilt, **halfright** und **right** zu addieren.

Algorithmen, die diese Rechenmethode benutzen, sollten deswegen darauf achten, die Absorptionsregel so selten wie möglich zu benutzen. Sinnvollerweise sollten hier zunächst die Vektoren addiert werden, die sich am leichtesten komponieren lassen. Das sind zunächst Vektoren mit derselben Richtung (d. h. der zweite Vektor hat die Richtung **forward**), und dann Vektoren mit entgegengesetzten Richtungen (d. h. der zweite Vektor hat die Richtung **backward**). Im Falle von 8 Richtungen sollten als nächstes die orthogonalen Vektoren behandelt werden (d. h. der zweite Vektor hat die Richtung **left** oder **right**), und erst ganz zum Schluß die Vektoren, bei denen die Absorptionsregel Anwendung findet.

Im Fall zwei und drei der oben beschriebenen QMV-Additionen kann es dazu kommen, daß der resultierende Vektor eine Richtung hat, die – in einem absoluten Referenzsystem

betrachtet – von der des zweiten Summanden abweicht (z. B. wenn ein **backward**-Vektor auf einen **forward**-Vektor addiert wird, wobei der erstere länger ist als der zweite). Dann muß der folgende Vektor, dessen Richtung ja relativ zum zweiten Summanden angegeben war, so verändert werden, daß die Richtung nun relativ zum Ergebnis der Addition angegeben wird.

Seien $A = \langle \text{dist1 dir1 speed1} \rangle$, $B = \langle \text{dist2 dir2 speed2} \rangle$ und $C = \langle \text{dist3 dir3 speed3} \rangle$ drei aufeinanderfolgende Vektoren in einer QMV-Sequenz, und A und B werden addiert. Wenn der resultierende Vektor dieselbe Richtung hat wie B , muß mit C nichts weiter gemacht werden. Wenn eine andere Richtung herauskommt, muß die Richtung von C dementsprechend angepaßt werden.

Dies kann der Fall sein, wenn:

1. die Addition eines längeren **backward**-Vektors die relative Richtung invertiert ($A + B = \langle |\text{dist1}-\text{dist2}| \text{-dir1 erg-speed} \rangle$): Dann muß die Richtung von C ebenfalls invertiert werden und C wird zu $\langle \text{dist3 -dir3 speed3} \rangle$.
2. durch die Absorptionsregel A als Resultat der Addition herauskommt. Dann müssen die Richtungen von B und C komponiert werden, um C wieder korrekt anzupassen. Da qualitative Richtungen eine Gruppe bilden, funktioniert dies folgendermaßen: ($\text{dir} \in \{\text{left, right}\}$):
 - $\text{dir} \circ \text{-dir} = \text{forward}$
 - $\text{dir} \circ \text{dir} = \text{backward}$
 - $\text{dir} \circ \text{forward} = \text{dir}$
 - $\text{dir} \circ \text{backward} = \text{-dir}$
3. A und B gleichlange, entgegengesetzte Vektoren sind und das Resultat der Addition somit $\langle \text{zero zero zero} \rangle$ beträgt. Dann muß die Richtung von C relativ zur Richtung des Vektors, der vor A steht⁴, wiedergegeben werden. Dies erreicht man durch Komposition der Richtungen von A , B und C .

Wenn die Richtungen nicht in einem relativen Bezugssystem angegeben sind wie hier, sondern in einem absoluten, wird die Sache natürlich einfacher, da beliebige Vektoren addiert werden können und nicht nur aufeinanderfolgende. Außerdem muß keine Neuausrichtung von nachfolgenden Vektoren vorgenommen werden, da die Richtungen ja absolut sind.

Deswegen ist es sinnvoll, das Rechnen im relativen Bezugssystem zu vermeiden und für Berechnungen die QMV-Sequenz, wenn möglich, in ein beliebiges absolutes Bezugssystem umzusetzen (indem die absolute Richtung des ersten QMV gesetzt wird und die

⁴Falls da keiner steht, ist C der erste Vektor mit einer echten Richtung in der Sequenz. Diese ist immer **forward**.

absolute Richtung der anderen Vektoren aus den dazu relativen berechnet wird). Dann steht das Instrument der im folgenden beschriebenen QMV-Algebra zur Verfügung, in der genauer gerechnet werden kann, weil es keine Absorptionsregel gibt. In Kapitel 6 besprechen wir noch ausführlich, wann diese Umrechnung zwischen unterschiedlichen Referenzsystemen problematisch sein kann und wann sie unproblematisch ist.

Eine QMV-Algebra

Wenn wir uns der zu Anfang des Abschnitts 4.1.4 gemachten Annahmen entsinnen, die dem Rechnen mit QMVs zugrunde liegen, stellen wir fest, daß die Vektoren $e^1 = \langle \text{very-close east} \rangle$ und $e^2 = \langle \text{very-close south} \rangle$ linear unabhängig sind und deswegen die Basis eines zweidimensionalen Vektorraums bilden (siehe dazu auch [82]). Wenn wir also die QMVs in Vektoren zu dieser Basis umrechnen, können wir wie aus der linearen Algebra bekannt mit ihnen rechnen. Z.B. ergibt

$$\langle \text{close east} \rangle^1 = \begin{pmatrix} n \\ 0 \end{pmatrix}$$

und

$$\langle \text{very-far north} \rangle^2 = \begin{pmatrix} 0 \\ -2 \cdot n^4 \end{pmatrix}.$$

Die QMVs zeichnen sich dadurch aus, daß eine Komponente der Vektoren immer Null ist. Daraus folgt, daß nicht jedes Mitglied dieses Vektorraums als einzelner QMV ausgedrückt werden kann. Wenn man in diesem Vektorraum rechnet, erhält man natürlich laufend solche „Mischvektoren“ als Ergebnisse, z. B.

$$\langle \text{very-close east} \rangle^7 + \langle \text{very-close north} \rangle^5 = \begin{pmatrix} 7 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ -5 \end{pmatrix} = \begin{pmatrix} 7 \\ -5 \end{pmatrix}.$$

Diese Vektoren müssen am Ende der Berechnungen wieder in die QMV-Repräsentation zurücktransformiert werden. Dies geschieht mit dem selben Verfahren, das numerische Vektoren auf QMVs abbildet.

In unserem Beispiel:

$$\begin{pmatrix} 7 \\ -5 \end{pmatrix} \equiv \langle \text{very-close northeast} \rangle^9,$$

oder, wenn nur 4 Richtungen unterschieden werden, $\langle \text{very-close east} \rangle^9$.

4.2 Fuzzy-Modellierung von Bewegungsverläufen

Ein großes Problem bei diskreten Intervallen ist, daß Bewegungsverläufe, die sich nur minimal unterscheiden, sehr unterschiedlich repräsentiert werden, wenn die numerischen Werte immer in der Nähe der Intervallgrenzen liegen, aber einmal auf der einen und das andere mal auf der anderen Seite der Grenze. Ein Beispiel für die Richtungsdomäne ist in Abbildung 4.3 gegeben. Dieses Verhalten läuft natürlich konträr zu dem Anspruch, daß Ähnliches auch ähnlich repräsentiert werden sollte.

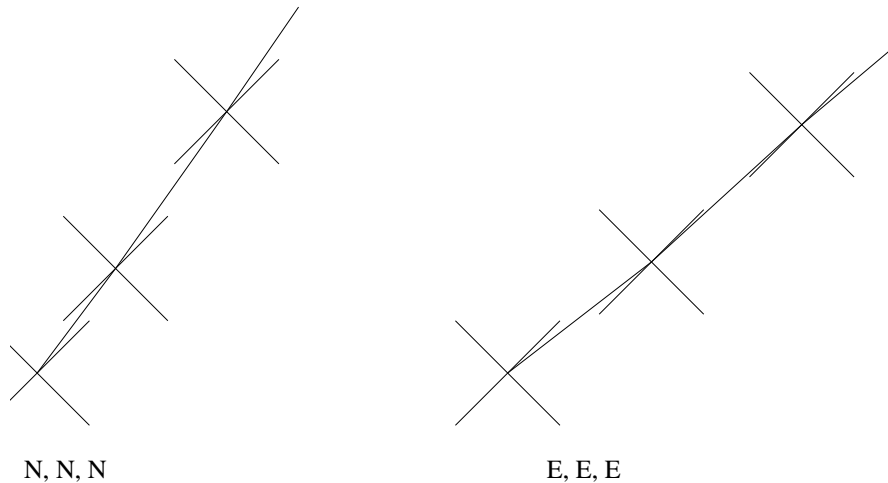


Abbildung 4.3: Ähnliche Bewegungsverläufe werden unähnlich repräsentiert

Um dies zu beheben, kann man entweder die Quantity Spaces oder die Bewegungsvektoren als Fuzzy-Mengen modellieren. Während klassische Mengen durch charakteristische Funktionen ausgezeichnet sind, die nur zwei Werte annehmen können (1 oder 0, d. h. ein Element gehört zu einer Menge oder nicht), wird bei einer Fuzzy-Menge diese Zweierwertigkeit aufgegeben (siehe [9], S. 5):

Definition 4.1

Sei G eine (stets klassische) Grundmenge, μ_A eine (stets klassische) Funktion der Grundmenge G in das Einheitsintervall $[0, 1]$ der reellen Achse:

$$\mu_A : G \rightarrow [0, 1].$$

Dann heißt die Menge A aller Paare $(x, \mu_A(x))$

$$A = \{(x, \mu_A(x)) | x \in G\}$$

eine Fuzzy-Menge (unscharfe Menge, fuzzy set) über G .

Das Rechnen mit Fuzzy-Mengen bringt in aller Regel mit sich, daß die betreffenden Mengen ständig unschärfer werden. Das bedeutet, daß entweder die klassische Grundmenge, die auf das Intervall $[0, 1]$ abgebildet wird, größer wird, oder die Zugehörigkeitswerte $\mu_A(x)$ kleiner werden, oder beides. Das Resultat einer Operation mit zwei Fuzzy-Mengen ist am Ende meist eine völlig neue Menge mit anderer Grundmenge und/oder anderer Zugehörigkeitsfunktion. Repräsentiert man einen Quantity Space als Fuzzy-Menge und definiert z. B. eine Addition auf dieser Fuzzy-Menge, so fällt das Resultat in aller Regel nicht eine andere, vordefinierte Fuzzy-Menge aus der Menge der Quantity Spaces, sondern wird eine neue Fuzzy-Menge sein. Die Quantity Spaces wären also nicht abgeschlossen gegen die Addition, es sei denn, man sucht nach jedem Rechenschritt die Fuzzy-Menge aus den vordefinierten Quantity Spaces heraus, die dem Resultat am nächsten kommt. Dies ist aber vergleichsweise teuer, und aus Gründen der Vergleichbarkeit von Bewegungsverläufen wollen wir andererseits auch nicht, daß sich die Repräsentationen auf unterschiedliche Fuzzy-Mengen beziehen, sondern natürlich immer auf dieselben, vordefinierten. Deswegen haben wir uns dafür entschieden, bei einer scharfen Diskretisierung des Raumes zu bleiben, die Bewegungsvektoren unscharf zu modellieren und diese in die gewählte Diskretisierung abzubilden, wenn nötig und sinnvoll.

Diese Lösung geht also von einer scharfen und diskreten Aufteilung des Raumes aus wie in Abschnitt 4.1 beschrieben. Bei Kombination von Richtungs- und Distanzintervallen wird der Raum in diskrete Flächen mit scharfen Grenzen parkettiert (siehe Abbildung 4.4).

Jeder zweidimensionale (klassische) Vektor fällt in genau eine dieser Flächen. Dies entspricht der Umwandlung in einen QMV mit diskreten Intervallen wie in Abschnitt 4.1 beschrieben.

4.2.1 Fuzzy Numerical Vectors

Wie kann man nun vermeiden, daß für Vektoren, die nahe irgendwelcher Intervallgrenzen liegen, eine entweder-oder-Entscheidung für ein bestimmtes Intervall getroffen werden muß und man Gefahr läuft, ähnliche, um die Intervallgrenzen verlaufende Bewegungsverläufe, unähnlich zu repräsentieren? Dazu setzt man bei der Modellierung der Bewegungsvektoren an.

Mit Meßfehlern behaftete numerische Daten lassen sich als Fuzzy-Zahlen modellieren:

Definition 4.2

Eine Fuzzy-Zahl A ist eine konvexe Fuzzy-Menge auf der Grundmenge der reellen Zahlen \mathbb{R} , die folgende Eigenschaften erfüllt (siehe [9], S. 124):

1. $\mu_A(x) = 1$ für genau ein $x \in \mathbb{R}$
2. μ_A ist stückweise stetig.

Fuzzy-Zahlen vom Typ LR sind besonders einfach zu handhaben.

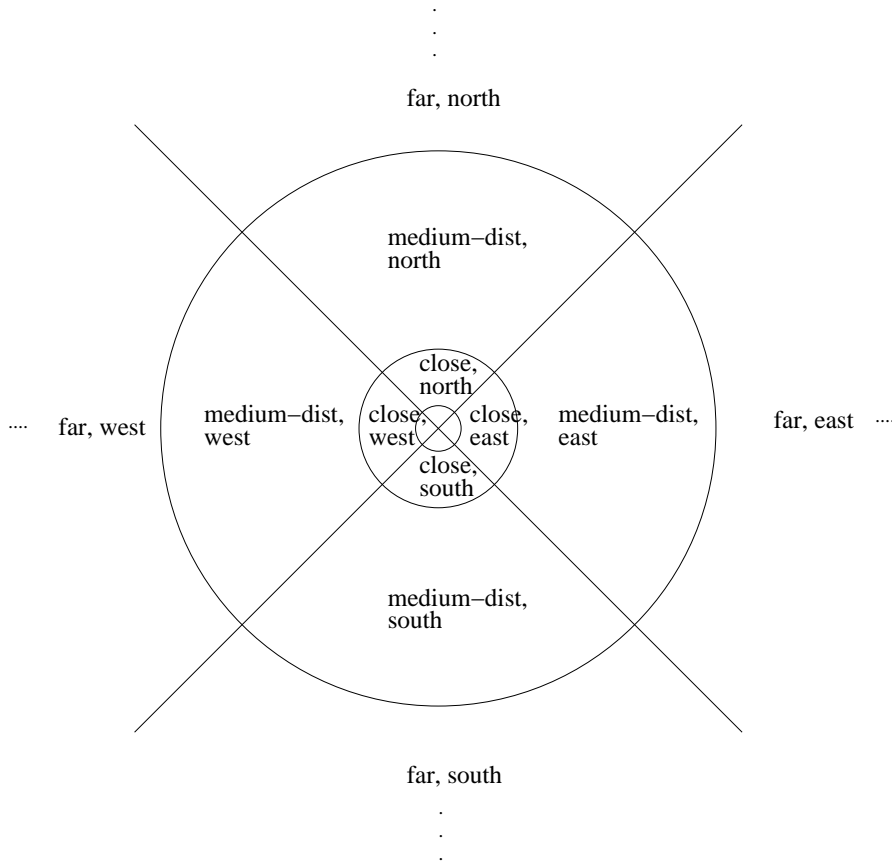


Abbildung 4.4: Kombination von Richtungs- und Distanzintervallen

Definition 4.3

Eine Fuzzy-Zahl heißt vom Typ *LR*, wenn es Referenzfunktionen *L*, *R* mit folgenden Eigenschaften gibt:

1. $L, R : \mathbb{R}_0^+ \rightarrow [0, 1]$
2. $L(0) = R(0) = 1$
3. *L* und *R* sind monoton fallend.
4.
$$\mu_A(x) = \begin{cases} L(\frac{m-x}{\alpha}) & x \leq m, \\ R(\frac{x-m}{\beta}) & x > m. \end{cases}$$
 $\alpha > 0$ ist die linke Spannweite, $\beta > 0$ ist die rechte Spannweite, *m* der Gipfelwert von *A*. Man schreibt $A = (m, \alpha, \beta)_{LR}$.

Der Einfachheit halber benutzen wir im Weiteren Fuzzy-Zahlen mit $L = R = \max(0, 1-x)$. Dann können wir den Index LR auch weglassen. Außerdem modellieren wir den Gipfelwert immer mittig, d. h. linke und rechte Spannweite sind gleich ($\alpha = \beta$). Zur besseren Übersicht schreiben wir deswegen die Spannweite mit dem gleichen Buchstaben wie den Gipfelwert, nur mit einem Punkt: $A = (m, \dot{m}, \dot{m})$.

Ein Fuzzy-Bewegungsvektor im zweidimensionalen Raum⁵ soll nun modelliert werden als eine Funktion $\mu : \mathbb{R} \times \mathbb{R} \rightarrow [0, 1]$. Dazu bieten sich verschiedene Möglichkeiten an:

1. Die Grundmenge $\mathbb{R} \times \mathbb{R}$ bildet eine Ellipse mit den Spannweiten der beiden Fuzzy-Koordinaten als Durchmesser; $\mu : \mathbb{R} \times \mathbb{R} \rightarrow [0, 1]$ ist dann ein Kegel über dieser Ellipse.
2. Die Grundmenge $\mathbb{R} \times \mathbb{R}$ bildet ein Rechteck mit den Spannweiten der beiden Fuzzy-Koordinaten als Seitenlängen; $\mu : \mathbb{R} \times \mathbb{R} \rightarrow [0, 1]$ ist dann ein Kegel über diesem Rechteck (Pyramide).
3. Die Grundmenge $\mathbb{R} \times \mathbb{R}$ bildet ein Kreis-/Ringsegment mit den Spannweiten der beiden Fuzzy-Koordinaten als Winkel und Radius (bzw. Breite); $\mu : \mathbb{R} \times \mathbb{R} \rightarrow [0, 1]$ ist dann ein Kegel über diesem Kreis- bzw. Ringsegment (siehe Abbildung 4.5 für eine Darstellung dieser Grundmenge).

Betrachtet man die Diskretisierung der Ebene in Abbildung 4.4, wird man feststellen, daß die letzte Möglichkeit die natürlichste ist, wenn man Fuzzy-Vektoren in diese Parkettierung abbilden will: Dazu ist es nötig, den Fuzzy-Vektor mit der gegebenen Parkettierung zu schneiden und die Zugehörigkeiten zu den jeweiligen Kacheln zu berechnen. Modelliert man die Fuzzy-Vektoren als Kegel über einem Ringsegment, wobei der Ring seinen Mittelpunkt in der Mitte der Parkettierung hat (also am Fuße des Vektors), so erhält man bei Schnitten mit dieser Parkettierung als Grundflächen wiederum lauter Ringsegmente, deren anteilige Volumina leicht zu berechnen sind.

Definition 4.4

Sei $(r, \phi)^T$ ein numerischer Bewegungsvektor. Der Vektor $(R, \Phi)^T$ (R und Φ Fuzzy-Zahlen mit den Gipfelwerten r und ϕ) heißt zu $(r, \phi)^T$ gehöriger Fuzzy Numerical Vector (FNV).

Es gibt unendlich viele zu $(r, \phi)^T$ gehörige FNVs mit unterschiedlichen Spannweiten.

Definition 4.5

Sei $A = (R, \Phi)$ ein FNV mit $R = (r, \dot{r}, \dot{r})$ und $\Phi = (\phi, \dot{\phi}, \dot{\phi})$ vom Typ LR mit $L = R = \max(0, 1-x)$. Dann ist

$$\mu_A(x, \alpha) = \mu_R(x)\mu_\Phi(\alpha).$$

⁵Der dreidimensionale Fall ist analog.

4.2.2 Rechnen mit FNVs

Die Addition auf zwei Fuzzy-Zahlen vom gleichen Typ LR mit linker gleich rechter Spannweite ist folgendermaßen definiert (vgl. [9], S. 127):

Definition 4.6

Seien $A = (m, \dot{m}, \ddot{m})_{LR}$ und $B = (n, \dot{n}, \ddot{n})_{LR}$ Fuzzy-Zahlen vom gleichen Typ LR . Dann heie die fuzzy Zahl $A \oplus B$ gem

$$A \oplus B = (m, \dot{m}, \ddot{m})_{LR} \oplus (n, \dot{n}, \ddot{n})_{LR} = (m + n, \dot{m} + \dot{n}, \ddot{m} + \ddot{n})_{LR}$$

die Summe der Fuzzy-Zahlen A und B .

Die Vektoraddition auf FNVs kann kanonisch definiert werden:

Definition 4.7

Seien

$$A = \begin{pmatrix} X_1 \\ Y_1 \end{pmatrix} \text{ und } B = \begin{pmatrix} X_2 \\ Y_2 \end{pmatrix}$$

FNVs mit X_i, Y_i Fuzzy-Zahlen vom gleichen Typ LR . Dann heit der FNV

$$A \oplus B = \begin{pmatrix} X_1 \oplus X_2 \\ Y_1 \oplus Y_2 \end{pmatrix}$$

die Summe der FNVs A und B .

Da unsere FNVs allerdings aufgrund der Modellierung in Polarkoordinaten vorliegen, mu fur die Addition ein bichen mehr Aufwand getrieben werden.

Fur die Umrechnung von Polarkoordinaten in kartesische Koordinaten gilt ja bekanntlich:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r \cdot \cos \phi \\ r \cdot \sin \phi \end{pmatrix}$$

Fur die Addition der Gipfelwerte zweier FNVs $\begin{pmatrix} r \\ \phi \end{pmatrix}, \begin{pmatrix} r' \\ \phi' \end{pmatrix}$ gilt also:

$$\begin{pmatrix} x'' \\ y'' \end{pmatrix} = \begin{pmatrix} r \cdot \cos \phi + r' \cdot \cos \phi' \\ r \cdot \sin \phi + r' \cdot \sin \phi' \end{pmatrix}.$$

Sobald nichts mehr addiert werden soll, werden die Gipfelwerte wieder in Polarkoordinaten zurcktransformiert:

$$\begin{pmatrix} r'' \\ \phi'' \end{pmatrix} = \begin{pmatrix} \sqrt{x''^2 + y''^2} \\ \arctan \frac{y''}{x''} \end{pmatrix}$$

Die Spannweiten müssen anschliessend gesondert berechnet werden; sie addieren sich einfach.

Definition 4.8

Seien

$$A = \begin{pmatrix} (r_1, \dot{r}_1, r_1) \\ (\phi_1, \dot{\phi}_1, \phi_1) \end{pmatrix} \text{ und } B = \begin{pmatrix} (r_2, \dot{r}_2, r_2) \\ (\phi_2, \dot{\phi}_2, \phi_2) \end{pmatrix}$$

FNVs in Polarkoordinaten mit Fuzzy-Zahlen vom gleichen Typ LR als Komponenten. Dann heit der FNV

$$A \oplus B = \begin{pmatrix} (\sqrt{(r_1 \cdot \cos \phi_1 + r_2 \cdot \cos \phi_2)^2 + (r_1 \cdot \sin \phi_1 + r_2 \cdot \sin \phi_2)^2}, \dot{r}_1 + \dot{r}_2, r_1 + r_2) \\ (\arctan \frac{r_1 \cdot \sin \phi_1 + r_2 \cdot \sin \phi_2}{r_1 \cdot \cos \phi_1 + r_2 \cdot \cos \phi_2}, \dot{\phi}_1 + \dot{\phi}_2, \phi_1 + \phi_2) \end{pmatrix}$$

die Summe der FNVs A und B.

4.2.3 Fuzzy Motion Vectors

Die FNVs werden durch wiederholte Addition natrlich immer unschrfer, d. h. die Spannweiten werden jedesmal grer. Das bedeutet, da ein FNV einen recht groen Bereich der Ebene abdecken kann, in dem die Zugehrigkeitsfunktion Werte grer 0 annimmt. Dieser Bereich mitsamt seinen Fuzzy-Zugehrigkeitsfunktionen mu nun in die in Abbildung 4.4 dargestellte Diskretisierung der Ebene abgebildet werden.

Die Fuzzy-Zugehrigkeitsfunktion eines FNV $A \mu_A(r, \phi)$ ergibt einen Kegel ber einem Kreis- bzw. Ringsegment. Dieser Kegel fllt mglicherweise in verschiedene Bereiche der in Abbildung 4.4 dargestellte Diskretisierung. Wieviel von der Kegel anteilig in welchem Bereich dieser Diskretisierung liegt, gibt nun genau der FMV (Fuzzy Motion Vector) an.

Definition 4.9

Gegeben sei eine Diskretisierung des Raums in n Distanzbereiche und m Richtungsbe-
reiche. Ein FMV F auf dieser Diskretisierung ist ein $n \cdot m$ -Tupel mit Komponenten aus dem Intervall $[0, 1] \in \mathbb{R}$:

$$F = (\mu_{(Dist_1, Dir_1)}, \dots, \mu_{(Dist_1, Dir_m)}, \dots, \mu_{(Dist_n, Dir_1)}, \dots, \mu_{(Dist_n, Dir_m)})^T$$

Jede Komponente des FMV steht fr eine Kachel der Parkettierung, wie in Abbildung 4.4 zu erkennen.

4.2.4 Abbildung FNV in FMV

Um einen FNV A in einen FMV $A_{n \cdot m}$ abzubilden, mu man feststellen, in welchen Teilen der diskreten Parkettierung berhaupt Funktionswerte grer Null angenommen

werden und wieviel Anteil vom Volumen des durch $\mu_A(r, \phi)$ definierten Kegels in jeden dieser Teile fällt. Dazu muß $\mu_A(r, \phi)$ mit den Kreisen und Geraden (bzw. Zylindern und Ebenen), die die Distanz- und Richtungsbereiche begrenzen, geschnitten werden. Daraus ergeben sich Teilvolumina, deren Anteile am Gesamtvolumen des FNV A wiederum den Zugehörigkeitsgrad des FNV zu dem jeweiligen Richtungs-/Distanzbereich ergeben.

Das Gesamtvolumen des Kegels $\mu_A(r, \phi)$ ist relativ einfach zu berechnen. Im Folgenden rechnen wir oBdA mit einem FNV $A = \begin{pmatrix} (0, \dot{r}, \dot{r}) \\ (0, \dot{\phi}, \dot{\phi}) \end{pmatrix}$. Dann ist

$$\mu_A(r, \phi) = \left(1 - \frac{\phi}{\dot{\phi}}\right) \left(1 - \frac{r}{\dot{r}}\right).$$

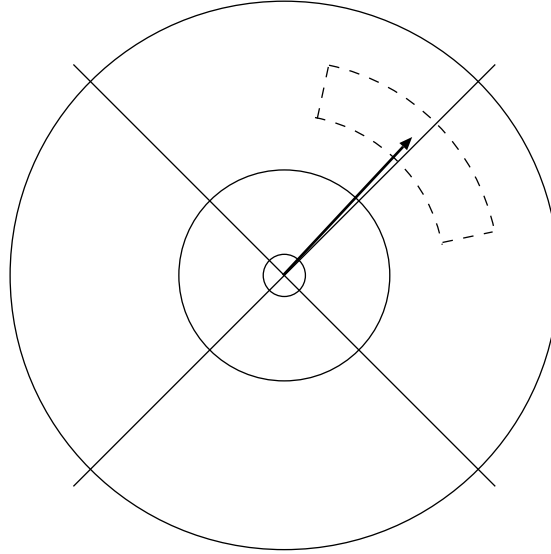
Es gilt:

$$\begin{aligned} V &= \int_{-\dot{r}}^{\dot{r}} \int_{-\dot{\phi}}^{\dot{\phi}} \left(1 - \frac{\phi}{\dot{\phi}}\right) \left(1 - \frac{r}{\dot{r}}\right) d\phi dr \\ &= \frac{1}{\dot{\phi}\dot{r}} \int_{-\dot{r}}^{\dot{r}} \int_{-\dot{\phi}}^{\dot{\phi}} (\dot{\phi} - \phi)(\dot{r} - r) d\phi dr \\ &= \frac{1}{\dot{\phi}\dot{r}} \left(\int_{-\dot{r}}^{\dot{r}} \int_{-\dot{\phi}}^{\dot{\phi}} \dot{\phi}\dot{r} d\phi dr + \int_{-\dot{r}}^{\dot{r}} \int_{-\dot{\phi}}^{\dot{\phi}} \dot{\phi}r d\phi dr + \int_{-\dot{r}}^{\dot{r}} \int_{-\dot{\phi}}^{\dot{\phi}} \phi\dot{r} d\phi dr + \int_{-\dot{r}}^{\dot{r}} \int_{-\dot{\phi}}^{\dot{\phi}} \phi r d\phi dr \right) \end{aligned}$$

Dies sind alles Integrale linearer Funktionen. Durch die konstanten Integrationsgrenzen sind sie relativ einfach zu berechnen. Um die Teilvolumina zu bestimmen, die auf die jeweiligen Kacheln der Parkettierung entfallen, muß der Kegel des FNV mit den Grenzen der Parkettierung geschnitten werden. Die Mitte des Richtungskreuzes und der Mittelpunkt der konzentrischen Distanzkreise fällt dabei auf den Fuß des FNV. Da die Kacheln der Parkettierung die gleiche Gestalt haben wie die Grundfläche des FNV (Kreis- oder Ringsegment, wobei der Kreismittelpunkt am Fuß des FNV liegt), ist diese Operation abgeschlossen: Als Ergebnis erhält man wiederum jeweils ein Kreis- oder Ringsegment (siehe Abbildung 4.5).

Es müssen zwei Arten von Schnitten berechnet werden:

1. Schneiden mit einer Richtungsbegrenzung (bei 4 Richtungen die Geraden $x = y$ bzw $x = -y$): dabei ändern sich die die Grenzen des Integrals $\int_{-\dot{r}}^{\dot{r}} dr$ nicht.
2. Schneiden mit einem Kreisbogen: dabei ändern sich die die Grenzen des Integrals $\int_{-\dot{\phi}}^{\dot{\phi}} d\phi$ nicht.

Abbildung 4.5: Abbildung $FNV \rightarrow FMV$

Ansonsten müssen bei o.a. Volumenberechnung lediglich die Integrationsgrenzen geändert werden!

Das auf jede Kachel $(Dist, Dir)$ entfallende Teilvolumen $TV_{(Dist,Dir)}$ in Relation zum Gesamtvolumen V des FNV ergibt den Wert der Mitgliedschaftsfunktion $\mu_{(Dist,Dir)}$:

$$\mu_{(Dist_i,Dir_j)} = \frac{TV_{(Dist_i,Dir_j)}}{V}, 1 \leq i \leq n, 1 \leq j \leq m.$$

Der Gipfelpunkt des in Abbildung 4.5 dargestellten FNV fällt in die Kachel $(medium-dist, north)$. Der Gipfelpunkt eines nur minimal nach rechts abweichenden Vektors gleicher Länge fiel in die Kachel $(medium-dist, east)$. Als QMVs würden diese beiden Vektoren also mit völlig unterschiedlichen Richtungen repräsentiert werden, obwohl sie sich kaum unterscheiden. Repräsentiert man diese Vektoren jedoch als FMV, so ist das Ergebnis in dieser Beziehung schon erfreulicher: es wären alle $\mu_{(Dist_i,Dir_j)} = 0$, außer $\mu_{(medium-dist,north)}$ und $\mu_{(medium-dist,east)}$, die jeweils ungefähr den Wert 0,5 annehmen würden. Vergleicht man die resultierenden FMVs miteinander, so stellt man fest, daß sie ganz ähnlich sind, während man dies bei den QMVs nicht sagen kann.

Die Abbildung von FNV in FMV ist zwar nicht unbedingt trivial. Allerdings muß sie, wie wir noch sehen werden, für die im Folgenden vorgestellten Algorithmen nicht in jedem Rechenschritt stattfinden, sondern nur, wenn Vergleiche oder Ausgaben gemacht werden müssen. Solange kann man mit den FNVs rechnen, was denkbar einfach ist.

4.2.5 Abbildung FMV in QMV

So erhaltene FMV-Sequenzen sind zwar hervorragend vergleichbar, aber für einen naiven Benutzer eines technischen Systems ebenso unhandlich wie jede andere numerische Repräsentation, wenn nicht sogar noch unintuitiver. Für die Benutzerinteraktion ist diese Repräsentation also ungeeignet und muß für die Ausgabe an die Benutzungsschnittstelle in einen QMV umgewandelt werden. Dies erreicht man einfach mit einer Maximumfunktion über FMVs:

Definition 4.10

Sei A_c ein $n \cdot m$ -FMV. Dann gilt:

$$\begin{aligned} \max(A_c) = \\ \max(\mu_{(Dist_1, Dir_1)}, \dots, \mu_{(Dist_1, Dir_m)}, \dots, \mu_{(Dist_n, Dir_1)}, \dots, \mu_{(Dist_n, Dir_m)})^T = \\ (Dist_x, Dir_y) \text{ mit } \mu_{(Dist_x, Dir_y)} > \mu_{(Dist_i, Dir_j)} \forall (i, j) \in (n \times m) (i, j \neq x, y). \end{aligned}$$

4.2.6 Abbildung QMV in FNV

Umgekehrt möchte man diese Repräsentation auch bei Benutzereingaben vermeiden. Deswegen muß es auch eine Möglichkeit geben, QMVs in FNVs umzuwandeln. Dabei kann natürlich nicht irgendeine Meßgenauigkeit als Maßstab für die Spannweite der Fuzzy-Zahlen dienen. Man weiß allerdings genau, welche Kachel der Parkettierung der QMV $\langle Dist_i, Dir_j \rangle$ repräsentiert, nämlich natürlich die Kachel $(Dist_i, Dir_j)$. Deswegen kann man bei der Umwandlung eines solchen QMV als Gipfelpunkt die Mitte der jeweiligen Intervalle benutzen.

Wie man die Spannweiten definiert, ist eine Frage der Anwendung. Wenn man jeweils genau die Hälfte der betreffenden Intervallbreite als Spannweite nimmt, erhält man natürlich als Grundfläche des FNV genau die Kachel $(Dist_i, Dir_j)$ der Parkettierung. Das bedeutet aber auch, daß, wenn man den so erhaltenen FNV in einen FMV umwandelt, alle $\mu_{(Dist_k, Dir_l)}$ mit $k \neq i$ und $l \neq j$ Null werden und $\mu_{(Dist_i, Dir_j)} = 1$. Wenn man QMV-Sequenzen vergleichen will und die oben beschriebenen negativen Effekte vermeiden, ist dies sicher nicht geschickt, wohingegen diese Abbildung gut gewählt ist, wenn man mit den FNVs weiterrechnen will. Für den reinen Vergleich ist eine überlappende Modellierung, wie in Abbildung 4.6 für die Richtungsdomäne dargestellt, sicher besser, da dann auch Anteile des FNV in die benachbarten Kacheln fallen.

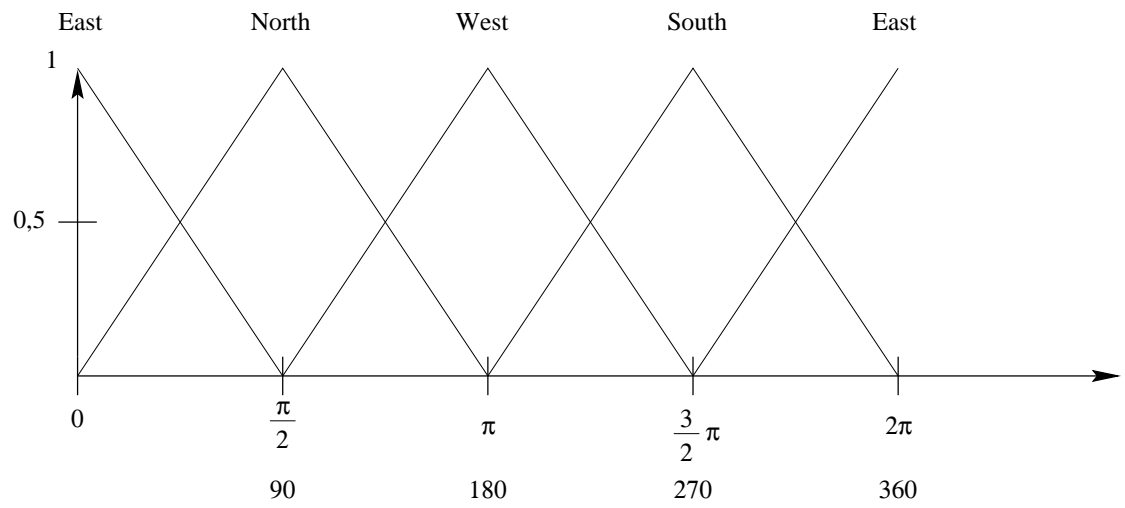


Abbildung 4.6: Fuzzy-Richtungen

5 Konstruktion der qualitativen Repräsentation

5.1 Von numerischen Daten zu QMVs

Nachdem nun geklärt ist, wie unsere qualitative Repräsentation aussieht und wie diskretisiert werden soll, stellt sich die Frage, wie die Umwandlung von beobachteten numerischen Werten in qualitative Werte stattfinden soll.

Eine Möglichkeit ist die direkte Abbildung der Segmente des in Abschnitt 3.2 eingeführten Polygonzugs nach jedem Abtastschritt auf die qualitativen Intervalle. Beim Rechnen mit QMVs muß man sich allerdings im Klaren sein, daß man, wie in Abschnitt 4.1.4 besprochen, möglicherweise große Fehler macht. Es ist deswegen besser, dies, soweit möglich, zu vermeiden.

Bei der Konstruktion der qualitativen Repräsentation wollen wir letztendlich Informationen über die Form, die ein Bewegungsverlauf beschreibt, extrahieren. Deswegen scheint es sinnvoll, dafür irrelevante Details schon auf der Ebene der numerischen Repräsentation wegzufiltern. So erhält man einen Bewegungsverlauf, der nur noch die größeren Änderungen der Bewegungsrichtung enthält. Hat man numerische Daten zur Verfügung, die mit großen Meßfehlern behaftet sind, ist es sinnvoll, diese als FNVs zu modellieren und mit FNVs zu rechnen. Erst der vereinfachte Bewegungsverlauf wird anschließend in FMVs (zum Vergleichen) und QMVs (für die Benutzerinteraktion und zur qualitativen Weiterverarbeitung) umgewandelt.

Die im Folgenden vorgestellten Generalisierungsalgorithmen können prinzipiell mit exakten numerischen Daten, FNVs oder QMVs gerechnet werden: auf allen drei Datentypen ist eine Vektoraddition definiert, wie wir im vorigen Kapitel gesehen haben. Aus den oben angegebenen Gründen ist es sinnvoll, sie immer auf die exakteste vorliegende Repräsentation anzuwenden und eine Umwandlung in QMVs gegebenenfalls erst anschließend vorzunehmen.

5.2 Generalisierung

Generalisierung eines Bewegungsverlaufs bedeutet, diesen zu vereinfachen. Dazu abstrahiert man von irrelevanten Details des Bewegungsverlaufs. Was in diesem Zusammenhang irrelevant ist, ist eine Frage, der z.B. auch Kartographen gegenüberstehen, wenn sie eine Karte in einem größeren Maßstab darstellen sollen. Der Unterschied ist allerdings, daß der Kartograph ein (relativ) statisches Objekt vorliegen hat und die Daten, für die er eine Generalisierung vornehmen soll, vollständig vorliegen (siehe z.B. [17, 34, 68]). Bewegungsverarbeitung sollte allerdings on the fly geschehen, während immer wieder neue Daten hereinkommen, denn es ist unrealistisch anzunehmen, daß das visuelle System Bewegungsdaten erst verarbeitet, wenn sie vollständig vorliegen. Aber auch für technische Anwendungen wie z.B. in der Robotik kann es durchaus von Nutzen sein, eine qualitative Repräsentation on the fly erzeugen zu können.

5.2.1 Σ -Generalisierung

Eine sehr einfache Idee der Generalisierung ist es, eine bestimmte Schrittweite als Parameter zu wählen und alle Vektoren, die in diese Schrittweite fallen, aufzusummieren.

Sei σ die Distanz, innerhalb der alle Vektoren zusammengefaßt werden (Schrittweite der Generalisierung). Damit man keine Vektoren splitten muß oder neue Ecken im Bewegungsverlauf erhält, die im Original gar nicht vorkommen, wird σ nur als Näherung verstanden.

1. Aufeinanderfolgende Vektoren derselben Richtung werden addiert und zu einem großen Vektor zusammengefaßt. Bei numerischen Daten ist dieser Schritt relativ unwichtig, da aufeinanderfolgende Vektoren der genau gleichen Richtung nicht so oft vorkommen, bei QMV-Sequenzen ist dies allerdings ein häufiger Fall. Vektoren der Länge 0 werden gelöscht.
2. Die Längen aller Vektoren werden aufsummiert, bis σ erreicht wird. Dabei versucht man, sich möglichst dicht an σ anzunähern:

$$\begin{aligned}\sigma_l &= \sum_{i=1}^n v_i \leq \sigma \\ \sigma_u &= \sum_{i=1}^{n+1} v_i > \sigma\end{aligned}$$

Wenn $|\sigma - \sigma_l| < |\sigma - \sigma_u|$, ist der generalisierte Vektor die Summe der ersten n Vektoren, ansonsten die Summe der ersten $n + 1$.

3. Als letzter Schritt werden aufeinanderfolgende Vektoren mit derselben Richtung in der generalisierten Sequenz aufsummiert.

Dies kann alles in einer Iteration erledigt werden. Die 3 Schritte werden wiederholt, bis keine Eingabe mehr vorhanden ist.

Dieser sehr einfache Algorithmus hat eine Komplexität von $O(n)$ und die resultierende Vektorsequenz ist auf jeden Fall nicht komplizierter als (d. h. sie enthält weniger als oder gleichviel Vektoren wie) die Eingabe.

In folgenden Fällen ändert sich an der Eingabe nichts:

1. Die Eingabe besteht aus einem einzigen Vektor.
2. Die Eingabe besteht aus lauter Vektoren, die $\geq \sigma$ sind.
3. Die Eingabe besteht aus lauter Vektoren, die etwas kürzer als σ sind, so daß die Summe von je zwei Vektoren einen größeren Abstand zu σ hat als die Länge eines einzelnen Vektors. Das ist der Fall bei einer Länge im Intervall von $[\frac{2}{3}\sigma, \sigma[$.

Die Anzahl der Vektoren der generalisierten Spur ist begrenzt durch die Summe der Längen der Vektoren v der Originalspur $O(\sum_{v \in O} |v|)$ und σ . Wenn die Eingabesequenz aus lauter kurzen Vektoren besteht, die gut zu Vektoren der Länge σ zusammengefaßt werden können, wird die resultierende Sequenz ungefähr $\frac{\sum_{v \in O} |v|}{\sigma}$ Vektoren enthalten; wenn die Eingabesequenz schon sehr viele Vektoren deutlich länger als σ enthält, kann das natürlich auch viel weniger sein.

Der schlechteste Fall, d. h. der, bei dem die Ergebnisspur im Verhältnis die meisten Vektoren hat, ist der Fall 3 der oben geschilderten Fälle, bei denen Ausgabe gleich Eingabe ist: Wenn alle x Vektoren der Eingabesequenz eine Länge von etwas mehr als $\frac{2}{3}\sigma$ haben, enthält die Ausgabesequenz natürlich genau wieder jene x Vektoren. Der Übergang zum schlechtesten Fall liegt also gerade da, wo alle Vektoren eine Länge von $\frac{2}{3}\sigma$ haben. Im Verhältnis zur Summe der Längen der Vektoren der Originalspur und σ ausgedrückt ist das:

$$x = y \times \frac{\sum_{v \in O} |v|}{\sigma} \rightarrow x = y \times \frac{x \times \frac{2}{3} \times \sigma}{\sigma} \rightarrow x = y \times x \times \frac{2}{3} \rightarrow y = \frac{1}{\frac{2}{3}} = \frac{3}{2}$$

Die obere Schranke für die Anzahl der Vektoren der generalisierten Spur ist also

$$1,5 \times \frac{\sum_{v \in O} |v|}{\sigma}.$$

Bewegungen, die deutlich kleiner sind als σ , werden unterdrückt.

Abbildung 5.1 zeigt eine Bewegungsspur mit der dazugehörigen Σ -Generalisierung. Die „geschwungene“, kontinuierliche Spur ist die Originalspur, erzeugt durch eine Mausebewegung. Die eckigen Punkte geben die Abtastpunkte wieder, wobei die Abtastung alle

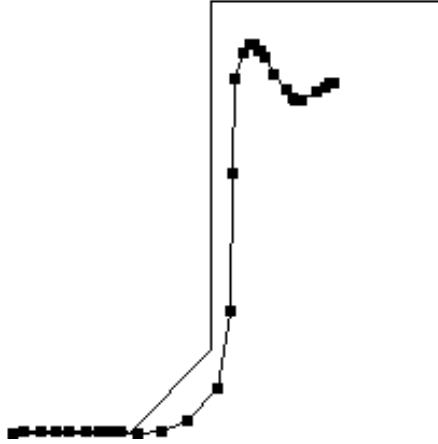


Abbildung 5.1: Originalspur und Σ -Generalisierung mit einer Schrittweite von $30 \times \text{very-close}$, symbolisch gerechnet (siehe Kapitel 4.1.4)

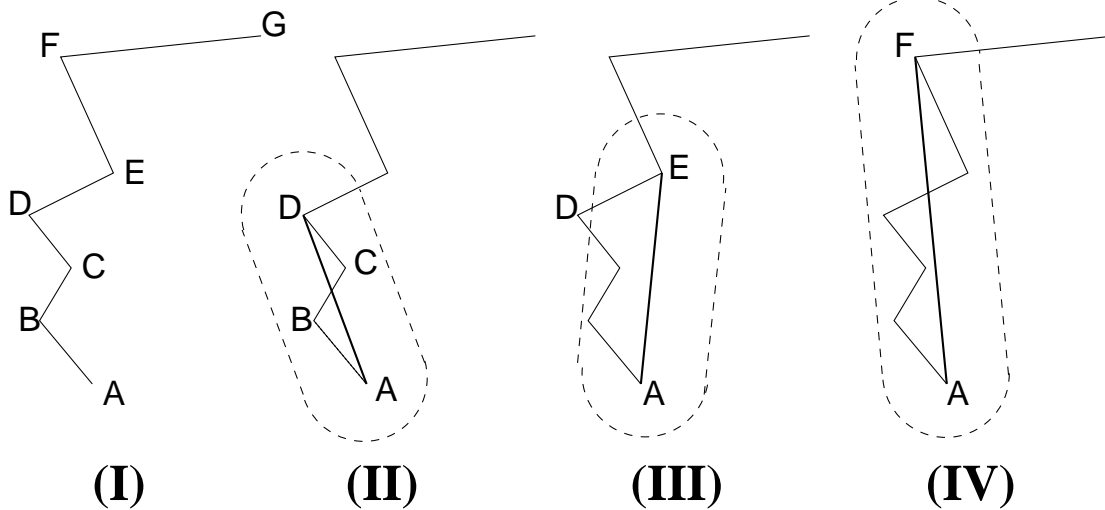
30ms erfolgte. Generalisiert wurde in diesem Fall auf der QMV-Sequenz, die direkt nach der Abtastung aus der Spur erzeugt wurde mit einer Diskretisierung in 8 gleichgroße Richtungsbereiche und 5 Distanzbereiche. Gerechnet wurde dabei mit der in Abschnitt 4.1.4 beschriebenen Methode des symbolischen Rechnens. Die abgebildete Spur ist eine Visualisierung der resultierenden QMV-Sequenz.

Dieser einfache Algorithmus hat den Nachteil, daß er nicht wirklich sensitiv für die großen Richtungswechsel ist, also möglicherweise die Grobstruktur des Bewegungsverlaufs nicht besonders gut wiedergibt, da er einfach alle Vektoren innerhalb einer bestimmten Schrittweite addiert.

5.2.2 ε -Generalisierung

Besser wäre es deswegen, einen Algorithmus zu finden, der eine gegebene Bewegungspur T so in eine Spur U vereinfacht, daß U sich um weniger als eine Distanz ε von T unterscheidet.

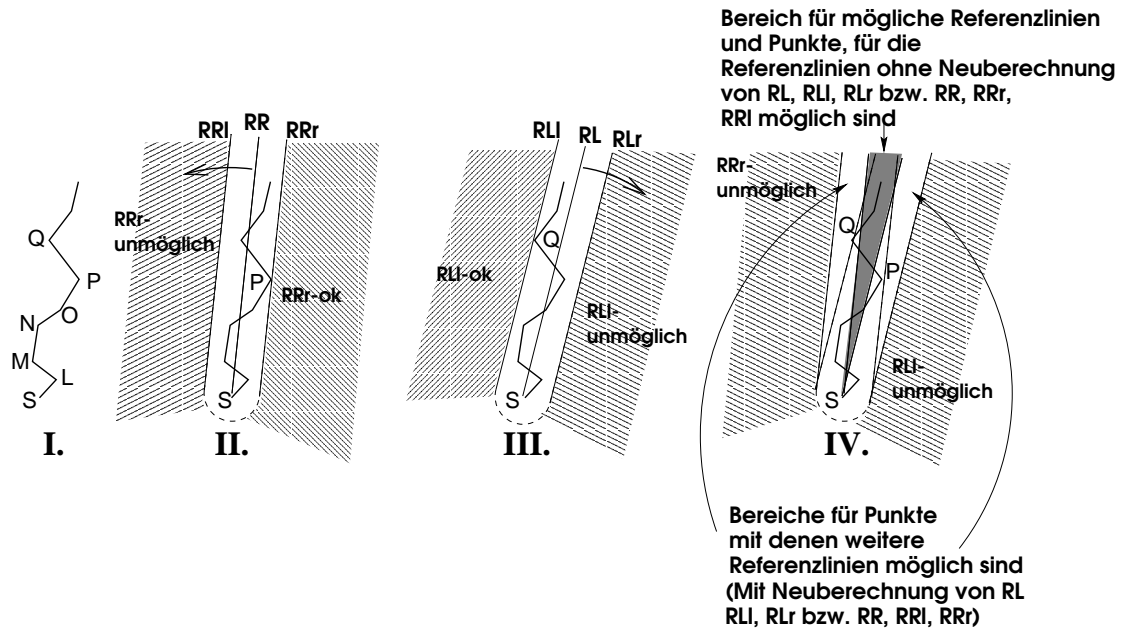
In [68] wird ein Algorithmus vorgestellt, der dies leistet, aber nicht inkrementell ist. Er beruht auf der Idee der rekursiven Verfeinerung der größten Generalisierung eines Polygonzugs, nämlich der Verbindung von Anfangs- und Endpunkt in einer Strecke. Abbruchkriterium ist dabei ein festgelegter Mindestabstand ε : Wenn kein Punkt des Originalpolygons mehr weiter als ε von der Generalisierung entfernt ist, terminiert der Algorithmus. In [82] wurde diese Grundidee aufgegriffen und auf eine inkrementelle Version übertragen.

Abbildung 5.2: Schritte der ε -Generalisierung

Wir erklären das Prinzip des Algorithmus anhand des in [Abbildung 5.2](#) wiedergegebenen Beispiels:

Beginnend mit dem ersten eintreffenden Meßpunkt (hier A), wird immer versucht, zwischen dem ersten und dem letzten eingetroffenen Meßpunkt eine gerade Linie zu ziehen. Kriterium ist, daß jeder Punkt zwischen Anfangs- und Endpunkt in der ε -Umgebung des generalisierten Segments liegt. In [Abbildung 5.2 II](#) ist die Distanz der Punkte B und C zu [AD] kleiner als ε , so daß [AD] eine gültige Generalisierung des Polygonzugs zwischen A und D darstellt. Nun trifft der nächste Meßpunkt, E, ein, der nun als Endpunkt des generalisierten Segments getestet wird. Jetzt liegt D außerhalb der ε -Umgebung von [AE] (siehe [Abbildung 5.2 III](#)). Dies ist für den Algorithmus noch kein Grund, [AD] als letzte gültige Generalisierung zu betrachten und ein neues Segment zu beginnen, denn es besteht die Möglichkeit, daß neue Meßpunkte eintreffen, so daß die Referenzstrecke wieder weiter nach links wandert und D wieder innerhalb der ε -Umgebung liegt. F ist solch ein Punkt. Erst wenn unmöglich ein zukünftiger Punkt noch bewirken könnte, daß wieder alle Punkte in der ε -Umgebung der Referenzlinie liegen, wird die Generalisierung des Segments abgeschlossen. Endpunkt des Segments ist der letzte Punkt, für den alle vorigen Punkte noch in der ε -Umgebung lagen. Er ist auch gleichzeitig der Anfangspunkt für die Referenzlinie des nächsten Segments. Diese Schritte werden solange iteriert, bis die Eingabe beendet ist.

[Abbildung 5.3](#) illustriert, wie getestet wird, ob eine zukünftige Referenzlinie R das ε -Kriterium wieder erfüllen könnte: Der erste Endpunkt der Referenzlinie ist gleichzeitig

Abbildung 5.3: Test der ε -Bedingung

der Startpunkt. Von S aus hat die Referenzlinie alle Freiheitsgrade in der Rotation; jeder beliebige zweite Punkt erfüllt selbstverständlich das ε -Kriterium, da beide Punkte die Endpunkte der Referenzstrecke bilden und somit natürlich weniger als ε von der Linie entfernt sein würden. Wenn der zweite Punkt noch in der ε -Umgebung von S liegt, schränkt er die Freiheitsgrade der Referenzstrecke für weitere Punkte immer noch nicht ein, da jede Strecke das ε -Kriterium erfüllen würde.

Falls der zweite Punkt der Eingabe allerdings weiter als ε vom Startpunkt S entfernt ist, werden die Freiheitsgrade der Referenzlinie für weitere Punkte eingeschränkt: Während sie vorher um jeden beliebigen Winkel nach rechts oder links geneigt werden konnte, ohne daß mit einem neuen Punkt ein alter Punkt des Polygonzugs aus der ε -Umgebung der Strecke hinausfällt, begrenzt jeder weiter entfernte Punkt den zulässigen Winkel.

In unserem Beispiel würde für die Punkte S und L jede Referenzlinie die ε -Bedingung erfüllen, da sie nicht weiter als ε voneinander entfernt sind. Das Gleiche gilt für M. Sobald allerdings der Punkt N dazukommt, werden die Rotationsmöglichkeiten der Referenzlinie eingeschränkt: Z.B. könnte die Referenzlinie nicht von S aus wieder rückwärts laufen, da S und N weiter als ε voneinander entfernt sind.

Sei nun die Sequenz in unserem Beispiel bis zum Punkt vor P abgearbeitet. In [Abbildung 5.3 II](#) ist dargestellt, was passiert, wenn der Punkt P neu dazukommt: Er verhindert jede

weitere Rotation der Referenzlinie nach links. In in 5.3 III ist dargestellt, wie der neu hinzukommende Punkt Q eine weitere Rotation nach rechts verhindert.

Generell gilt, daß es für jeden Punkt in den Halbebenen **RRr-unmöglich** oder **RLl-unmöglich** unmöglich ist, eine Referenzlinie zu finden. **RRl** ist die Referenzlinie, für die P gerade noch das ε -Kriterium erfüllt, **RLr** die Referenzstrecke, für die Q gerade noch das ε -Kriterium erfüllt. **RRr** geht parallel zu **RR** durch P, **RRl** liegt im Abstand ε auf der anderen Seite parallel zu **RR**. Jeder Punkt links von **RRl** macht es unmöglich, daß eine einzige Referenzstrecke das ε -Kriterium noch erfüllen könnte. Analoges gilt für **RL**, **RLl**, **RLr**. Diese (Halb-)Geraden sind leicht zu berechnen.

Allgemein wird jedes Segment folgendermaßen generalisiert:

1. Der erste Punkt X_1 und der zweite Punkt X_2 sind Anfangs- und Endpunkt der ersten Referenzstrecke R_1 .
2. Für jeden neu hinzukommenden Punkt X_n wird getestet, ob er in **RRr-unmöglich** oder **RLl-unmöglich** liegt.
 - a) Wenn das nicht der Fall ist, wird getestet, ob X_n in **RRr-ok** oder in **RLl-ok** liegt.
 - i. Falls ja, wird X_n als neuer Randpunkt benutzt, und **RR**, **RRr**, **RRl** (oder **RL**, **RLl**, **RLr**) werden neu berechnet.
 - ii. Falls nein, muß man hier gar nichts machen und und kann mit dem nächsten Punkt der Eingabe bei (2) weitermachen.
 - b) Wenn X_n in **RRr-unmöglich** oder **RLl-unmöglich** liegt, wurde das Ende des letzten Segments bereits überschritten. Dann wird ein Punkt X_k ($k < n$), der den Endpunkt der letzten korrekten Referenzlinie gebildet hat, Endpunkt des generalisierten Segments R_1 . Gleichzeitig beginnt der Algorithmus mit X_k als neuem Anfangspunkt wieder beim Schritt (1), um das nächste Segment R_2 zu berechnen, usw.

Dieser Algorithmus erzeugt ganz offenbar eine Vektorsequenz, in der alle Ecken gleichzeitig Ecken der Originalsequenz sind. Deshalb kann die resultierende Sequenz niemals komplexer sein als die Originalsequenz. Kein Punkt der Originalsequenz ist weiter als eine bestimmte Distanz ε von der generalisierten Sequenz entfernt.

Die durchschnittliche Komplexität des Algorithmus beträgt $O(n)$ wegen der im Prinzip linear-inkrementellen Abarbeitung, d.h. normalerweise werden die meisten Punkte nur einmal bearbeitet. Im schlimmsten Fall, wenn viel Backtracking stattfindet (d.h. wenn bei vielen Segmenten noch gewartet wird, ob noch ein Punkt kommt, der bewirkt, daß das ε -Kriterium für das Segment wieder erfüllt ist, dieser Punkt aber dann doch nicht kommt), muß jeder Punkt zweimal bearbeitet werden und die Komplexität beträgt $O(n^2)$.

Die Originaldynamik des Bewegungsverlaufs, also der Wechsel von Schnell/Langsam oder gar Stillstand, gehen sowohl mit der ε - als auch mit der Σ -Generalisierung verloren. Es kann nur noch eine Durchschnittsgeschwindigkeit über jedes generalisierte Segment berechnet werden. Dazu kann man entweder die Geschwindigkeit nehmen, die sich ergibt aus der Länge des generalisierten Segments und der Zeit, die in der Originalsequenz gebraucht wurde, um zum Endpunkt des Segments zu gelangen. Das sagt dann natürlich nichts mehr über die tatsächliche Geschwindigkeit des sich bewegenden Objekts in der Originalspur aus: Es könnte sich sehr schnell bewegt haben, aber es wurde einfach viel wegeneralisiert. Eine andere Möglichkeit ist, aus allen *Geschwindigkeiten* der Originalspur, die zum jeweils generalisierten Segment gehört, einen Durchschnitt zu berechnen, also die Länge dieses Teils Originalspur im Verhältnis zur benötigten Zeit zu betrachten. Das würde zwar die tatsächliche Geschwindigkeit des Objekts besser wiedergeben, aber letztendlich doch teilweise für die generalisierte Sequenz eine ziemlich irreführende Dynamik erzeugen: Für kurze, schnelle Segmente der Generalisierung könnte unter Umständen in der Originalspur sehr viel Zeit gebraucht worden sein, je nachdem, ob dort viel wegeneralisiert wurde.

Die beste Möglichkeit ist natürlich, Dynamikwechsel auch als Generalisierungskriterium mit aufzunehmen, und so ein kombiniertes Kriterium Dynamik/Spur z. B. als ε -Bedingung für die Generalisierung zu benutzen. Dies ist für zukünftige Erweiterungen des MM-TOOLS geplant.

5.2.3 Beispiele

Abbildung 5.4 zeigt die ε -Generalisierung einer Bewegungsspur mit einem ε von 100 Pixeln. Im MM-TOOL wurde die nichtinkrementelle Version der ε -Generalisierung implementiert, die ähnliche Ergebnisse erzeugt wie die inkrementelle Version.

Eine Implementierung der hier geschilderten inkrementellen ε -Generalisierung wurde in einem autonomen Roboter zur Navigation aufgrund von Eigenbewegung implementiert. Dies ist ausführlich im Kapitel 8.1.1 erläutert. Abbildung 5.5 zeigt eine Trajektorie dieses Roboters, die in den Gängen eines Bürogebäudes zurückgelegt wurde. Die Kästchen symbolisieren die durch Generalisierung gefundenen Ecken. Als ε wurde hier die jeweilige Gangbreite benutzt.

5.3 Konstruktion der Shape-Repräsentation

5.3.1 Segmentierung

Der erste Schritt zur Konstruktion der SHAPE-Repräsentation ist die Segmentierung der QMV-Sequenz in basale SHAPES. Welche SHAPES dies sind, hängt von der gewählten Diskretisierung der Richtungsdomäne ab. Bei vier Richtungsbereichen ist dies

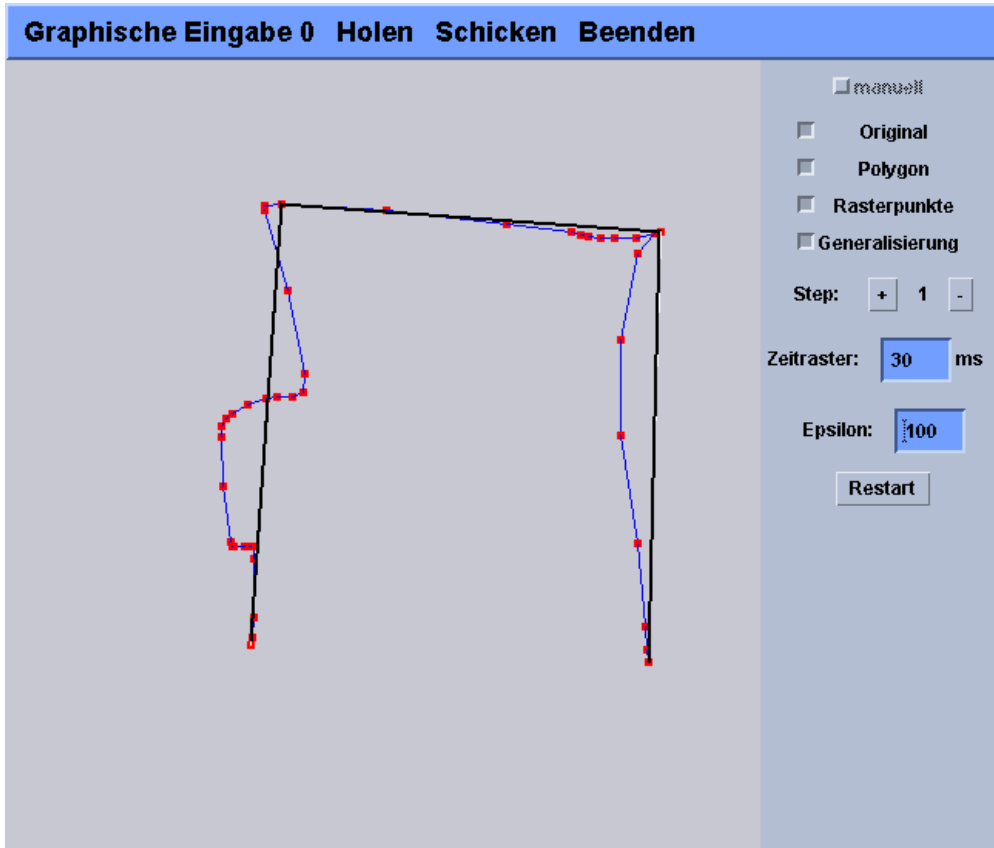


Abbildung 5.4: Originalspur (fein) und ε -Generalisierung (dick) im MM-TOOL

die Menge $\{\text{left-turn, right-turn, backward-turn, straight-line}\}$, bei acht Richtungsbereichen $\{\text{left-turn, right-turn, backward-turn, straight-line, half-left-turn, half-right-turn, three-quarter-left-turn, three-quarter-right-turn}\}$. Jeder der basalen SHAPES kann in verschiedenen Größen vorkommen, z.B. in den Größen `big`, `medium-size` und `small`. Im dreidimensionalen Raum funktioniert die Ermittlung der basalen SHAPES analog; je nach Diskretisierung der Richtungsdomäne im dreidimensionalen Raum werden die möglichen Sektorenübergänge identifiziert und benannt.

Diese SHAPES können anschließend zu komplexeren SHAPES kombiniert werden, die einem vordefinierten SHAPE-Vokabular entnommen sind, z.B. der Aresti-Notation oder einer beliebigen anderen Menge von MOTION SHAPES. Für eine Aufgabe wie die automatische Generierung von Routenbeschreibungen kann man z.B. (außer den basalen SHAPES) Elemente wie `u-turn-left`, `u-turn-right`, `s-curve-left`, `s-curve-right`, `loop-`

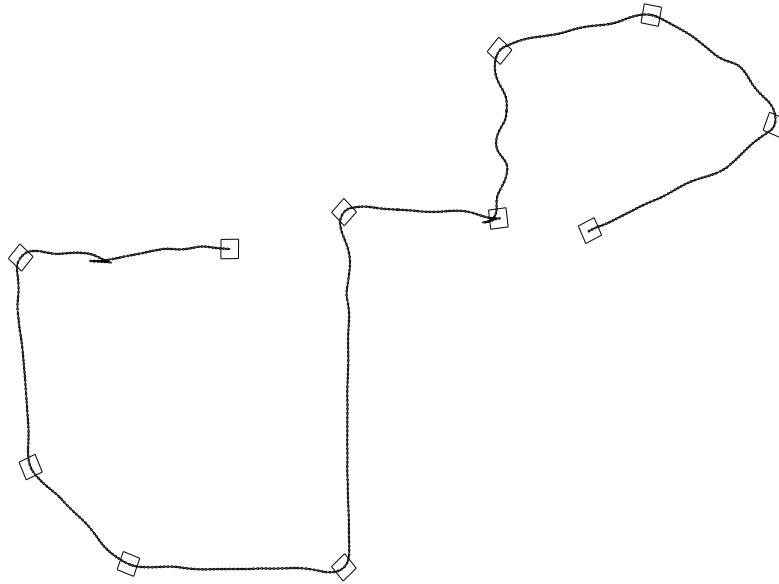


Abbildung 5.5: Originalspur einer Roboterbewegung und durch inkrementelle ε -Generalisierung gefundene Ecken (Kästchen). Die Verbindung der Ecken durch gerade Linien wurde in der Darstellung weggelassen.

`left`, `loop-right` wählen.

Die Eingabe für den Segmentierungsalgorithmus ist eine generalisierte QMV-Sequenz; das bedeutet, daß die Feinstruktur des Bewegungsverlaufs bereits geglättet wurde und nur die Information über Richtungswechsel, die uns für die Konstruktion der SHAPE-Repräsentation interessiert, erhalten blieb. Deswegen können wir auch jeden Richtungswechsel in der QMV-Sequenz als Kurve, d.h. als einen der „Turns“ aus der Menge der basalen SHAPES, auffassen.

Das Problem bei der Segmentierung ist, daß mit der diskreten Darstellung in der QMV-Repräsentation die Information über die Größe der Kurven verlorengegangen ist. Es ist in der QMV-Repräsentation unklar, ob eine Sequenz, die zunächst nach Osten führt und dann ungefähr gleich weit nach Norden, in `straight-line`, `left-turn`, `straight-line` segmentiert werden soll oder in einen einzigen großen `left-turn`. In der ersten Möglichkeit sind die geraden Anteile gegenüber den `left-turn` betont, in der zweiten Möglichkeit sind keine geraden Anteile mehr vorhanden; der einzige SHAPE ist eine große Kurve. Weil die Information, ob die Kurve ursprünglich gerade Anteile und spitze Ecken hatte (was der ersten Möglichkeit im Beispiel entspricht), oder sehr rund war (was der zweiten Möglichkeit im Beispiel entspricht), in der QMV-Sequenz nicht mehr enthalten ist, sind beide Möglichkeiten legitim und welche man auswählt eine bloße Designent-

scheidung oder kann als Parameter in den Algorithmus eingehen.

Wir haben uns hier zunächst dazu entschieden, jeder Kurve den größtmöglichen Radius zu geben, d.h. also eher von runden als von eckigen Bewegungsverläufen auszugehen. So scheint eine möglichst umfassende Klassifizierung in zusammengesetzte SHAPES am ehesten möglich, da sich so die meisten direkt aufeinanderfolgenden Kurven ergeben. Bei Sequenzen mit vielen langen **straight-lines** zwischen kleinen Kurven ist die Entscheidung, wie die Linien und Kurven zusammengefaßt werden sollen, schwieriger als bei Sequenzen mit vielen unterschiedlich großen, direkt aufeinanderfolgenden Kurven mit nur wenigen, möglichst kurzen **straight-lines** dazwischen.

Eine Erweiterung des Algorithmus soll allerdings variabel segmentieren können. So ist es z.B. für die automatische Erzeugung von Routenbeschreibungen wie in Abschnitt 8.2 beschrieben, nicht nötig, den Bewegungsverlauf in große Kurven zu segmentieren. Im Gegenteil sollen Richtungsänderungen an Kreuzungen möglichst scharf segmentiert werden. Nur in Ausnahmefällen ist hier eine Segmentierung in große SHAPES sinnvoll, wenn z.B. die Straße einen Bogen macht und somit die Form des Bewegungsverlaufs selbst als Landmarke dienen kann. Eine Implementierung dieser variablen Segmentierung ist allerdings erst für die Zukunft geplant.

Der Algorithmus, zuerst vorgestellt in [82], arbeitet auf QMV-Sequenzen, die als Polygonzug im QMV-Vektorraum vorliegen (siehe Abschnitt 4.1.4). Wir beschränken uns im Folgenden auf den Fall, daß die Richtungsdomäne in 4 Bereiche diskretisiert ist (acht Bereiche sind analog). Dann besteht die Spur aus horizontalen und vertikalen Linien wie in Abbildung 5.6 (I), die eine Visualisierung der QMV-Sequenz

```
<very-far slow forward> <far slow right> <medium-dist very-slow right>
<medium-dist very-slow right> <very-close very-slow left> <far medium-
vel left>
```

darstellt.

Bis auf das erste und letzte Stück des Polygonzugs hat jedes Stück an zwei Kurven Anteil. Um den maximalen Radius dieser Kurven zu ermitteln, sortieren wir die Stücke nach der Länge. Bei jedem Stück, das an zwei Kurven Anteil hat, zählt die Länge nur zur Hälfte, da es ja in (mindestens) zwei Teile geteilt wird. Im Beispiel in Abbildung 5.6 ergibt sich so die Ordnung e, c, d, b, f, a.

Der Algorithmus beginnt mit der kürzesten Strecke e. Sie hat Anteil an zwei Kurven und wird deswegen in der Mitte geteilt. Die angrenzenden Strecken d und f werden um die halbe Länge von e gekürzt (dargestellt in Abbildung 5.6 (II)). Die verbleibenden Linien werden dann wiederum nach ihrer neuen Länge sortiert: d hat nun nur noch an einer Kurve Anteil und wird deswegen mit seiner vollen neuen Länge sortiert; e und f haben an keiner weiteren Kurve mehr Anteil und werden deswegen aussortiert. Mit dieser neuen Ordnung c, b, d, a beginnt der Algorithmus nun wieder von vorn bei der nun kürzesten Strecke. Wenn er mit einer Strecke beginnt, die nur (noch) an einer

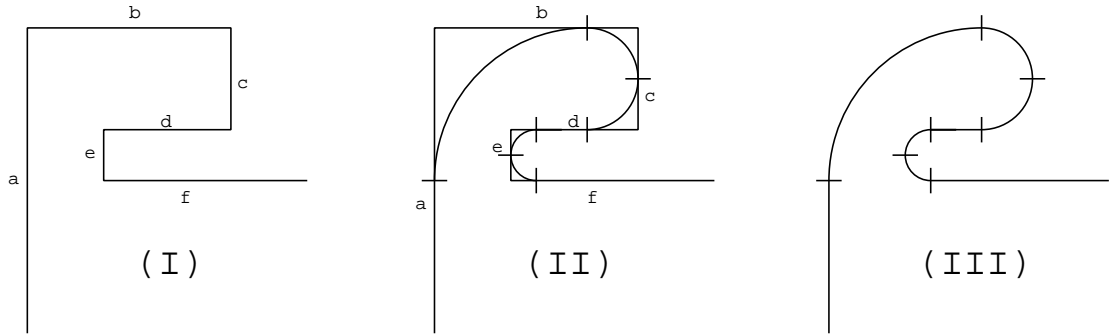


Abbildung 5.6: Segmentierung einer QMV-Sequenz

Kurve Anteil hat, wird dieser Kurve selbstverständlich die ganze Strecke zugeteilt, wie man z.B. an Strecke b sehen kann. Wenn keine Strecken mehr übrig sind, terminiert der Algorithmus.

Abbildung 5.6 (III) zeigt den Graph der SHAPE-Repräsentation der QMV-Sequenz. Sie lautet:

```
<straight-line medium-size slow><right-turn medium-size slow><right-turn
small very-slow><right-turn small very-slow><straight-line very-small
very-slow><left-turn very-small very-slow><left-turn very-small very-
slow><straight-line big medium-vel>
```

Wie in [82] dargelegt, arbeitet der Algorithmus auch on the fly mit einer Vorausschau von lediglich zwei bis drei Vektoren normalerweise gut.

5.3.2 Klassifizierung

Bisher wurde die QMV-Sequenz nur in elementare SHAPES segmentiert. Diese können nun zu komplexeren SHAPES zusammengefaßt werden, d. h. in einer Sequenz aus basalen SHAPES werden Subsequenzen herausgegriffen und nach bestimmten Regeln klassifiziert. In diesen Regeln wird ein Vokabular von komplexeren Bewegungsmustern wie Schleifen, Spiralen, Zickzacks, Kunstflugfiguren, Eislauffiguren, oder was immer für eine bestimmte Anwendung passend ist, eingeführt und als Sequenzen von basalen (oder auch komplexen) SHAPES definiert.

Es sind verschiedene Möglichkeiten denkbar, wie dies geschehen kann:

- Die Regeln, welche basalen SHAPES zu welchen komplexen SHAPES zusammengefaßt werden, können für dasselbe SHAPE-Vokabular unterschiedlich definiert werden, oder es können gleich verschiedene SHAPE-Vokabulare definiert werden.

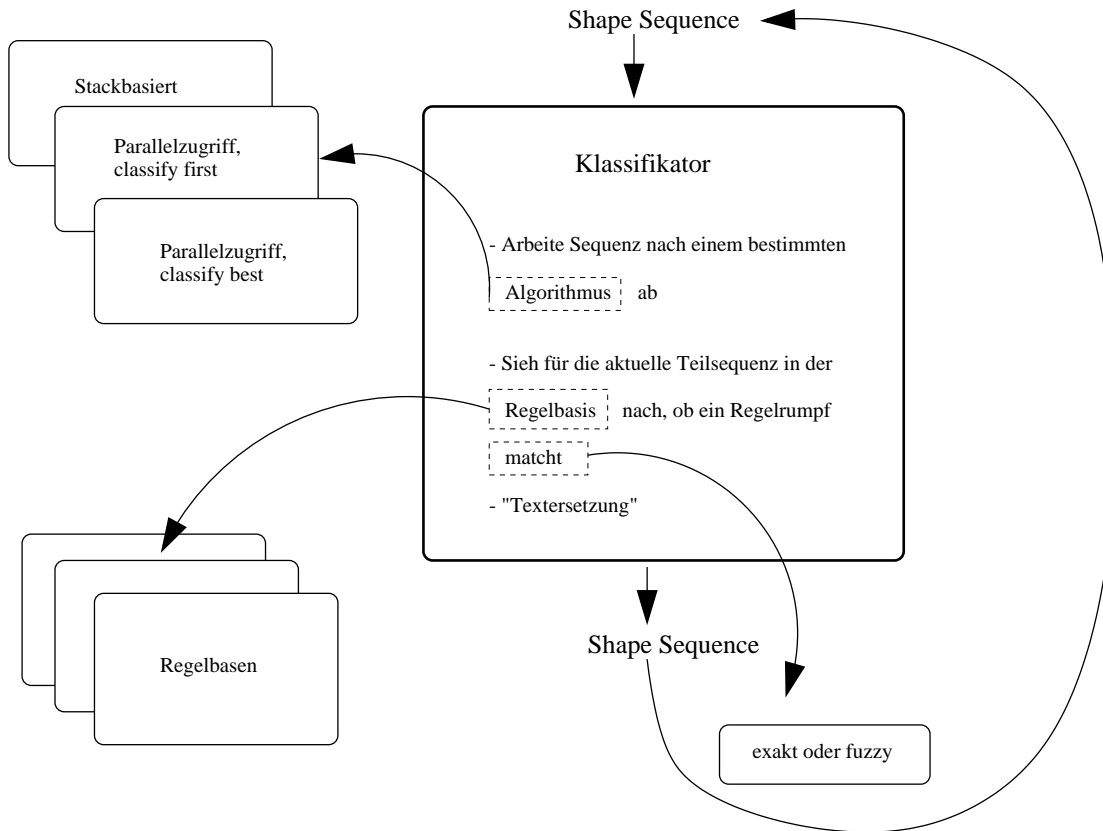


Abbildung 5.7: Modularer Aufbau der Klassifikation

- Es sind unterschiedliche Arten denkbar, wie die SHAPE-Sequenzen bei der Klassifikation on the fly abgearbeitet werden. So können z. B. die eintreffenden SHAPES auf einen Stack gelegt werden und es wird klassifiziert, sobald eine Regel angewendet werden kann, oder man legt die eintreffenden SHAPES in einen Array mit Parallelzugriff ab und klassifiziert nicht sofort, sondern wartet ein paar SHAPES ab, ob sich eine bessere Klassifikationsmöglichkeit ergibt; usw.
- Auch das Matching für die Klassifikationsregeln kann unterschiedlich definiert werden; so ist denkbar, daß der Rumpf einer Regel immer exakt matchen muß, damit sie anwendbar ist, oder daß eine gewisse Unschärfe erlaubt ist.

In Abbildung 5.7 ist gezeigt, wie die einzelnen Möglichkeiten modular implementiert und austauschbar zusammengefügt werden können.

Wenden wir uns zunächst den Regeln zu, die die Konstruktion der komplexen SHAPES definieren.

Regeln für die Klassifikation

Ein Beispielvokabular bestehend aus den SHAPES {u-turn-left, u-turn-right, s-curve-left, s-curve-right, loop-left, loop-right} kann z. B. für jede Größe *size* wie in Regelbasis 5.1 definiert werden.

Regelbasis 5.1 Einfache Regelbasis

<u-turn-left <i>size</i> >	←	<left-turn <i>size</i> > <left-turn <i>size</i> >
<u-turn-right <i>size</i> >	←	<right-turn <i>size</i> > <right-turn <i>size</i> >
<s-curve-left <i>size</i> >	←	<u-turn-left <i>size</i> > <u-turn-right <i>size</i> >
<s-curve-right <i>size</i> >	←	<u-turn-right <i>size</i> > <u-turn-left <i>size</i> >
<loop-left <i>size</i> >	←	<u-turn-left <i>size</i> > <u-turn-left <i>size</i> >
<loop-right <i>size</i> >	←	<u-turn-right <i>size</i> > <u-turn-right <i>size</i> >

Selbst bei dieser einfachsten Regelmenge wird deutlich, daß die Klassifikation der basalen MOTION SHAPES in komplexere mehrstufig stattfindet: SHAPES wie u-turn-left sind aus basalen SHAPES aufgebaut, während die Bausteine von SHAPES wie s-curve-right bereits selbst komplexe SHAPES sind. Die (weiter unten vorgestellten) Klassifikationsalgorithmen müssen also immer wieder rekursiv benachbarte SHAPES zusammenfassen, auf die eine der Regeln anwendbar ist. Dies geschieht so lange, bis keine Regel mehr anwendbar ist.

Neben diesen einfachen Regeln sind auch “kompliziertere” Regeln denkbar, bei denen ein MOTION SHAPE nicht mehr nur aus je zwei SHAPES niedrigerer Komplexität aufgebaut ist, sondern mehrere Klassifikationsregeln für denselben komplexen SHAPE auch nicht ganz so eindeutige Fälle berücksichtigen: Wenn z.B. zwischen zwei turns gleicher Größe eine viel kleinere straight-line vorhanden ist, können diese drei SHAPES ebenfalls zu einem u-turn zusammengefaßt werden, wie in Regelbasis 5.2 exemplarisch gezeigt.

Bei zwei Regeln pro Regelkopf hängt die Anwendungsreihenfolge von dem Algorithmus ab, mit dem die SHAPE-Sequenz abgearbeitet wird.

Um noch mehr basale SHAPES zusammenfassen zu können, kann es auch sinnvoll sein, SHAPES unterschiedlicher Größe zu kombinieren. Dazu können entweder zusätzliche SHAPES in das Vokabular aufgenommen werden, oder aber stattdessen neue Größenangaben zugelassen werden. So könnte man z.B. die beiden entgegengerichteten u-turns in unserem Beispiel zu einer s-curve-right kombinieren, die in der Größe abnehmend ist. Die Menge der Größenangaben {small, medium-sized, big} muß dann erweitert werden um Größenangaben wie small-increasing, medium-size-increasing, medium-size-decreasing, big-decreasing.

Regelbasis 5.2 Regelbasis mit mehr als einer Regel für komplexe SHAPES

<code><u-turn-left size></code>	\leftarrow	<code><left-turn size></code> <code><left-turn size></code>
	\leftarrow	<code><left-turn size></code> <code><straight-line size-2></code> <code><left-turn size></code>
<code><u-turn-right size></code>	\leftarrow	<code><right-turn size></code> <code><right-turn size></code>
	\leftarrow	<code><right-turn size></code> <code><straight-line size-2></code> <code><right-turn size></code>
<code><s-curve-left size></code>	\leftarrow	<code><u-turn-left size></code> <code><u-turn-right size></code>
	\leftarrow	<code><u-turn-left size></code> <code><straight-line size-2></code> <code><u-turn-right size></code>
<code><s-curve-right size></code>	\leftarrow	<code><u-turn-right size></code> <code><u-turn-left size></code>
	\leftarrow	<code><u-turn-right size></code> <code><straight-line size-2></code> <code><u-turn-left size></code>
<code><loop-left size></code>	\leftarrow	<code><u-turn-left size></code> <code><u-turn-left size></code>
	\leftarrow	<code><u-turn-left size></code> <code><straight-line size-2></code> <code><u-turn-left size></code>
<code><loop-right size></code>	\leftarrow	<code><u-turn-right size></code> <code><u-turn-right size></code>
	\leftarrow	<code><u-turn-right size></code> <code><straight-line size-2></code> <code><u-turn-right size></code>

In Regelbasis 5.3 sind die entsprechenden Regeln (für $size \in \{\text{small}, \text{medium-size}, \text{big}\}$) für `s-curve-right` exemplarisch angegeben.

Algorithmen zur Abarbeitung der Shape-Sequenz während der Klassifikation

Einfache stackbasierte Lösung Die Abarbeitung der SHAPE-Sequenz kann mit einem einfachen Stack realisiert werden. Die SHAPES werden in der Reihenfolge, in der sie erzeugt werden, auf den Stack gelegt. Sobald oben auf dem Stack eine Kombination von SHAPES liegt, auf die eine der vordefinierten Klassifikationsregeln anwendbar ist, werden diese SHAPES vom Stack entfernt, und der neu konstruierte komplexere SHAPE wird oben auf dem Stack abgelegt. Dabei hat die Kombination von SHAPES auf dem Stack Vorrang vor der Abarbeitung neuer SHAPES; diese werden erst auf den Stack gelegt, wenn auf die obersten Stackelemente keine Regeln mehr anwendbar sind.

Lösungen mit Memory Buffer In Abschnitt 2.3 ist ein in unserem Projekt durchgeführtes Experiment zur Reproduktion von Trajektorien beschrieben (siehe auch Abbildung 2.3).

Um die im dort ermittelten Reproduktionsleistungen, die ja nicht nur eine

Regelbasis 5.3 Regelbasis mit neuen Größen

```

<s-curve-right size> ← <u-turn-right size> <u-turn-left size>
                       ← <u-turn-right size> <straight-line size-2>
                       <u-turn-left size>

    <s-curve-right
size-decreasing> ← <u-turn-right size> <u-turn-left size-1>
                  ← <u-turn-right size> <u-turn-left size-2>
                  ← <u-turn-right size> <straight-line size>
                    <u-turn-left size-1>
                  ← <u-turn-right size> <straight-line size>
                    <u-turn-left size-2>
                  ← <u-turn-right size> <straight-line size-1>
                    <u-turn-left size-1>
                  ← <u-turn-right size> <straight-line size-1>
                    <u-turn-left size-2>

```

Wahrnehmungs-, sondern auch eine Gedächtnisleistung sind, zu erklären, sind nicht nur Algorithmen erforderlich, sondern auch ein Gedächtnismodell. In Anlehnung an ein in [80] vorgeschlagenes Gedächtnismodell für tiefere Schichten der Bewegungswahrnehmung haben wir die folgenden Modelle entwickelt.

Memory Buffer mit LIFO-Abarbeitung Statt eines simplen Stacks oder LIFO-Buffers wird ein Memory Buffer benötigt, der parallelen Zugriff auf alle Elemente erlaubt. Das ist notwendig, weil das Experiment darauf hinweist, daß ab einer bestimmten Länge¹ des Bewegungsverlaufs nicht mehr alle Details im Gedächtnis behalten werden können. Dann wird aber nicht der Anfang vergessen², sondern Muster, die nicht besonders auffällig sind. Deswegen muß es möglich sein, die aktuellen Gedächtnisinhalte zu inspizieren und, wenn der Buffer voll ist, nicht den ältesten Inhalt vergessen, sondern den, der von einer „Auffälligkeitsfunktion“ den niedrigsten Wert zugewiesen bekommt.

Unserer Vorstellung nach existiert irgendwo im Gehirn eine am Prozeß der Bewegungswahrnehmung beteiligte Struktur mit einer bestimmten Speicherkapazität³.

¹Länge meint hier nicht die zeitliche Dauer, sondern die Anzahl von SHAPES, die im Gedächtnis gehalten werden müssen

²Es ist unklar, ob die Details des Bewegungsverlaufs wirklich vergessen werden oder evtl. nur falsch mit auffälligeren Teilen verschmelzen. Dies muß experimentell ermittelt werden.

³Diese Speicherkapazität muß empirisch bestimmt werden; dazu sind allerdings Vorstudien notwendig, die überhaupt erst einmal ein SHAPE-Vokabular für die menschliche Bewegungswahrnehmung ermitteln. Diese Experimente sind in Vorbereitung.

Die in SHAPES segmentierten Elemente des Bewegungsverlaufs werden dort der Reihe nach abgelegt und weiterverarbeitet. Nach dem Stack-Prinzip wird immer versucht, so schnell wie möglich die oben liegenden Inhalte zu reduzieren, d.h. eine der Klassifikationsregeln anwenden, die Originalinhalte löschen und den neu klassifizierten SHAPE „oben“ im Buffer ablegen.

Wenn der Buffer voll ist, aber trotzdem eine neue Eingabe zur Weiterbearbeitung ansteht, muß ein Element des Buffers verdrängt werden. Das ist das Element, das von der schon erwähnten Auffälligkeitsfunktion den niedrigsten Wert zugewiesen bekommt.

Diese Auffälligkeitsfunktion muß Gegenstand weiterer experimenteller Forschungen sein. Im Moment nehmen wir an, daß ein SHAPE umso auffälliger ist, je mehr basale SHAPES an seiner Konstruktion beteiligt waren. Wenn viele basale SHAPES an der Konstruktion eines Bufferelements beteiligt waren, so wird in diesem Element ganz offenbar auf wenig Platz viel Information gespeichert. Deswegen scheint es eine sinnvolle Strategie zu sein, nicht ein Element wie dieses zu verdrängen, sondern eins, das weniger Information beinhaltet. Dieses Basismaß von Auffälligkeit muß noch durch andere Faktoren gewichtet werden, wie z.B. Alter des Elements im Buffer, Größe des SHAPES, usw. Auch scheint es, daß das erste und das letzte Bufferelement wegen primacy/recency-Effekten besser erinnert werden. Das erste Element höher zu gewichten ist auch aus algorithmischen Gründen sinnvoll, da darauf geachtet werden muß, daß nicht die obersten Elemente verdrängt werden, weil die ja noch an einer weiteren Klassifikation beteiligt sein könnten, die dann evtl. nicht mehr vorgenommen werden kann.

Außerdem sind offenbar SHAPES mit spitzen Ecken auffälliger als runde, was, sollte sich dies experimentell bestätigen, für unseren Ansatz ein Problem darstellt, da diese Information mit der QMV-Repräsentation verloren geht. Im nächsten Abschnitt wird eine Lösung dafür vorgestellt.

Abbildung 5.8 zeigt die Konstruktion einer komplexen SHAPE-Sequenz aus dem Bewegungsverlauf aus Abbildung 5.6 in einem Memory Buffer der Größe 5.

Memory Buffer mit „combine best“-Abarbeitung Auch diese Lösung arbeitet mit einem Memory Buffer. Es geht hier allerdings nicht darum, die menschliche Kognition zu modellieren (es wird hier auch nichts vergessen oder verdrängt wie im vorigen Ansatz), sondern die Verwendung einer Struktur mit Parallelzugriff hat hier lediglich algorithmische Gründe. Die Regelanwendung findet hier nicht sobald möglich statt, wie bei den beiden vorigen Ansätzen, sondern es wird ausgenutzt, daß man durch den Parallelzugriff eine bestimmte Anzahl SHAPES gleichzeitig überblicken kann. Wenn es verschiedene Kombinationsmöglichkeiten aufgrund der Regeln gibt, nimmt man diejenige Regel, die es erlaubt, die auffälligsten SHAPES zu erzeugen.

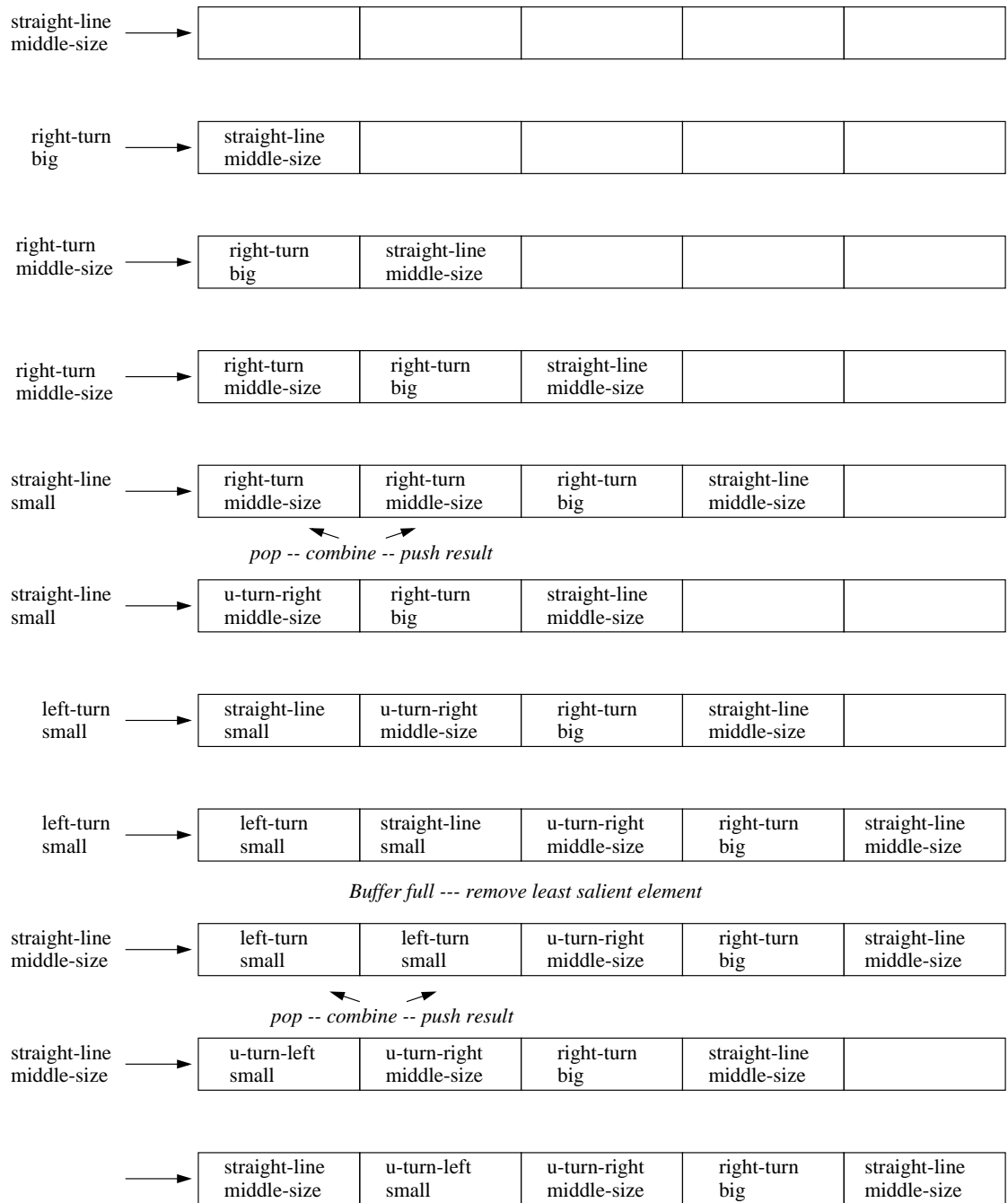


Abbildung 5.8: Klassifikation mit Memory Buffer

5.3.3 Weitere Attribute für Shapes

Neben den sich schon aus den assoziierten QMV-Sequenzen direkt ergebenden Attributen für SHAPES wie Größe oder interne Charakteristik der Bewegungsdynamik können den SHAPES auch weitere Attribute zugewiesen werden. Aus dem vorangegangenen Abschnitt geht hervor, daß die Unterscheidung rund vs. eckig für die Modellierung menschlicher kognitiver Leistungen möglicherweise wichtig sein kann. Durch die Diskretisierung der Richtungsdomäne und die diskrete Abtastung des Bewegungsverlaufs geht diese Information allerdings in der QMV-Repräsentation verloren bzw. wird gar nicht erst ermittelt, wie im Abschnitt 5.3.1 ausgeführt wurde.

Durch die Idee der Nachbarschaften zwischen Richtungsrelationen, wie sie in [42] für Orientierungsrelationen angegeben sind, ergibt sich aber eine Möglichkeit, dies trotzdem in die SHAPE-Repräsentation einfließen zu lassen. Demnach hat jede Orientierungsrelation zwei Nachbarn, und zwar jeweils die Relation, die man im Orientierungsfadenkreuz erreicht, wenn man im bzw. gegen den Uhrzeigersinn in das nächste Intervall geht. So ist z. B. in Abbildung 3.3(I) **west** der linke Nachbar (bzw. der Nachbar gegen den Uhrzeigersinn) von **north**, und **east** der rechte Nachbar (bzw. der Nachbar im Uhrzeigersinn), wohingegen diese Relationen in 3.3(II) überhaupt nicht benachbart sind, denn dort ist der linke Nachbar von **north** **northwest** und der rechte **northeast**.

Bei einer „runden“ Kurve ist der Übergang zwischen den einzelnen Richtungsrelationen langsam und gleichmäßig. Das bedeutet, daß keine benachbarte Richtungsrelation ausgelassen werden kann. Bei einer Diskretisierung in 8 Richtungsgebiete muß eine Links- oder Rechtskurve demzufolge von **forward** über **forward-left** oder **forward-right** gehen, bevor **left** oder **right** kommt. Bei einer eckigen Kurve wird die Richtungsrelation in der Mitte ausgelassen und es folgt auf **forward** direkt **left** oder **right**. Daraus folgt, daß bei einer Diskretisierung in 4 Richtungsgebiete lediglich ein „eckiger“ Übergang möglich ist, nämlich beim **backward-turn**. Für die SHAPE-Repräsentation bedeutet dies, daß eine Reihe von SHAPES, die schon in der Menge der basalen SHAPES enthalten sind, auch durch eine Klassifikation aus (anderen) basalen SHAPES entstehen können: Bei 8 Richtungsgebieten kann ein **left-turn** entweder direkt aus der Segmentierung kommen oder aus der Zusammenfassung von zwei **half-left-turns**. Dies ist aber genau der Unterschied zwischen einer eckigen Kurve (erste Möglichkeit) und einer runden (zweite Möglichkeit)! Bei 4 Richtungsgebieten ergibt die Unterscheidung zwischen „rund“ und „eckig“ wenig Sinn, da es für den einzigen eckigen basalen SHAPE (**backward-turn**) keine runde Entsprechung gibt.

Man kann aber für die Diskretisierung in 8 Richtungsgebiete ein neues Attribut *acuity* einführen, das für jeden basalen SHAPE wie folgt zu bestimmen ist: Wenn es sich um einen Richtungswechsel handelt, der in der QMV-Repräsentation in einen benachbarten Richtungsgebiet führt, so ist der dazugehörige basale SHAPE glatt (*smooth*). Handelt es sich jedoch um einen Richtungswechsel in einen nicht benachbarten Richtungsgebiet, so

ist der dazugehörige basale SHAPE spitz (acute), im Gegensatz zum gleichen SHAPE, der aus Klassifikation entstanden ist. Komplexe SHAPES sind dann glatt, wenn die beteiligten (basalen oder nicht basalen) SHAPES glatt sind. Ansonsten sind sie spitz.

Beispiel 5.1

Im Folgenden sind die Konstruktoraufrufe für die zur Verfügung stehenden basalen und komplexen SHAPES eines SHAPE-Vokabulars zur kognitiven Modellierung bei 8 Richtungsbereichen aufgelistet. Der erste Parameter ist ein Gewicht, das die Auffälligkeit des SHAPES für die Memory-Buffer-Klassifikation wiedergibt. Der zweite Parameter ist der Name des SHAPE. Der dritte Parameter ist die egozentrische Richtung der QMV-Repräsentation, die zu diesem SHAPE korrespondiert, falls es sich um einen basalen SHAPE handelt. Falls es kein basaler SHAPE ist, hat dieses Attribut den Wert „noBasicShape“. Der vierte Parameter schließlich ist die Acuity, die diesem SHAPE zugewiesen wird:

```
public static ShapePsy8 STANDSTILL =
    new ShapePsy8(0, "standstill", "zero", "zero");
public static ShapePsy8 STRAIGHT_LINE =
    new ShapePsy8(1, "straight_line", "forward", "smooth");
public static ShapePsy8 HALF_LEFT_TURN =
    new ShapePsy8(2, "half_left_turn", "forward_left", "smooth");
public static ShapePsy8 HALF_RIGHT_TURN =
    new ShapePsy8(2, "half_right_turn", "forward_right", "smooth");
public static ShapePsy8 LEFT_TURN =
    new ShapePsy8(3, "left_turn", "left", "sharp");
public static ShapePsy8 RIGHT_TURN =
    new ShapePsy8(3, "right_turn", "right", "sharp");
public static ShapePsy8 BACKWARD_TURN =
    new ShapePsy8(3, "backward_turn", "backward", "sharp");
public static ShapePsy8 U_TURN_RIGHT =
    new ShapePsy8(4, "u_turn_right", "noBasicShape", "zero");
public static ShapePsy8 U_TURN_LEFT =
    new ShapePsy8(4, "u_turn_left", "noBasicShape", "zero");
public static ShapePsy8 THREE_QUARTER_TURN_LEFT =
    new ShapePsy8(4, "three_quarter_turn_left", "backward_left", "sharp");
public static ShapePsy8 THREE_QUARTER_TURN_RIGHT =
    new ShapePsy8(4, "three_quarter_turn_right", "backward_right", "sharp");
public static ShapePsy8 LOOP =
    new ShapePsy8(5, "loop", "noBasicShape", "zero");
public static ShapePsy8 S_CURVE =
    new ShapePsy8(5, "s_curve", "noBasicShape", "zero");
```

Auch beim Attribut „Acuity“ gibt es wieder (wie schon bei der Klassifikation mit gleichen oder verschiedenen Größen) die Möglichkeit, bei der Klassifikation das Matching exakt oder fuzzy zu implementieren. Im ersten Fall matcht eine Regel nur dann, wenn alle Attribute genau übereinstimmen; im zweiten Fall wird weniger Gewicht auf genaue Übereinstimmung mancher Attribute gelegt.

Eine erweiterte Möglichkeit, die Acuity der SHAPES zu bestimmen, erfordert eine Vorbehandlung der QMV-Sequenz. Wenn in der QMV-Sequenz Teilsequenzen vorhanden sind, bei denen der Richtungswechsel durch alle benachbarten Richtungsbereiche führt, bei denen der mittlere QMV aber viel kürzer ist als die beiden angrenzenden, so kann dies durchaus auch als spitze Kurve aufgefaßt werden:

<far east> <far southeast> <far south> <far southwest> <far west>

repräsentiert einen gleichmäßigen Verlauf durch alle angrenzenden Richtungsbereiche, während in

<very-far east> <very-close southeast> <very-far south> <very-close southwest> <very-far west>

so große Längenunterschiede bei den beteiligten Vektoren vorhanden sind, daß man die kurzen Vektoren auch vernachlässigen kann und dann in der SHAPE-Repräsentation eine eckige Kurve enthält. Dazu muß man in einer Vorbearbeitung der QMV-Sequenz alle Tripel $q_1q_2q_3$ von QMVs ermitteln, die folgende Eigenschaften haben:

1. $q_1 \neq q_2 \neq q_3$
2. Die Richtungsrelationen bilden eine konzeptuelle Nachbarschaft im Sinne von [29]. Die Richtung R_2 von q_2 ist ein Nachbar der Richtung R_1 von q_1 und die Richtung R_3 von q_3 ist ein Nachbar von R_2 . Das bedeutet, daß, wenn man auf dem Orientierungsfadenkreuz wie in Abbildung 3.3 den Punkt pos_n , der sich zunächst im Intervall R_1 befindet, im (oder gegen den) Uhrzeigersinn rotiert, er erst in das Intervall R_2 und dann in das Intervall R_3 gelangt, ohne daß er dazwischen in andere Intervalle gerät.
3. Die Distanzen der QMVs verhalten sich folgendermaßen: D_2 von $q_2 \ll D_1$ von $q_1 \approx D_3$ von q_3 ,

Aus diesen Tripeln wird q_2 gelöscht. Die resultierende QMV-Sequenz wird dann wie beschrieben segmentiert und in eine SHAPE-Sequenz mit den entsprechenden Acuties umgewandelt.

5.3.4 Beispiele

Die Beispiel-Sequenz in Abbildung 5.6(III) wurde durch Segmentierung aus einer QMV-Sequenz erzeugt. Wenn wir annehmen, daß die QMV-Sequenz mit Diskretisierung in

acht Richtungsgebiete erzeugt wurde, entsteht durch Segmentierung eine basale Shape-Sequenz mit folgenden Acuties:

```
<straight-line smooth medium-size slow> <right-turn sharp medium-size  
slow> <right-turn sharp small very-slow> <right-turn sharp small very-  
slow> <straight-line smooth very-small very-slow> <left-turn sharp very-  
small very-slow> <left-turn sharp very-small very-slow> <straight-line  
smooth big medium-vel>
```

Eine stackbasierte Klassifikation mit der einfachsten Regelbasis 5.1 erzeugt folgende SHAPE-Sequenz:

```
<straight-line smooth medium-size slow> <right-turn sharp medium-size  
slow> <u-turn-right sharp small very-slow> <straight-line smooth very-  
small very-slow> <u-turn-left sharp very-small very-slow> <straight-line  
smooth big medium-vel>.
```

Eine Klassifikation mit derselben Regelbasis, aber einem Memorybuffer der Größe 5 ergibt:

```
<straight-line smooth medium-size slow> <right-turn sharp medium-size  
slow> <u-turn-right sharp small very-slow> <u-turn-left sharp very-small  
very-slow> <straight-line smooth big medium-vel>.
```

Es wurde der am wenigsten auffällige SHAPE, nämlich die sehr kurze *straight-line* zwischen den beiden *u-turns*, vergessen. Als Auffälligkeitsfunktion wurde hier lediglich die minimale Anzahl basaler SHAPES, aus denen ein SHAPE aufgebaut ist, gewichtet mit der Größe des SHAPE und seiner Acuity, implementiert. Andere Faktoren, wie z. B. Verweildauer im Buffer oder Primacy/Recency-Effekte, wurden nicht berücksichtigt.

Eine Klassifikation mit der Regelbasis 5.3 erzeugt hier das Ergebnis

```
<straight-line smooth medium-size slow> <right-turn sharp medium-size  
slow> <s-curve-right sharp small-decreasing very-slow> <straight-line  
smooth big medium-vel>.
```

6 Referenzsysteme

In früheren Kapiteln dieser Arbeit wurden bereits Referenzsysteme eingeführt. Das geschah z.B., wenn wir Richtungswechsel in einem Bewegungsverlauf mit Himmelsrichtungen wie „Nord“, „Süd“, „Ost“ und „West“ bezeichneten, oder mit Bezeichnern relativ zur eigenen Orientierung im Raum oder der bisherigen Bewegungsrichtung wie „Vorwärts“, „Rückwärts“, „Rechts“, „Links“. Doch das Problem der unterschiedlichen Referenzsysteme ist nicht nur ein Problem der unterschiedlichen Bezeichner, wie wir sehen werden.

Ein Referenzsystem bietet einen Bezugspunkt bzw. ein Koordinatensystem¹, relativ zu dem räumliche Objekte lokalisiert werden können. Das Problem, daß es davon mehrere unterschiedliche gibt, begegnet einem in vielen Wissenschaften. In [51] wird ein umfassender Überblick über die wichtigsten Konzeptionen von Referenzsystemen in Philosophie, Neurowissenschaften, Psychologie und Linguistik gegeben. Besonders verwirrend ist hierbei, daß über die Grenzen der Disziplinen hinweg, aber sogar innerhalb ein und derselben Disziplin, jeweils gleiche Konzepte unterschiedlich und unterschiedliche gleich bezeichnet werden können. Einige dieser Konzepte werden wir hier kurz erläutern, bevor wir erklären, welchen Sprachgebrauch wir in dieser Arbeit verwenden. Der Focus all dieser Konzepte ist auf der Verankerung von *Richtungsrelationen*. Auch in der Bewegungsrepräsentation sind unterschiedliche Referenzsysteme hauptsächlich bei der Beschreibung der Richtungen problematisch. Die Diskussion verschiedener Distanz-Referenzsysteme lassen wir deswegen außer Acht.

Relativ vs. absolut Die Unterscheidung zwischen relativem und absolutem Raum geht, wie schon in Kapitel 2 erwähnt, auf Newton und Leibniz zurück. Während für Newton der Raum ein abstrakter, unendlicher, unbeweglicher dreidimensionaler Container war (mit Ursprung in der Mitte des Universums), war Leibniz ein strikter Relativist: er sah Raum lediglich als System von Relationen zwischen Objekten an, wie übrigens die meisten Psychologen und Philosophen den psychologischen Primat des relativen Raumes konstatieren. Eine Ausnahme hiervon ist Kant, der bekann-

¹Über die Frage, ob für ein Referenzsystem ein Bezugspunkt ausreicht, oder ein Koordinatensystem nötig ist, ist in der Literatur viel diskutiert worden. Wir möchten nicht näher auf diese Diskussion eingehen. Wenn ein Referenzsystem auch ein Koordinatensystem beinhaltet, nennen wir es auch „Referenzrahmen“. Die meisten der in diesem Kapitel diskutierten Referenzsysteme sind Referenzrahmen. Wir verwenden trotzdem meist den allgemeineren Ausdruck „Referenzsystem“.

termaßen den absoluten Raum als a priori-Erkenntnis auffaßte. Diese Auffassung wird von Forschern wie O’Keefe und Nadel [67] unterstützt, die im Hippocampus eine Art absoluter Raumrepräsentation gefunden haben. Später wurde häufig der relative Raum mit dem egozentrischen Koordinatensystem assoziiert, und der absolute mit nicht-egozentrischen, so daß die Unterscheidung relativ-absolut oft mit der Unterscheidung egozentrisch-allozentrisch gleichgesetzt wird.

Egozentrisch vs. allozentrisch Dies bezeichnet die Unterscheidung zwischen Koordinatensystemen mit Ursprung im Körper des Menschen oder Roboters im Unterschied zu Koordinatensystemen, die ihren Ursprung anderswo haben. Diese Unterscheidung ist in den Neurowissenschaften gebräuchlich, wo viele verschiedene egozentrische Koordinatensysteme bekannt sind, z. B. für Kopf- und Armbewegungen, Saccaden, etc. Es ist allgemein akzeptiert, daß es auf dieser Ebene nötig ist, zwischen egozentrischem und allozentrischem Referenzsystem zu unterscheiden; diese Notwendigkeit ergibt sich schon aus dem bereits oben erwähnten Vorhandensein von nicht-egozentrischen Raumrepräsentationen im Hippocampus.

Deiktisch vs. intrinsisch Linguisten unterscheiden üblicherweise das deiktische vom intrinsischen Referenzsystem, um Mehrdeutigkeiten in Sätzen wie „Der Junge steht vor dem Auto“ benennen zu können – im deiktischen Bezugssystem liegt der Ursprung des Koordinatensystem beim Sprecher, im intrinsischen im Referenzobjekt. Hin und wieder wird hier ein drittes Bezugssystem eingeführt – das extrinsische –, das z. B. durch die Schwerkraft definiert wird und eine (weitere) mögliche Interpretation von Begriffen wie „über“ und „auf“ gibt. Leider wird der Begriff „deiktisch“ von Linguisten auch nicht immer einheitlich verwendet. In [51] werden drei mögliche Bedeutungen dafür identifiziert.

Levinson selbst verwirft nach einer genauen Analyse der unterschiedlichen Fälle die oben referierten Unterscheidungen und unterscheidet in [51] nur noch drei linguistische Referenzsysteme, das intrinsische, das relative und das absolute. Das **intrinsische** Referenzsystem ist objektzentriert, und Ursprung und Achsen des Koordinatensystems werden durch dem Referenzobjekt inhärente Eigenschaften definiert, wie z. B. die Vorderseite eines Hauses mit der Eingangstür. In diesem Referenzsystem bedeutet „Der Junge steht vor dem Auto“, daß er an dieser intrinsischen Vorderseite zu lokalisieren ist. Außer dem lokalisierten Objekt („Junge“) und Referenzobjekt („Auto“) sind keine weiteren Entitäten involviert; in diesem Referenzsystem werden ausschließlich zweistellige räumliche Relationen bezeichnet.

Im **relativen** Bezugssystem ist die Sache komplizierter. Hier werden grundsätzlich dreistellige Relationen bezeichnet, und zwar zwischen lokalisiertem Objekt, Referenzobjekt und einem Blickpunkt (z. B. der Sprecher), in dem das Koordinatensystem seinen Ursprung hat. In diesem Bezugssystem bedeutet „Der Junge steht vor dem Haus“, daß, vom

Blickpunkt aus gesehen, der Junge vor dem Haus steht, d. h. Teile des Hauses von dem Jungen verdeckt werden und nicht umgekehrt. Daß hier alle Relationen dreistellig sind, bedeutet nicht, daß immer drei Objekte oder Entitäten involviert sein müssen. Häufig fallen auch Blickpunkt und Referenzobjekt zusammen, z. B. in Sätzen wie „Der Ball ist vor Dir“ oder „Der Ball ist vor mir“.

Im **absoluten** Referenzsystem werden wiederum wie im intrinsischen zweistellige Relationen bezeichnet. Der Unterschied ist, daß es hier ein Koordinatensystem mit festen Achsen gibt, das nicht rotiert werden kann, und seinen Ursprung immer im Referenzobjekt hat. Diese Achsen (Richtungen) werden durch Gegebenheiten definiert, die aus der Umgebung kommen, z.B. Himmelsrichtungen, Schwerkraft, usw.

Diese Unterscheidung ist aus linguistischer Sicht sicher die sinnvollste und methodisch sauberste. Obwohl manche von Levinsons Darlegungen auch für unser Thema wichtig sind, wie z.B. die Überlegungen zum absoluten Referenzsystem, wie wir später noch sehen werden, stehen wir doch grundsätzlich vor anderen Problemen als die Linguisten. Levinsons Unterscheidungen beziehen sich nämlich auf eine Post-Hoc-Situation: Eine sprachliche Äußerung wird in ein bestimmtes Referenzsystem eingebettet, und je nachdem stehen verschiedene Referenzsysteme gleichwertig zur Auswahl, woraus sich Mehrdeutigkeiten ergeben. Auch bei der Generierung sprachlicher Äußerungen, die eine räumliche Anordnung beschreiben, muß man lediglich darauf achten, welche Mehrdeutigkeiten sich für den Empfänger der Nachricht ergeben.

Bei uns stellt sich die Situation anders dar: Nicht eine statische Situation wird beschrieben, sondern ein ganzer Bewegungsverlauf wird beobachtet (gemessen) und repräsentiert. Es stellt sich im Folgenden die Frage, welche Referenzsysteme hier zur Auswahl stehen, wodurch sie sich unterscheiden, und welche Implikationen die Verwendung des einen oder anderen Systems für die Repräsentation und die Weiterverarbeitung der Daten hat.

6.1 Egozentrisches versus allozentrisches Referenzsystem

Das relative Referenzsystem nach Levinson heißt bei uns „egozentrisch“, weil hier Blickpunkt und das sich bewegende Objekt („ego“) immer zusammenfallen. In dieser Nomenklatur bleibend, nennen wir das absolute Referenzsystem allozentrisch. Im egozentrischen Referenzsystem heißen Bewegungsrichtungen „Vorwärts“, „Rückwärts“, „Links“, „Rechts“, . . . , im allozentrischen „Nord“, „Süd“, „Ost“, „West“, Aber der Unterschied besteht nicht nur in verschiedenen Benennungen. Wie bereits bemerkt, ist der entscheidende Unterschied, daß im allozentrischen Referenzsystem ein fester Bezugspunkt vorhanden ist, wie z. B. der Nordpol oder der Polarstern, und das sich bewegende Individuum in der Lage ist, seine Orientierung relativ zu diesem Bezugspunkt zu messen. Zunächst möchten wir noch ein paar weitere Begriffe klären. Wir machen dabei zwei Annahmen, die im Zusammenhang mit Bewegungsrepräsentation vernünftig scheinen: 1.

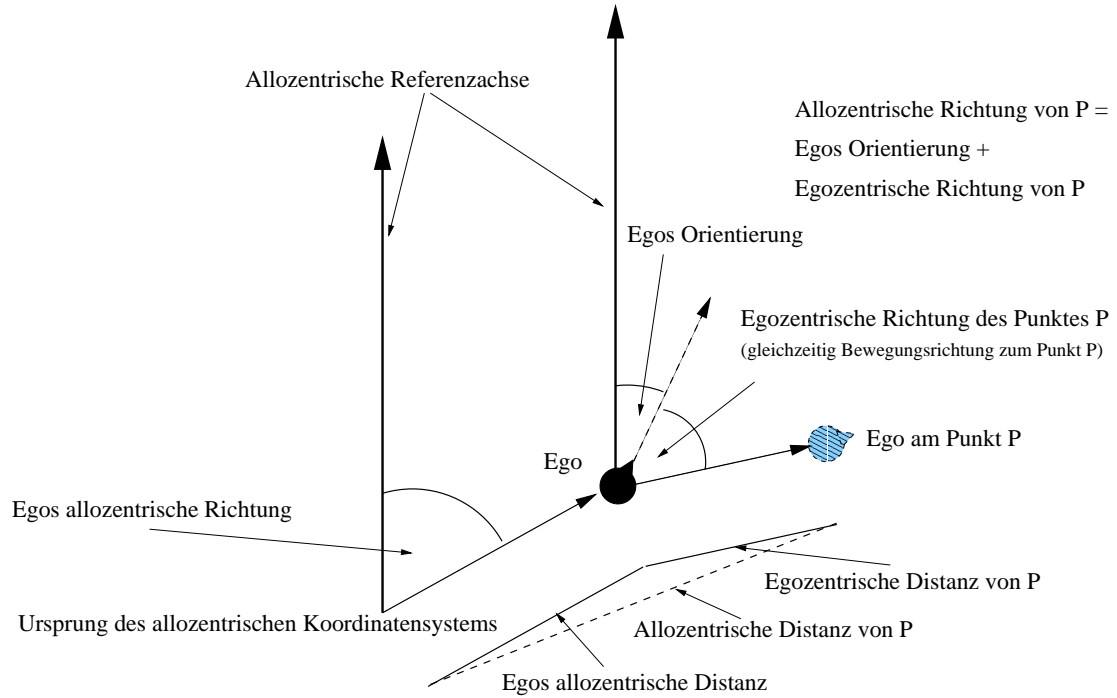


Abbildung 6.1: Illustration grundlegender Konzepte

Die sich bewegenden Objekte haben ein intrinsisches „Vorne“ und 2. sie bewegen sich bezüglich dieses Vorne normalerweise vorwärts (wie ein Motorrad).

Die meisten der folgenden Definitionen stützen sich auf die Arbeit von Klatzky [48].
Abbildung 6.1 illustriert die Begriffe.

Die Referenzachse ist eine gerichtete Achse, bezüglich der Orientierung und Richtung gemessen werden.

Die Orientierungsachse eines sich bewegenden Objekts zeigt von der intrinsischen Vorderseite aus nach vorn.

Die Orientierung eines Objekts im Raum ist der Winkel zwischen der Orientierungsachse und einer externen Referenzachse.

Die Richtung zwischen Punkt A und Punkt B ist der Winkel zwischen der Referenzachse und der Verbindungslinie zwischen A und B.

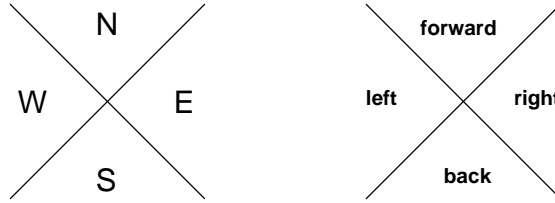


Abbildung 6.2: Allozentrische und egozentrische Diskretisierung der Richtungsdomäne

Wenn die Punkte A und B benachbarte Ecken eines einen Bewegungsverlauf beschreibenden Polygonzugs sind, ist die Richtung zwischen A und B natürlich auch die Bewegungsrichtung zwischen A und B.

Ob wir uns im egozentrischen oder allozentrischen Referenzsystem befinden, hängt von der Wahl der Referenzachse ab. Im allozentrischen Referenzsystem ist die Referenzachse außerhalb des sich bewegenden Objekts liegend und ändert sich nicht. Im egozentrischen Referenzsystem dagegen liegt die Referenzachse im sich bewegenden Objekt. Sie entspricht der Orientierungsachse des Objekts. Die allozentrische Orientierung des egozentrischen Referenzsystems ändert sich demzufolge ständig, wenn sich das Objekt im Raum bewegt. Daraus erwachsen auch die im Folgenden diskutierten Probleme mit diesem Referenzsystem.

6.2 Referenzsysteme in Messung versus Repräsentation

Zunächst scheint der Unterschied zwischen dem egozentrischen und dem allozentrischen Referenzsystem lediglich in den qualitativen Benennungen der Quantity Spaces zu liegen (siehe Abbildung 6.2). Wenn wir nur die Repräsentation von Bewegung betrachten, dann stimmt das auch: Die allozentrische QMV-Sequenz

```
<far north><close west><far west><close south><med-dist west>
```

ist äquivalent zur folgenden egozentrischen Sequenz:

```
<far forward><close left><far forward><close left><med-dist right>.
```

6.2.1 Probleme bei egozentrischer Messung

Wenn ein Bewegungsverlauf gemessen und direkt in eine QMV-Sequenz umgewandelt wird, liegt der Fall anders. Da im egozentrischen Referenzsystem keine absolute und unveränderliche Referenzachse vorhanden ist, muß das „Richtungsraaster“ auf jedem Meß-

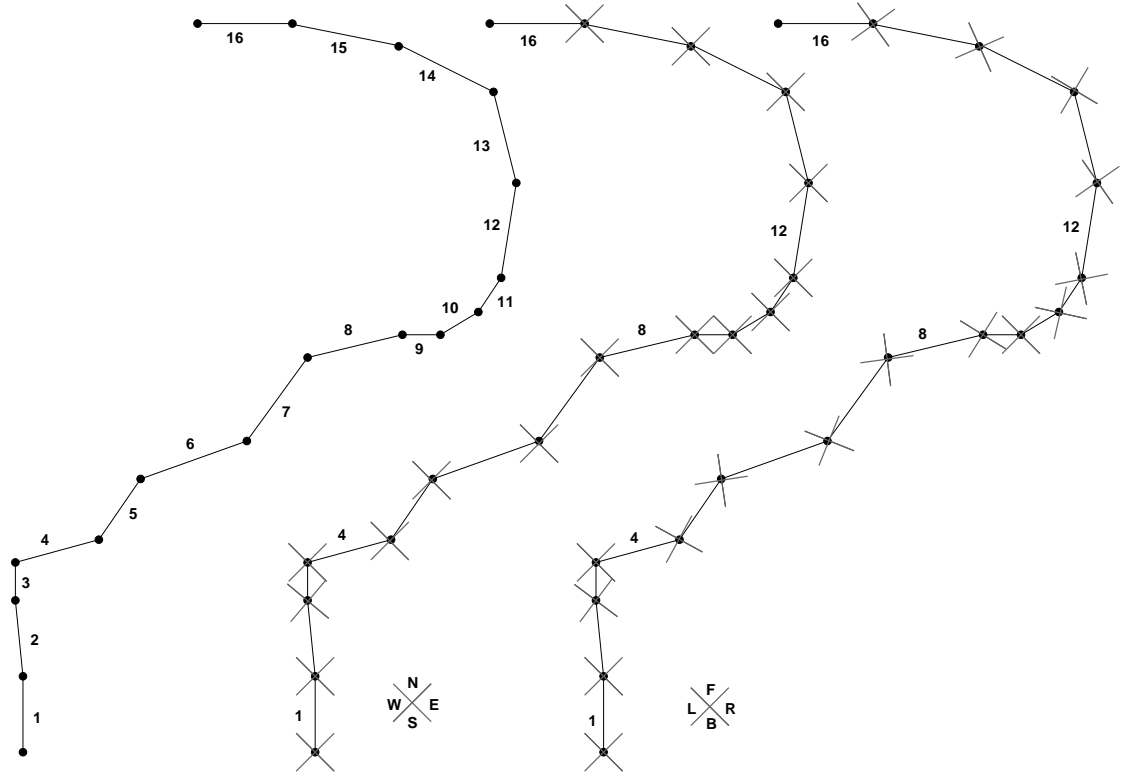


Abbildung 6.3: Messung im allozentrischen und egozentrischen Referenzsystem

punkt neu ausgerichtet werden (Abbildung 6.2 rechts), und zwar abhängig von der allozentrischen Orientierung des sich bewegenden Objekts (oder Person) im Raum. Die Mitte des Bereichs „Forward“ fällt dabei zusammen mit der Orientierungsachse des Objekts.

Abbildung 6.3 zeigt die Abbildung eines Bewegungsverlaufs (mit Vektoren v_1 bis v_{16}) auf eine qualitative Beschreibung mit egozentrischen (rechts) und allozentrischen (Mitte) Richtungen. Betrachtet man nur die Richtungen, entstehen so die folgenden Sequenzen:

allozentrisch: N, N, N, E, N, E, N, E, E, E, N, N, N, W, W, W

egozentrisch: F, F, F, R, F, F, F, F, F, F, F, F, F, L, F, F

Bis Vektor v_4 sind die Sequenzen äquivalent. Die Zickzackbewegung ab Vektor v_4 ist allerdings nur noch bei der allozentrischen Messung sichtbar. In der egozentrischen Messung wird die Zickzackbewegung nicht bemerkt, da die Abweichung von der Geradeausrichtung in jedem Meßpunkt kleiner als 45° ist. Noch schlimmer ist allerdings, daß auch die ganze Linkskurve zwischen v_9 und v_{13} unbemerkt bleibt. Lediglich der Richtungswechsel zwischen v_{13} und v_{14} taucht auch bei egozentrischer Messung auf, da hier der

Richtungswechsel in einem einzigen Meßpunkt groß genug war. Daß nur große Richtungswechsel in einem einzelnen Abtastzyklus bemerkt werden, nicht aber viele kleine, die sich in mehreren Abtastzyklen zu einem großen summieren, führt dazu, daß diese Art der egozentrischen Messung bei gleicher Bewegungsspur, aber unterschiedlicher Geschwindigkeit, extrem unterschiedliche Ergebnisse liefern kann.

Im allozentrischen Referenzsystem wird wegen des festen Referenzobjekts (z.B. Nordpol) und der dadurch definierten Referenzachse sowie der Fähigkeit, die eigene Orientierung relativ zu dieser Referenzachse zu messen, jeder große Richtungswechsel letztendlich bemerkt, egal in wievielen Abtastzyklen er vollzogen wird: Da sich das „Richtungsraaster“ nicht dreht (siehe Abbildung 6.3 Mitte), wird ein großer Richtungswechsel auf jeden Fall immer ein Überqueren der Grenze von einem Richtungsbereich zum benachbarten (also z. B. von „Norden“ nach „Osten“) bedeuten. Da sich im egozentrischen Referenzsystem dieses Raster ständig mit der Orientierung des Objekts im Raum dreht, werden langsame Richtungswechsel aufgrund des groben qualitativen Rasters nie bemerkt.

Dies ist natürlich ein besonders großes Problem, wenn der Bewegungsverlauf sofort auf qualitative Kategorien übertragen wird. Grundsätzlich ist dieses Phänomen allerdings auch bei numerischer Messung vorhanden, da diese natürlich nie ohne Meßfehler stattfindet, und sich dieser Fehler ähnlich wie im diskreten Fall auch im numerischen Fall fortpflanzt. Auch bei Menschen tritt dieses Phänomen auf: In der Wüste, wo man sich nicht an Landmarken orientieren kann, neigen Menschen dazu (sofern sie keinen Kompass haben und allozentrisch messen können), in großen Kreisen gegen den Uhrzeigersinn zu gehen, glauben aber, sie würden immer dieselbe allozentrische Richtung halten.

6.2.2 Mögliche Lösungen

Daß das Ergebnis der qualitativen Messung geschwindigkeitsabhängig ist, kann behoben werden, indem die Abbildung auf die qualitativen Kategorien nicht in jedem Taktzyklus sofort vorgenommen wird. Eine Möglichkeit ist z. B., diese Abbildung immer dann vorzunehmen, wenn eine festgelegte Distanz zurückgelegt wurde.

Bei dieser Art der Messung wird in jedem Meßzyklus die gleiche Distanz d zurückgelegt. Da man bei der egozentrischen Messung ja keine fixen Referenzobjekte zur Verfügung hat, können nur subjektive Richtungen wie Lenkeinschlag oder Drehwinkel gemessen werden. Die Abbildung in die Richtungsbereiche muß hier also von dem über die ganze Strecke summierten Wert in entsprechend definierte Intervalle stattfinden.

Die Geschwindigkeit wird auf die Zeit abgebildet, die benötigt wird, um diese fixe Strecke zurückzulegen. Ein Bewegungsverlauf wird dann durch QMVs repräsentiert, die aus einer egozentrischen Richtung D und der Anzahl von Taktzyklen t bestehen, die benötigt wurde, um die Distanz d zurückzulegen ($\langle t, D \rangle$), wobei der Zähler i hier wieder (wie schon in Abschnitt 3.2) bedeutet, daß für i Meßzyklen keine Veränderung in Richtung und Geschwindigkeit stattfand. Zum Beispiel:

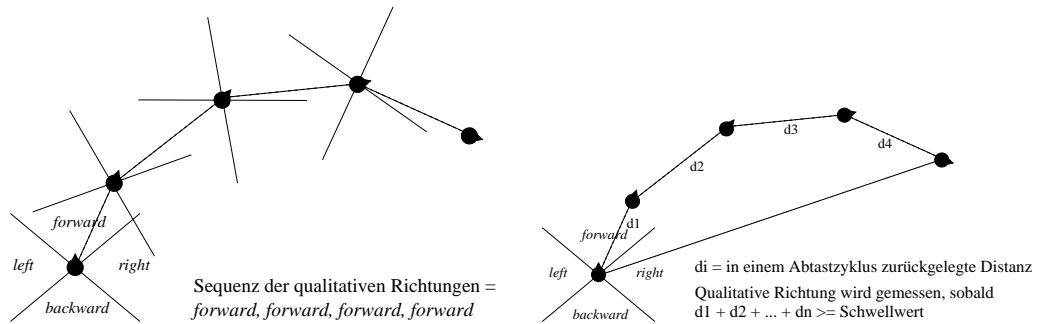


Abbildung 6.4: Links: Nichterkennen einer langsamen Kurve bei egozentrischer Messung
Rechts: Lösung

```
<6 forward>5 <6 left>2 <6 left>3 <6 left>1 <4 forward>6
<4 right>6.
```

Wenn die allozentrische Richtung des ersten Vektors bekannt ist, kann diese Sequenz natürlich in das allozentrische Referenzsystem transformiert werden. Wenn z. B. die allozentrische Richtung des ersten Vektors *east* war, ergibt sich:

```
<6 east>5 <6 north>2 <6 west>3 <6 south>1 <4 south>6 <4 east>6.
```

Bei qualitativer Repräsentation aller Komponenten erhält man auch bei dieser Art der Messung dieselben Sequenzen wie in Abschnitt 3.2. Wenn man z. B. annimmt, daß die Meßdistanz in das Intervall „close“ fällt, kann man die qualitativen Distanzen errechnen, indem man die Zählerinformation mit der Meßdistanz multipliziert (also z. B.: 5 mal die Meßdistanz vorwärts ergibt *medium-dist*). Die qualitativen Geschwindigkeiten ergeben sich direkt aus t : Je größer t , desto langsamer hat sich das Objekt bewegt; hier wird z. B. $t = 6$ auf *slow* abgebildet:

```
<medium-dist east slow> <close north slow> <close west slow>
<close south slow> <medium-dist south medium-vel> <medium-
dist east medium-vel>.
```

Wenn, wie bereits in Kapitel 5 vorgeschlagen, der Bewegungsverlauf zunächst generalisiert wird, bevor er in eine QMV-Sequenz umgewandelt wird, sollten die Probleme mit egozentrischer Messung nicht so gravierend sein, daß bedeutende Richtungswechsel nicht bemerkt werden, da durch die Generalisierung mit den numerischen Daten die kleineren Richtungswechsel sowieso unterdrückt werden.

6.2.3 Konsequenzen bei späteren Berechnungen

Auch wenn Anstrengungen unternommen werden, diese Probleme im egozentrischen Referenzsystem so klein wie möglich zu halten, treten sie natürlich, wie bereits ausgeführt, grundsätzlich immer auf.

Durch die Verzerrungen in den Richtungsrelationen kann es zu Problemen kommen, wenn die relative Position von zwei Objekten auf der Bewegungsspur zueinander erschlossen werden soll. Manche Inferenzen, die im allozentrischen Referenzsystem eindeutig sind, führen im egozentrischen Referenzsystem zu Mehrdeutigkeiten. Wenn z. B. eine egozentrisch gemessene QMV-Sequenz die Sequenz `left, left, left` enthält, so kann dies nicht ohne weiteres zu einem `right` vereinfacht werden, während im allozentrischen Fall eine Kurve von `north, east, south, west` sicherlich zu einer Sequenz `north, west` zusammengefaßt werden kann.

In einem Beispielszenario exploriert ein autonomer mobiler Roboter ein Gebäude und soll dabei eine Karte erstellen. Der Roboter verfügt nur über eine spärliche Sensorausstattung und kann nur wenig Information über seine Umgebung sammeln. Im wesentlichen kann er nur Korridore, Korridorkreuzungen und Türen erkennen. Durch Messung im egozentrischen Referenzsystem werden zwar die Distanzen zwischen diesen Landmarken einigermaßen korrekt gemessen, aber die Richtungen sind verzerrt. Was muß man beachten, wenn man den Roboter mit der Fähigkeit ausstatten will, aus den explorierten Routen eine Übersichtskarte zu erzeugen?

Die Spur von S1 nach Z1 in Abbildung 6.5 ist ähnlich gekrümmt wie die Spur in Abbildung 6.3, nur daß hier der eine scharfe Richtungswechsel aus der Spur in Abbildung 6.3 auch noch fehlt. Das heißt, daß bei egozentrischer Messung wie in Abbildung 6.3 die Krümmung des Korridors überhaupt nicht bemerkt wird. Die Sequenz der ermittelten Richtungswechsel zusammen mit den gemessenen Distanzen suggeriert, daß (wenn man annimmt, daß die Orientierung des Roboters am Punkt S1 nördlich war) Z1 nordöstlich von S1 liegt, während es in der Realität nordwestlich von S1 liegt.

T1 oben rechts in Abbildung 6.5 ist eine graphische Repräsentation der Spur, bei der Landmarken, Start und Ziel der Spur die Knoten eines Graphen sind und die relativen Richtungen und Distanzen zwischen diesen Knoten erhalten bleiben.

Wenn nun eine zweite Spur von S2 nach Z2 dazukommt, und man identifizieren kann, daß die beiden Spuren sich an den Punkten K1 und K2 treffen, was kann man dann inferieren? Wenn man versucht, die beiden Graphen lediglich an den Punkten K1 und K2 zu matchen, erhält man als Ergebnis, daß Z1 und Z2 identisch sind. Diese falsche Lösung ist in dem Graphen links unten in Abbildung 6.5 dargestellt.

Da bei der egozentrischen Messung Verzerrungen aufgetreten sind, muß man auch Verzerrungen beim Matchen solcher schematischer Repräsentationen zulassen. Dies ist rechts unten in Abbildung 6.5 dargestellt. Das bedeutet aber, daß man auf die Frage, wie die Umgebung strukturiert sein könnte, wesentlich mehr mögliche Lösungen erhält. Dies er-

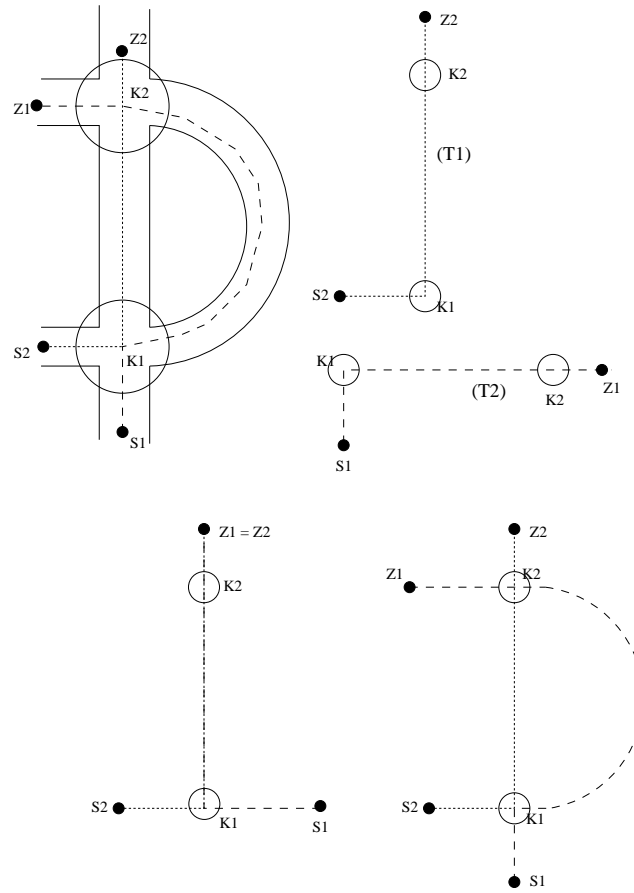


Abbildung 6.5: Oben links: Zwei Bewegungsspuren in einem Korridornetz. Oben rechts: Graph-Repräsentation mit Erhaltung der egozentrischen Richtungen. Unten links und unten rechts: Zwei mögliche Graph-Matchings.

höht die Komplexität von Algorithmen zur Constraint-Erfüllung und ähnlichem. Um die Zahl der möglichen Lösungen wieder einzuschränken, müssen deswegen bei egozentrischer Messung noch andere Informationsquellen zusätzlich herangezogen werden.

In unserem Beispiel könnten das noch zusätzliche Landmarken sein. Wenn z. B. auf den beiden verschiedenen Wegen von $K1$ nach $K2$ auch eine unterschiedliche Anzahl von Türen liegt, könnte durch diese Zusatzinformation die falsche Lösung eliminiert werden. Dazu müssen diese Landmarken allerdings sehr zuverlässig erkannt werden. Eine andere Möglichkeit in unserem Beispiel wäre, zu bemerken, daß die Kreuzung $K1$ *nicht* aus entgegengesetzten Richtungen angesteuert wurde, so daß die falsche Lösung unten links in Abbildung 6.5 nicht in Frage kommt.

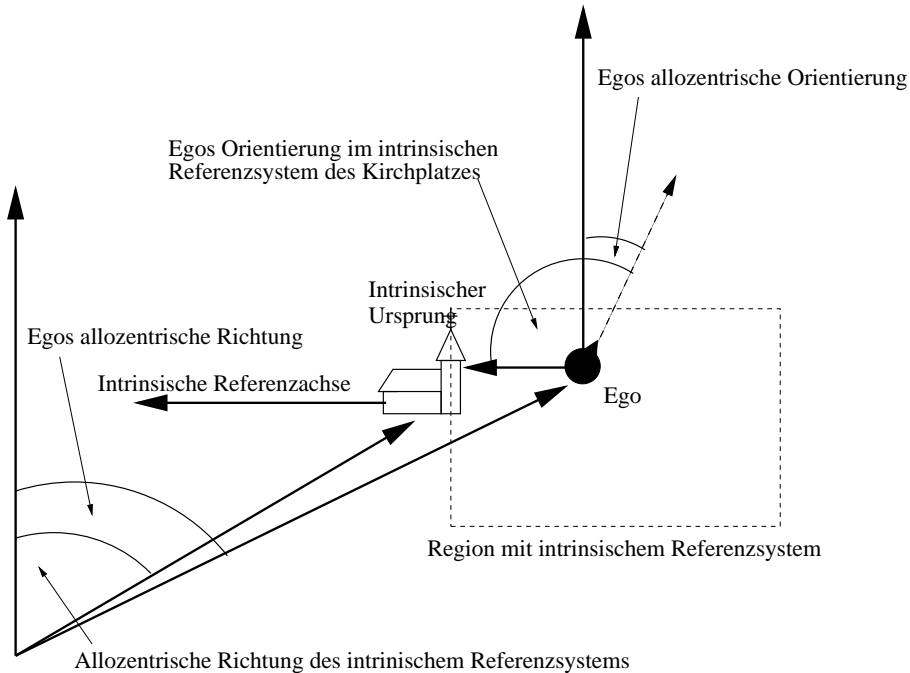


Abbildung 6.6: Zusammenhang zwischen intrinsischem und allozentrischem Referenzsystem

6.3 Lokale Referenzsysteme

Die wenigsten Landmarken der wirklichen Welt sind unstrukturierte Punkte; die meisten haben im Gegenteil eine beträchtliche räumliche Ausdehnung und eine Struktur, die es ihnen erlaubt, eigene Koordinatensysteme zu definieren. So hat z. B. eine Kirche eine intrinsische Vorderseite und definiert somit intrinsische Referenzachse und -rahmen.

Diese Referenzsysteme sind fest und invariabel verankert und funktionieren somit für die Repräsentation von Bewegungsverläufen wie allozentrische Referenzsysteme, was sie de facto auch sind. Die Umrechnung zwischen zwei allozentrischen Referenzsystemen entspricht einer Koordinatentransformation und ist unproblematisch. Für die Umwandlung vom egozentrischen Referenzsystem in ein intrinsisches gilt genau dasselbe, was wir schon für die Umrechnung vom egozentrischen in das allozentrische Referenzsystem festgestellt haben.

Aber auch Landmarken, die unstrukturiert sind wie z. B. der Obelisk in München, können für einen Bewegungsverlauf ein Referenzsystem definieren. Diese Art von Landmarken hat keine Referenzachse, aber definiert einen Referenzpunkt, auf den sich Richtungsan-

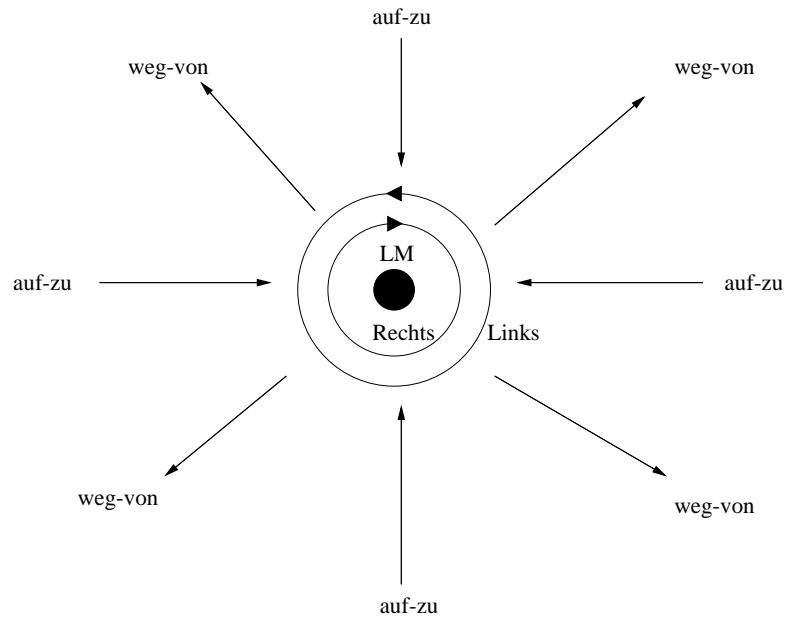


Abbildung 6.7: Ungerichtetes Referenzsystem

gaben beziehen können.

Solche ungerichteten Referenzsysteme, wie in [Abbildung 6.7](#) dargestellt, sind rotationsinvariant und definieren im Prinzip nur topologische Relationen, da man ein statisches links und rechts nicht unterscheiden kann. Wenn man zur Beschreibung eines Bewegungsverlaufs solch ein Referenzsystem benutzt, ist es wichtig, den Bewegungsverlauf noch anderweitig zu verankern. „Auf den Obelisk zu und dort rechts“ beschreibt jeden Bewegungsverlauf, der sich aus beliebiger Himmelsrichtung an den Obelisk annähert und dort rechts abbiegt. Um die Beschreibung eindeutig zu machen, müssen zusätzlich andere Hilfsmittel benutzt werden, die diese Möglichkeiten einschränken: z. B. allozentrische Richtungen („Von Norden auf den Obelisk zu . . .“) oder weitere Landmarken („Auf der Barer Str. Richtung Obelisk . . .“).

Bei der Benutzung lokaler Referenzsysteme ist es wichtig, anzugeben, ob es sich um ein intrinsisches Koordinatensystem oder ein ungerichtetes Referenzsystem handelt. Während die Umrechnung eines egozentrisch oder allozentrisch repräsentierten Bewegungsverlaufs in ein intrinsisches Referenzsystem lediglich eine Koordinatentransformation darstellt, erfordert die Umrechnung in ein ungerichtetes Referenzsystem wegen der Notwendigkeit zusätzlicher Verankerung mehr Aufwand. Wir beschränken uns in dieser Arbeit auf die Benutzung intrinsischer lokaler Referenzsysteme.

7 Ähnlichkeit und Komplexität

7.1 Ähnlichkeit von Bewegungsverläufen

7.1.1 Ähnlichkeit von QMV-Sequenzen

Die in dieser Arbeit beschriebenen Mechanismen, von Details von Trajektorien zu abstrahieren und sie grobgranular zu repräsentieren, erlauben die Definition von Ähnlichkeitsmaßen aufbauend auf den beschriebenen Repräsentationen und Algorithmen:

Shape-Ähnlichkeit Zwei QMV-Sequenzen sind SHAPE-ähnlich, wenn dieselbe SHAPE-Repräsentation aus ihnen erzeugt werden kann.

ε -Ähnlichkeit: Zwei QMV-Sequenzen sind ε -ähnlich, wenn ε der kleinste Faktor ist, der bei ε -Generalisierung zu derselben QMV-Sequenz führt.

SHAPE-Ähnlichkeit basiert natürlich ganz wesentlich darauf, welches ε für die Generalisierung benutzt wird, d. h. daß die ε -Ähnlichkeit das genauere Maß ist. Die SHAPE-Ähnlichkeit ist orientierungsunabhängig, denn SHAPE-Repräsentation ist orientierungsunabhängig.

Bezüglich ε -Ähnlichkeit sind zwei QMV-Sequenzen umso ähnlicher, je kleiner ε ist. Identische QMV-Sequenzen sind 0-ähnlich. Es kann natürlich vorkommen, daß, selbst wenn man ε gegen unendlich gehen läßt, die zu vergleichenden Sequenzen nie gleich werden. Dann sind die Sequenzen unähnlich.

Beiden Maßen ist gemeinsam, daß sie für praktische Zwecke weitgehend unbrauchbar sind, da es nicht praktikabel ist, beide Sequenzen mit immer größeren Faktoren zu generalisieren um zu sehen, ob die am Ende die gleichen Sequenzen herauskommen.

Ein Vergleich von QMV-Sequenzen kann deshalb nur komponentenweise vor sich gehen. In Kapitel 4.2 wurde die Problematik angesprochen, daß durch die Diskretisierung des Raums ähnliche Bewegungsverläufe möglicherweise sehr unähnlich repräsentiert werden. Wenn man entscheiden möchte, ob zwei QMV-Sequenzen möglicherweise einen ähnlichen Bewegungsverlauf repräsentieren, muß man recht großzügige Kriterien anlegen. So kann sich z. B. der numerische Vektor, der durch `<very-close east>` repräsentiert wird, von dem Vektor, der durch `<close south>` repräsentiert wird, nur minimal unterscheiden.

Betrachten wir im Folgenden zunächst Vergleiche der Spur. Dazu faßt man alle aufeinanderfolgenden QMVs der gleichen Richtung, die lediglich Geschwindigkeitsänderungen enthalten, zu einem einzigen QMV zusammen. Bei generalisierten QMV-Sequenzen ist das automatisch der Fall.

Allozentrische Ähnlichkeit

Definition 7.1

Zwei QMVs, deren allozentrische Richtung und Distanz um nicht mehr als ein Intervall voneinander abweichen, heißen allozentrisch ähnlich.

Zwei QMV-Sequenzen, deren QMVs komponentenweise allozentrisch ähnlich sind, heißen ebenfalls allozentrisch ähnlich.

Man bemerkt sofort die Problematik: Wenn jede Komponente der QMV-Sequenzen jeweils um ein Intervall abweicht, dann können die ursprünglichen Sequenzen zwar so ähnlich gewesen sein wie in Abbildung 4.3, aber es ist auch möglich, daß sie völlig unähnlich waren. Weil wir das nicht wissen, müssen wir auch solche QMV-Sequenzen noch als ähnlich akzeptieren.

Ein Problem beim Vergleich von aus ε -generalisierten Bewegungsverläufen entstandenen QMV-Sequenzen ist, daß kleine Abweichungen im gemessenen Bewegungsverlauf teilweise eine Generalisierung in relativ unterschiedliche Segmente bewirken können. Wenn die Originalspur z. B. ein Segment aufweist, das ungefähr so lang ist wie der Generalisierungsfaktor, kann dieses Segment bei minimalen Abweichungen von der Originalspur einmal als eigenständiges Segment generalisiert werden und als eigener QMV repräsentiert werden, oder aber einem angrenzenden Segment zugerechnet werden. Dieses Problem trat in einer Anwendung auf und ist in Abbildung 8.3 dargestellt.

Auch solchermaßen entstandene QMV-Sequenzen sollen natürlich noch als allozentrisch ähnlich eingeschätzt werden. Wenn es also beim komponentenweisen Vergleich von zwei bis dahin ähnlichen generalisierten QMV-Sequenzen dazu kommt, daß ein sehr kurzer Vektor mit einem viel längeren verglichen wird, so ist der kurze Vektor zu überspringen und der längere muß mit dem nächsten Vektor der anderen Sequenz verglichen werden. Ergibt dieses Verfahren, daß alle verglichenen QMVs allozentrisch ähnlich sind, so heißen die QMV-Sequenzen *generalisiert allozentrisch ähnlich*.

Egozentrische Ähnlichkeit

Für die egozentrischen Richtungsangaben gilt dasselbe Prinzip wie für die allozentrischen Richtungen:

Definition 7.2

Zwei QMVs, deren egozentrische Richtung und Distanz um nicht mehr als ein Intervall

voneinander abweichen, heißen egozentrisch ähnlich.

Zwei QMV-Sequenzen, deren QMVs komponentenweise egozentrisch ähnlich sind, heißen ebenfalls egozentrisch ähnlich.

Die egozentrische Ähnlichkeit ist orientierungsunabhängig. Identische Bewegungsverläufe, die in verschiedene allozentrische Richtungen gestartet wurden, sind nicht mehr allozentrisch ähnlich, aber sehr wohl egozentrisch ähnlich.

Bei dem im vorausgegangenen Abschnitt beschriebenen Problem, das durch unterschiedliche Generalisierungen entstehen kann, muß man bei der egozentrischen Ähnlichkeit allerdings anders vorgehen: Hier reicht es nicht, beim Vergleich das kurze Segment einfach auszulassen. Stattdessen muß man die egozentrischen Richtungen des kurzen und des nächsten langen Segments kombinieren. Wenn z. B. ein Bewegungsverlauf q_1 aus den QMVs `<far forward>` `<very-close forward-left>` `<very-far forward-left>` besteht, und ein Bewegungsverlauf q_2 aus den QMVs `<very-far forward>` `<very-far left>`, so sind diese als *generalisiert egozentrisch ähnlich* zu bewerten, denn `forward-left` und `forward-left` ergibt `left`.

Vergleich der Dynamik

Egozentrische und allozentrische Ähnlichkeit können um einen Vergleich der Dynamik erweitert werden.

Definition 7.3

Zwei QMVs, die egozentrisch (allozentrisch) ähnlich sind, und deren Geschwindigkeiten um nicht mehr als ein Intervall voneinander abweichen, heißen dynamisch egozentrisch (allozentrisch) ähnlich.

Zwei QMV-Sequenzen, deren QMVs komponentenweise dynamisch egozentrisch (allozentrisch) ähnlich sind, heißen ebenfalls dynamisch egozentrisch (allozentrisch) ähnlich.

Ergebnisse

Beispiel 7.1

q_1 und q_2 sind QMV-Sequenzen mit 8 Richtungsbereichen und 5 Distanzbereichen.

q_1 : `<very-far east slow>` `<very-far south fast>` `<far west fast>`

q_2 : `<far north fast>` `<far east slow>` `<very-far south very-fast>`

q_1 und q_2 sind nicht allozentrisch ähnlich, da die Richtungen der QMVs jeweils um mehr ein Intervall voneinander abweichen.

Rechnet man die QMV-Sequenzen in das egozentrische Referenzsystem um, ergibt sich folgendes:

q_1 : `<very-far forward slow>` `<very-far right fast>` `<far right fast>`

q_2 : <far forward fast> <far right slow> <very-far right very-fast>

Nun sind bei beiden QMV-Sequenzen die egozentrischen Richtungen identisch. Die Distanzen weichen jeweils nicht weiter als ein Intervall voneinander ab, was bedeutet, daß q_1 und q_2 egozentrisch ähnlich sind.

Vergleicht man die Geschwindigkeiten der Vektoren in q_1 und q_2 , stellt man fest, daß sie nicht dynamisch ähnlich sind.

Das Beispiel macht deutlich, daß die orientierungsunabhängige egozentrische Ähnlichkeit am ehesten geeignet ist, Bewegungsspuren nur bezüglich ihrer Form zu vergleichen. Egozentrisch ähnliche Bewegungsspuren ergeben oft ähnliche SHAPE-Sequenzen, egozentrisch unähnliche QMV-Sequenzen ergeben in jedem Fall unähnliche SHAPE-Sequenzen. Allozentrische Ähnlichkeit ist nicht orientierungsunabhängig und sollte deswegen da eingesetzt werden, wo es darum geht, zu vergleichen, ob zwei Bewegungsspuren eine ähnliche Route darstellen könnten. Es ist auch möglich, hier egozentrische Ähnlichkeit mit einem Vergleich der Ausgangsorientierungen zu benutzen, was im Grunde dasselbe ist. In der Anwendung in Abschnitt 8.1.1 ist das im Prinzip so gemacht worden.

Allgemein ist allerdings die QMV-Repräsentation für Vergleiche eher ungeeignet. Wenn man nicht das Risiko eingehen will, daß die QMV-Repräsentation sehr ähnlicher Sequenzen beim Vergleich als unähnlich eingestuft wird, muß man beim Vergleich von QMV-Sequenzen Sequenzen, die möglicherweise sehr unähnlich sind, als ähnlich akzeptieren.

7.1.2 Fuzzy-Ähnlichkeit

Die FMV-Repräsentation, wie sie in Abschnitt 4.2 eingeführt wurde, ist für Vergleiche besser geeignet. Hier kann man nicht nur entscheiden, ob die Sequenzen ähnlich sind oder nicht, sondern auch ein Abstandsmaß definieren.

Jede Komponente des FMV definiert den Anteil, zu dem der Vektor in eine bestimmte Kachel der Parkettierung fällt. Als Abstandsmaß zwischen zwei Vektoren kann man einfach die komponentenweise Differenz nehmen:

Definition 7.4

Seien

$$A_1 = (\mu_{(Dist_{1_1}, Dir_{1_1})}, \dots, \mu_{(Dist_{1_1}, Dir_{1_m})}, \dots, \mu_{(Dist_{1_n}, Dir_{1_1})}, \dots, \mu_{(Dist_{1_n}, Dir_{1_m})})^T$$

und

$$A_2 = (\mu_{(Dist_{2_1}, Dir_{2_1})}, \dots, \mu_{(Dist_{2_1}, Dir_{2_m})}, \dots, \mu_{(Dist_{2_n}, Dir_{2_1})}, \dots, \mu_{(Dist_{2_n}, Dir_{2_m})})^T$$

zwei $n \cdot m$ -FMV. Dann heißt

$$\mathcal{D}_{\text{FMV}}(A_1, A_2) = \sum_{i=1}^n \sum_{j=1}^m |\mu_{(\text{Dist}_{1_i}, \text{Dir}_{1_j})} - \mu_{(\text{Dist}_{2_i}, \text{Dir}_{2_j})}|$$

der Abstand zwischen A_1 und A_2 .

Der minimale Abstand zwischen zwei FMVs ist 0 (Identität), der maximale Abstand ist 2: Die $\mu_{(\text{Dist}_i, \text{Dir}_j)}$ in jedem FMV addieren sich zu 1. Wenn in jedem Vektor genau die Komponenten ungleich 0 sind, die im anderen 0 sind, dann ist der Abstand $\mathcal{D}_{\text{FMV}} = 2$ und damit maximal.

Definition 7.5

Seien $S_1 = A_1 \dots A_n$ und $S_2 = B_1 \dots B_n$ Sequenzen vom FMVs. Dann heißt

$$\mathcal{D}_{\text{FMVS}}(S_1, S_2) = \sum_{j=1}^n \mathcal{D}_{\text{FMV}}(A_j, B_j)$$

der Abstand zwischen S_1 und S_2 .

Der maximale Abstand zwischen zwei FMV-Sequenzen ist $2n$ wobei n die Länge der Sequenzen ist. Der minimale Abstand ist 0 (Identität).

Beispiel 7.2

A_1, A_2, A_3 sind $3 \cdot 2$ -FMVs.

$A_1 = (0, 0, 0, 3, 0, 7, 0, 0)^T$, $A_2 = (0, 3, 0, 7, 0, 0, 0, 0)^T$. $\mathcal{D}_{\text{FMV}}(A_1, A_2) = 0, 3 + 0, 7 + 0, 3 + 0, 7 + 0 + 0 = 2$.

$A_3 = (0, 0, 0, 2, 0, 5, 0, 3, 0)^T$; $\mathcal{D}_{\text{FMV}}(A_1, A_3) = 0 + 0 + 0, 1 + 0, 2 + 0, 3 = 0, 6$.

Manche FMVs mit Abstand 2 können allerdings ähnlicher sein als andere: Der Abstand 2 sagt nur, daß die Zugehörigkeitswerte ungleich 0 alle auf unterschiedlichen Kacheln liegen, aber nicht, wie weit diese Kacheln voneinander weg sind. Bei FMVs mit Abstand 2 könnte man deswegen noch zusätzlich den „metrischen“ Abstand der belegten Kacheln betrachten. Dazu muß man zunächst eine Metrik auf der Parkettierung des Raumes definieren, was ein Punkt für eine zukünftige Ergänzung der Repräsentation ist.

7.1.3 Ähnlichkeit von Shape-Sequenzen

SHAPE-Sequenzen sind rotationsinvariant und deswegen orientierungsunabhängig: Egal in welcher Richtung ein Bewegungsverlauf im Raum orientiert ist, es wird mit denselben Algorithmen immer dieselbe SHAPE-Sequenz erzeugt. Ein Vergleich von zwei SHAPE-Sequenzen kann deswegen immer nur orientierungsunabhängige Ergebnisse liefern.

SHAPE-Sequenzen kann man sehr akribisch vergleichen oder nur sehr oberflächlich. Der oberflächlichste Vergleich berücksichtigt nur die Abfolge der SHAPES, nicht aber die Attribute, die zu den SHAPES gehören. Dieser Vergleich würde die Gleichheit recht unterschiedlicher SHAPE-Sequenzen liefern; hier wären die Sequenzen

```
<straight-line smooth medium-size slow> <right-turn sharp medium-size  
slow> <u-turn-right sharp small very-slow> <u-turn-left sharp very-small  
very-slow> <straight-line smooth big medium-vel>
```

und

```
<straight-line smooth small slow> <right-turn smooth big very-fast>  
<u-turn-right smooth medium-size> <u-turn-left smooth big slow>  
<straight-line smooth big fast>
```

gleich.

Der akribischste Vergleich zweier SHAPE-Sequenzen stuft dieselben nur dann als gleich ein, wenn alle Attribute der SHAPES gleich sind.

Dazwischen ist natürlich jede Untermenge der Attribute, die man bei SHAPE-Sequenzen vergleichen möchte, möglich.

7.2 Komplexität von Bewegungsverläufen

Die *Beschreibungskomplexität* von Bewegungsverläufen bezieht sich natürlich immer auf die Repräsentation, in der diese Bewegungsverläufe dargestellt werden und ist ein Maß dafür, wie einfach oder kompliziert die Beschreibung des Bewegungsverlaufs in der gegebenen Repräsentation ist. Wir beschäftigen uns im Folgenden mit Komplexitätsmaßen für Bewegungsverläufe in QMV- und SHAPE-Repräsentation. Die *kognitive Komplexität* von Bewegungsverläufen bezieht sich darauf, wie einfach oder schwierig die Bewegungsverläufe vom menschlichen Wahrnehmungsapparat aufzunehmen, zu verarbeiten und wiederzugeben sind. Dies hängt natürlich nicht nur vom Bewegungsverlauf selbst ab, sondern wesentlich auch davon, wie Bewegungsverläufe in der menschlichen Kognition verarbeitet, repräsentiert und gespeichert werden.

7.2.1 Beschreibungskomplexität von QMV-Sequenzen

Bei QMV-Sequenzen sind die einzigen Beschreibungsmerkmale, die uns zur Verfügung stehen, die Änderungen der Richtung und die Änderungen der Geschwindigkeit. Da eine QMV-Sequenz, bei der sich keins von beiden je ändert, nur aus einem einzigen Vektor besteht und neue Vektoren erst dann erzeugt werden, wenn tatsächlich eine Änderung beobachtet wurde, ist eine QMV-Sequenz insgesamt betrachtet umso komplexer, je mehr Vektoren sie hat. Jeder Vektor kann nur aus einem Richtungs- oder Dynamikwechsel

entstanden sein, weswegen die absolute Anzahl von Vektoren $\mathcal{A}(q)$ ein erstes Maß dafür gibt, wie „kompliziert“ eine QMV-Sequenz q ist.

Nun ist es für eine QMV-Sequenz, die eine sehr kurze Trajektorie repräsentiert, eher wahrscheinlich, daß sie aus wenigen Vektoren besteht, als für eine lange; aber ist diese kurze QMV-Sequenz mit unwesentlich weniger Vektoren wirklich weniger komplex als die lange? Um ein Maß für die inhärente Komplexität \mathcal{I} einer QMV-Sequenz q zu erhalten, muß $\mathcal{A}(q)$ auf die absolute Länge von q (in Vielfachen von `very-close`) normiert werden. Der Faktor von 100 dient nur dazu zu verhindern, daß die Zahlen zu klein werden:

$$\mathcal{I}(q) = \frac{\mathcal{A}(q)}{\text{Länge von } q} \times 100$$

Vergleicht man die Komplexität zweier Bewegungsverläufe mit Hilfe dieses Maßes, muß man natürlich beachten, daß man so nur die Komplexität der *Repräsentationen* vergleicht und nicht die der tatsächlich stattgefundenen Bewegungsverläufe. Wenn zwei Bewegungsverläufe mit unterschiedlichen Abtastraten gemessen und als QMV-Sequenzen repräsentiert werden, können sehr unterschiedliche Komplexitäten herauskommen, obwohl die ursprünglichen Bewegungsverläufe vielleicht ganz ähnlich waren: Wenn ein Bewegungsverlauf alle 30ms seine Richtung ändert, und einmal mit einer Abtastrate von 30ms und einmal mit einer Abtastrate von 90ms gemessen wird, dann wird die QMV-Repräsentation im ersten Fall natürlich komplexer ausfallen als im zweiten Fall. Kennt man die Abtastraten, so kann man sie im Vergleich mit einbeziehen: Wenn z. B. der Bewegungsverlauf q_1 mit einer Abtastrate von 30ms, der Bewegungsverlauf q_2 mit einer Abtastrate von 90ms gemessen wurde, dann darf man nicht $\mathcal{I}(q_1)$ direkt mit $\mathcal{I}(q_2)$ vergleichen, sondern vielmehr $\mathcal{I}_{q_2}(q_1) = \mathcal{I}(q_1) \cdot \frac{30}{90} = \mathcal{I}(q_1) \cdot \frac{1}{3}$ und $\mathcal{I}(q_2)$.

$\mathcal{I}(q)$ und $\mathcal{A}(q)$ geben beide die Gesamtkomplexität der QMV-Sequenz bezüglich Richtung *und* Dynamikwechseln an. Diese Faktoren können aber auch getrennt interessant sein, wenn man sich z. B. für die räumliche Komplexität unabhängig von der dynamischen interessiert. Deswegen können diese Maße in die Komponenten $\mathcal{A}_d(q)$ für die dynamische Komplexität und $\mathcal{A}_s(q)$ für die räumliche Komplexität ($\mathcal{I}(q)$ analog) aufgesplittet werden.

Dabei werden für $\mathcal{A}_d(q)$ alle Vektoren gezählt, bei denen sich die Geschwindigkeit im Vergleich zum vorherigen Vektor geändert hat, und für $\mathcal{A}_s(q)$ alle Vektoren, bei denen sich die Richtung im Vergleich zum vorherigen Vektor geändert hat. $\mathcal{A}_{ds}(q)$ sei die Anzahl der Vektoren, bei denen sich im Vergleich zum vorherigen Vektor Geschwindigkeit *und* Richtung geändert haben.

Es gilt: $\mathcal{A}(q) = \mathcal{A}_d(q) + \mathcal{A}_s(q) - \mathcal{A}_{ds}(q) + 1$

Beispiel 7.3

Folgende QMV-Sequenzen q_1 und q_2 wurden mit Distanzintervallen, bei denen das Nachfolgerintervall jeweils 4mal so groß ist wie das Vorgängerintervall, erzeugt, d. h. `close` =

$4 \times \text{very-close}; \text{medium-dist} = 4 \times \text{close} = 16 \times \text{very-close}; \text{ usw:}$

q_1 : $\langle \text{close slow east} \rangle \langle \text{medium-dist medium-vel east} \rangle \langle \text{medium-dist medium-vel northeast} \rangle \langle \text{medium-dist medium-vel north} \rangle \langle \text{far medium-vel west} \rangle \langle \text{medium-dist medium-vel south} \rangle \langle \text{medium-dist medium-vel southeast} \rangle \langle \text{far fast east} \rangle \langle \text{medium-dist medium-vel northeast} \rangle \langle \text{medium-dist medium-vel north} \rangle \langle \text{far fast north} \rangle \langle \text{medium-dist medium-vel north} \rangle \langle \text{very-close very-slow northwest} \rangle \langle \text{close slow west} \rangle$

$$\mathcal{A}(q_1) = 14$$

$$\mathcal{I}(q_1) = \frac{14}{4+16+16+16+64+16+16+64+16+16+64+16+16+64+16+1+4} \times 100 = \frac{14}{329} \times 100 = 4,26$$

$$\mathcal{A}_s(q_1) = 10; \mathcal{I}_s(q_1) = \frac{10}{329} \times 100$$

$$\mathcal{A}_d(q_1) = 7; \mathcal{I}_d(q_1) = \frac{7}{329} \times 100$$

$$\mathcal{A}_{ds}(q_1) = 4; \mathcal{I}_{ds}(q_1) = \frac{4}{329} \times 100$$

$$\mathcal{A}(q_1) = 14 = 10 + 7 - 4 + 1 = \mathcal{A}_s(q_1) + \mathcal{A}_d(q_1) - \mathcal{A}_{ds}(q_1) + 1$$

$$\mathcal{I}(q_1) = \frac{14}{329} \times 100 = \frac{10+7-4+1}{329} \times 100 = \mathcal{I}_s(q_1) + \mathcal{I}_d(q_1) - \mathcal{I}_{ds}(q_1) + \frac{1}{\text{Länge von } q_1} \times 100$$

q_2 : $\langle \text{very-far fast northeast} \rangle \langle \text{medium-dist medium-vel east} \rangle \langle \text{medium-dist slow northeast} \rangle \langle \text{close very-slow northeast} \rangle$

$$\mathcal{A}(q_2) = 4$$

$$\mathcal{I}(q_2) = \frac{4}{256+16+16+4} \times 100 = \frac{4}{292} \times 100 = \frac{4}{292} \times 100 = 1,37$$

$$\mathcal{A}_s(q_2) = 2; \mathcal{I}_s(q_2) = \frac{2}{292} \times 100$$

$$\mathcal{A}_d(q_2) = 3; \mathcal{I}_d(q_2) = \frac{3}{292} \times 100$$

$$\mathcal{A}_{ds}(q_2) = 2; \mathcal{I}_{ds}(q_2) = \frac{2}{292} \times 100$$

$$\mathcal{A}(q_2) = 4 = 2 + 3 - 2 + 1 = \mathcal{A}_s(q_2) + \mathcal{A}_d(q_2) - \mathcal{A}_{ds}(q_2) + 1$$

$$\mathcal{I}(q_2) = \frac{4}{292} = \frac{2+3-2+1}{292} = \mathcal{I}_s(q_2) + \mathcal{I}_d(q_2) - \mathcal{I}_{ds}(q_2) + \frac{1}{\text{Länge von } q_2} \times 100$$

Die Sequenz q_2 ist demnach also nicht nur kürzer, sondern inhärent auch sehr viel weniger komplex als q_1 . Das gilt sowohl in der räumlichen als auch der dynamischen Variation.

7.2.2 Komplexität von Shape-Sequenzen

Die Beschreibungselemente der SHAPE-Repräsentation sind zur Gänze im SHAPE-Vokabular enthalten. Dies ist allerdings nicht so homogen wie die QMVs, da es sehr einfache SHAPES gibt wie z. B. *straight-line*, und komplexe SHAPES wie *loop-left*. Deswegen müssen die SHAPES s_i mit Gewichten $\omega(s_i)$ versehen werden, die ihre intrinsische Komplexität wiedergeben. Dabei muß man allerdings darauf achten, daß ein komplexer SHAPE eine geringere Komplexität erhält als die Summe der Komplexitäten der basalen SHAPES, aus denen er minimal aufgebaut ist: Ansonsten würde die Klassifikation basaler

SHAPES in komplexe in dem definierten Komplexitätsmaß ja keine Komplexitätsreduktion bewirken. Andererseits ist ein *loop-left* durchaus komplexer als eine *straight-line*, so daß man eine Zahl wählen sollte, die zwischen der Komplexität eines basalen SHAPES und der Summe der Komplexitäten der SHAPES liegt, aus denen ein komplexer SHAPE mindestens aufgebaut ist. Ein Beispiel für so eine Zahl ist die Iterationsstufe, auf der ein bestimmter SHAPE bei der SHAPE-Konstruktion frühestens auftauchen kann: Die basalen SHAPES kommen schon nach der Segmentierung vor und erhalten deswegen die Komplexität $0,5^1$; die *u-turns* erhalten die Komplexität 2; die *loops* die Komplexität 3, usw.

In Analogie zu den Komplexitätsmaßen auf QMV-Sequenzen kann man folgende Komplexitätsmaße auf einer SHAPE-Sequenz $s = s_1 s_2 \dots s_n$ definieren:

$$\mathcal{A}(s) = \sum_{i=1}^n \omega(s_i)$$

$$\mathcal{I}(s) = \frac{\mathcal{A}(s)}{n}$$

Da wir einen Zusammenhang zwischen SHAPE-Repräsentation und menschlicher Kognition vermuten, nehmen wir auch an, daß die Beschreibungskomplexität von SHAPE-Sequenzen und die kognitive Komplexität der von einer SHAPE-Sequenz repräsentierten Bewegungsverläufe korrelieren. Dies muß in psychophysischen Experimenten in Zusammenarbeit mit unseren Projektpartnern geklärt werden.

Struktur durch eine einprägsame Form scheint die kognitive Komplexität von Bewegungsverläufen zu verringern, wie in Abbildung 2.3 zu erkennen und in Abschnitt 2.3 diskutiert. Dieser Zusammenhang war ja eines der Hauptmotive für die Entwicklung der SHAPE-Repräsentation.

Aber die Form ist natürlich nicht der einzige Faktor, der Komplexität reduziert: Es können sich auch bestimmte Muster wiederholen, die zwar keine eigene Form darstellen, aber doch redundant sind. Wenn in einer SHAPE-Sequenz $s = s_1 s_2 \dots s_n$ eine Subsequenz $s_k \dots s_l$ maximaler Länge mehrfach vorkommt, so heiße $s_k \dots s_l$ ein *Muster*.

Das Komplexitätsmaß $\mathcal{K}_1(s)$, $s = s_1 s_2 \dots s_n$ versucht, dem komplexitätsreduzierenden Einfluß von solchen Mustern m Rechnung zu tragen. Der Index *unique* bedeutet, daß mehrfach vorkommende Muster oder SHAPES nur einmal gezählt werden; $|m|$ ist die Anzahl von SHAPES, aus denen ein Muster m besteht:

¹1 geht nicht, weil sonst ein *u-turn* genauso komplex ist wie zwei *turns*.

$\mathcal{K}_1(s) =$

$$\omega_1 * \sum_i \omega(s_{i_{\text{unique}}}) + \quad (7.1)$$

$$\omega_2 * \sum_i s_{i_{\text{unique}}} + \quad (7.2)$$

$$\omega_3 * \sum_i m_{i_{\text{unique}}} + \quad (7.3)$$

$$\omega_4 * \sum_i |m_{i_{\text{unique}}}| \quad (7.4)$$

wobei die ω_i Gewichte aus $[0, 1]$ sind, mit deren Hilfe variiert werden kann, welche Bedeutung jedem einzelnen Aspekt bei der Komplexitätsbetrachtung zukommen soll.

(7.1) gibt wieder, wie komplex die einzelnen SHAPES sind, die zu einer Sequenz beitragen.

(7.2) gibt wieder, wie komplex die Sequenz in Bezug auf Reichtum an verschiedenen SHAPES ist: Kommen überhaupt nur immer wieder die ein, zwei selben SHAPES vor oder viele verschiedene?

(7.3) spiegelt die interne Regelmäßigkeit der SHAPE-Sequenz wider: Wenn es interne Periodizität gibt, wird der Summand klein, wenn nicht, wird er groß. Ist kein Muster erkennbar, zählt jeder einzelne SHAPE in der Sequenz als Muster und der dritte Summand wird dementsprechend groß. Z.B.:

(I) *loop_left s_curve_right loop_right left_turn straight_line right_turn* $\rightarrow 6 * \omega_3$

(II) *loop_left straight_line loop_right straight_line loop_left straight_line loop_right straight_line* $\rightarrow 1 * \omega_3$.

Ein bis dato noch nicht angegangenes Problem ist das Parsen der SHAPE-Sequenz, um solche Muster automatisch zu erkennen.

(7.4) gibt wieder, wie lang jedes verschiedene Muster ist, da lange Muster auf einen komplexeren Bewegungsverlauf schließen lassen. Im Falle, daß gar kein Muster festgestellt wurde, zählt wieder jeder einzelne SHAPE mit der Länge 1. Im obigen 1. Beispiel würde das also $6 * \omega_4$ ergeben, im 2. Beispiel $4 * \omega_4$.

Beispiel 7.4

Für die Kurven aus Abbildung 7.1 wurden folgendes SHAPE-Vokabular und Gewichte $\omega(s_i)$ verwendet:

- Die Basic SHAPES *straight-line*, *left-turn*, *right-turn*, *backward-turn* erhalten das Gewicht 0,5.
- *u-turn-left*, *u-turn-right*, aufgebaut auf frühestens der 2. Iterationsstufe, erhalten das Gewicht 2.
- *loop-left*, *loop-right*, *s-curve-left*, aufgebaut auf frühestens der 3. Iterationsstufe, erhalten das Gewicht 3.

Dann ergibt sich \mathcal{K}_1 jeweils zu:

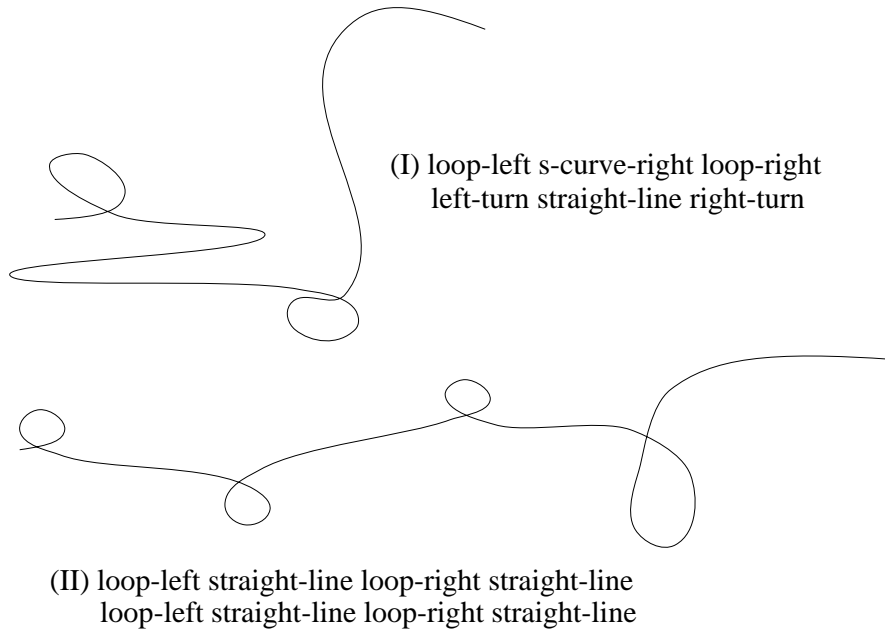


Abbildung 7.1: Bewegungsspuren und ihre zugehörigen SHAPE-Sequenzen

(I):

$$\begin{aligned} \omega_1 * 10,5 &+ \text{(Alle SHAPES verschieden)} \\ \omega_2 * 6 &+ \text{(Alle SHAPES verschieden)} \\ \omega_3 * 6 &+ \text{(jeder SHAPE ein Muster)} \\ \omega_4 * 6 &\text{(jedes Muster 1 lang)} \end{aligned}$$

(II):

$$\begin{aligned} \omega_1 * 6,5 &+ \text{(Nur 3 versch. SHAPES)} \\ \omega_2 * 3 &+ \text{(Nur 3 versch. SHAPES)} \\ \omega_3 * 1 &+ \text{(1 Muster)} \\ \omega_4 * 4 &\text{(jedes Muster 4 lang)} \end{aligned}$$

Wie an den Beispielen deutlich wird, fehlt in diesem Komplexitätsmaß jeder Bezug zu den Eigenschaften der Trajektorie, die über eine bloße SHAPE-Abfolge hinausgeht, also z.B. Regularitäten in der Dynamik oder in den Größenverhältnissen.

Weiterhin gibt \mathcal{K}_1 lediglich die inhärente strukturelle Komplexität wieder, entsprechend dem Maß \mathcal{I} ; die Länge der Sequenz wird nicht berücksichtigt. Die Sequenz *loop-left loop-left loop-left loop-left* erhält den gleichen Wert wie *loop-left loop-left loop-left loop-left*

loop-left loop-left loop-left loop-left loop-left, obwohl ein Beobachter die erste Sequenz vielleicht noch korrekt wiedergeben könnte, die zweite nicht mehr (weil er sich nicht merken kann, wieviele *loops* es waren).

Ein Maß, das dies wiedergibt, ist

$$\mathcal{K}_2 = \mathcal{K}_1 + \omega_5 \cdot \text{Länge der Sequenz}$$

Inwiefern dies der tatsächlichen kognitiven Komplexität entspricht, muß aber experimentell geklärt werden.

8 Anwendungen

8.1 Roboternavigation

Wie wir in den einleitenden Kapiteln bereits ausgeführt haben, besteht unser Interesse an der qualitativen Bewegungsrepräsentation nicht nur darin, menschliche kognitive Leistungen zu modellieren, sondern auch, zu praktischen Anwendungen beizutragen. In Zusammenarbeit mit dem Bremer Projekt „Bildfolgenbasierte semilokale 3D-Landmarken zur Navigation in dynamischen Umgebungen“ im DFG-Schwerpunktprogramm „Raumkognition“ wurden einige der vorgestellten Algorithmen und Repräsentationen in der Roboternavigation erfolgreich eingesetzt.

Das Bremer Projekt verfügt über einen autonomen Rollstuhl namens „Rolland“, ein kommerzielles Produkt der Firma Meyra (siehe Abbildung 8.1). Dieses Fahrzeug besitzt einen Vorderradantrieb und eine Steuerung durch die Hinterräder. Für eine genauere Beschreibung des Rollstuhls, der als sicheres System konzipiert ist, siehe [78].

8.1.1 Navigieren mittels aufgezeichneter Eigenbewegung

Für dieses Projekt war „Rolland“ an Sensorik lediglich mit 27 Polaroid Ultraschallsensoren ausgerüstet. Außerdem kann er seine aktuelle Geschwindigkeit und den Lenkeinschlag messen und so Koppelnavigation durchführen. Implementiert in der Basissoftware des Rollstuhls sind verschiedene Grundverhalten:

- *Wandverfolgung*: Der rechten oder linken Wand folgen, während der Rollstuhl vorwärts oder rückwärts fährt.
- *Gangzentrierung*: In der Mitte des Korridors entweder vorwärts oder rückwärts fahren.
- *Türeinbiegen*: Rechts oder links in eine Tür einbiegen.
- *Stopp*

Die Grundidee des in dieser Zusammenarbeit realisierten Navigationsansatzes ist folgende: Wenn der Rollstuhl mit Hilfe dieser Grundverhalten eine Route verfolgt, so gibt diese



Abbildung 8.1: Der Bremer Autonome Rollstuhl „Rolland“. Abbildung aus [62]

Bewegung die Struktur der Umgebung wieder: Links abbiegen wird ja nicht durch ein eigenes Grundverhalten realisiert, sondern dadurch, daß der Rollstuhl einer Wand folgt, die nach links abbiegt (z. B. weil dort ein Korridor abgeht). Die Odometrie des Rollstuhls zeichnet nun diese Bewegungen auf, und die aufgezeichneten Bewegungen (zusammen mit den zugehörigen Grundverhalten) werden zu einer Repräsentation der Route, die das System verfolgt hatte, verarbeitet. Folgt das System später der gleichen Route, wird eine ähnliche Bewegungsspur aufgezeichnet. Um diese Repräsentationen zur Navigation nutzen zu können, muß man die Routen von unnötigen Einzelheiten befreien, also generalisieren, und das Ergebnis jeweils vergleichen. So kann eine Selbstlokalisierung auf der Route durchgeführt werden. Aufgrund dieser Selbstlokalisierung kann dann entschieden werden, wann zwischen den Grundverhalten umgeschaltet werden muß.

Mittels dieser Technik können dem Rollstuhl nun Routen z. B. in Bürogebäuden durch simples Vormachen beigebracht werden: Jemand, der dazu in der Lage ist, setzt sich in den Rollstuhl und fährt die Routen mit Hilfe der normalen Steuerung durch den Joystick ab. Das System zeichnet die Bewegungsdaten auf, generalisiert sie, und speichert sie in

einer sehr kompakten Repräsentation. Später kann der Rollstuhl die gelernten Routen autonom abfahren. Eine ausführliche Beschreibung dieses Navigationsansatzes ist in [62] nachzulesen; er wurde von Thomas Röfer und Kollegen auf unsere Anregung und unter Benutzung der ε -Generalisierung aus Abschnitt 5.2.1 entwickelt und implementiert.

Als Repräsentation für die generalisierte Bewegungsspur wurden nicht QMV-Sequenzen gewählt, sondern gerade Linien, die durch verschiedene Winkel verbunden sind. Dies ist durchaus plausibel, wenn man bedenkt, daß eine Route in einem geschlossenen Gebäude im wesentlichen aus Korridoren mit Kreuzungen dazwischen besteht. Eine Routenbeschreibung ist also eine Sequenz von geraden Linien mit bestimmter numerischer Länge und Winkeln, z. B. „800 cm, 89°, 345 cm, -83°, 566 cm“. Die Distanzen werden aus den von der Odometrie des Rollstuhls gemessenen Entfernungen im Generalisierungsalgorithmus errechnet.

Als Generalisierungsalgorithmus wurde die ε -Generalisierung verwendet. Der Parameter ε ergibt sich hier ganz natürlich aus der Korridorbreite, die der Rollstuhl mit seinen Ultraschallsensoren messen kann. Welche Bewegungen der Rollstuhl innerhalb des Korridors macht, ob er hier einem Hindernis ausweichen oder rangieren muß, ist für die Repräsentation der Route irrelevant und muß deswegen weggeneralisiert werden. Interessant ist nur, welchen Korridor das System befahren hat.

Wird nun die Route erneut abgefahren, wird die Bewegungsspur wiederum aufgezeichnet und genauso wie im ursprünglichen Durchlauf generalisiert. Dies ist problemlos „on the fly“ möglich, da der Generalisierungsalgorithmus inkrementell ist. Die resultierende Repräsentation wird während der Fahrt mit der gespeicherten verglichen. Da die aktuelle Aufzeichnung nicht vollständig ist, kann sie natürlich nur mit dem Anfang der gespeicherten Route verglichen werden. Das Segment der gespeicherten Repräsentation, das mit dem letzten Segment der gerade gefahrenen Route verglichen wird, ist das *aktuelle Segment*. Zusammen mit der Länge des letzten Segments in der gerade gefahrenen Route definiert dies die *aktuelle Position* des Rollstuhls auf der gespeicherten Route. So findet die Selbstlokalisierung statt, mit deren Hilfe der Rollstuhl zwischen den Grundverhalten umschalten kann.

Dieser Ansatz erlaubt noch eine zweite Anwendung. Während die gespeicherte und die aktuell gefahrene Generalisierung verglichen werden, kann ein Algorithmus entscheiden, ob die beiden dieselbe Route repräsentieren oder nicht. So kann festgestellt werden, ob der Rollstuhl noch auf dem richtigen Weg ist oder sich verfahren hat, weil z. B. ein Korridor blockiert war und so nicht bemerkt wurde. Dann wird entweder ein Segment deutlich länger oder kürzer sein, oder ein Winkel zwischen zwei Segmenten deutlich abweichen. Ein Beispiel ist in Abbildung 8.2 dargestellt. Die erste Route ist die gelernte Route, die zweite eine korrekte Wiederholung der ersten, wohingegen bei der dritten Route ein Fehler bemerkt wurde, weil der Winkel zwischen dem vorletzten und dem letzten Segment zu groß ist.

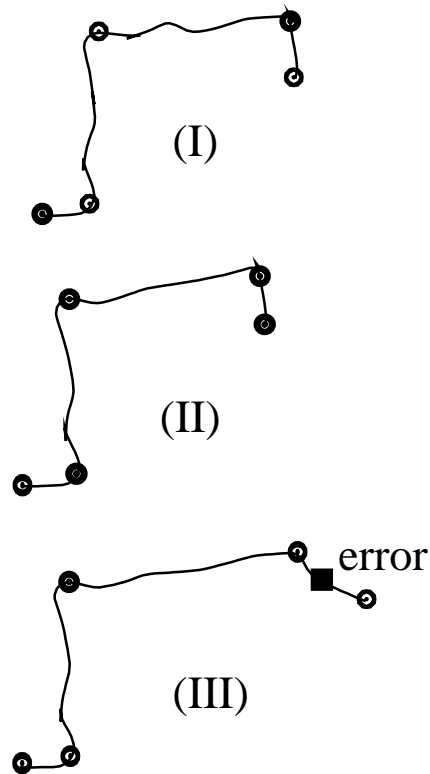


Abbildung 8.2: Drei Bewegungsspuren mit durch Generalisierung gefundenen Ecken. Abbildung aus [62].

Ein Problem des Ansatzes ist, daß implizit angenommen wird, daß ähnliche Bewegungsspuren auch zu ähnlichen Routenrepräsentationen generalisiert werden. In einigen Fällen stimmt dies allerdings nicht, wie auch schon im Abschnitt 7.1.1 erwähnt: Wenn z. B. ein (kurzer) Korridor ungefähr genauso breit wie lang ist, kann es passieren, daß er einmal als eigenständiges Segment generalisiert wird und einmal als Teil eines angrenzenden Segments (wie in Abbildung 8.3). Oder wenn ein Korridor leicht gekrümmt ist, können je nach tatsächlich gefahrener Position im Korridor recht unterschiedliche Generalisierungen entstehen. Dies muß während des Vergleichs in analoger Weise wie in der in Abschnitt 7.1.1 definierten „generalisierten egozentrischen Ähnlichkeit“ berücksichtigt werden. Der Vergleichsalgorithmus muß wegen der teilweise beträchtlichen Meßungenauigkeit in der Odometrie des Rollstuhls noch recht unterschiedliche Winkel und Längen als gleich akzeptieren. Auch das ist analog zur egozentrischen Ähnlichkeit, wie sie in Abschnitt 7.1.1 eingeführt wurde.

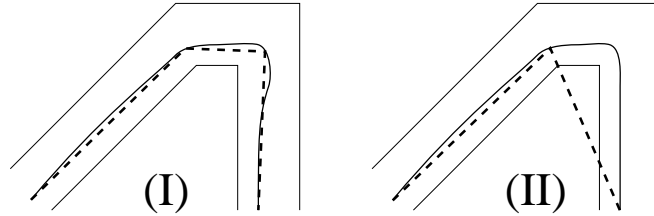


Abbildung 8.3: Unterschiedliche Generalisierungen bei ähnlichen Bewegungsverläufen.
Abbildung aus [62].

8.1.2 Navigieren mit groben, qualitativen Routenbeschreibungen

Die oben beschriebene Navigationsmethode funktioniert natürlich nur für genau die Routen, die einmal mit Hilfe des Joysticks abgefahren und gespeichert wurden. Das erfordert, daß jemand, der in der Lage ist, den Rollstuhl konventionell zu steuern, alle gewünschten Routen einmal abfährt. Besser wäre hier ein Navigationsverfahren, das es dem Behinderten selbst ermöglicht, die gewünschte Route einzugeben. Deswegen haben wir zusammen mit dem Bremer Projekt ein weiteres Navigationsverfahren entwickelt, das auf der Eingabe grober, qualitativer Routenbeschreibungen basiert.

Wie sollten diese Beschreibungen aussehen? Wie in [85] festgestellt, können Routenbeschreibungen, die Menschen üblicherweise geben, in Teilstücke segmentiert werden, die hauptsächlich in folgende Kategorien eingeordnet werden können: Startpunkt, Neuorientierung (Richtung), Weg/Wegfortschritt und Ziel. Außerdem ergab die Untersuchung, daß Menschen oft zusätzliche Informationen geben, wie z. B. Landmarken entlang des Wegs, die keine Abbiegepunkte markieren, Himmelsrichtungen, und die Form des Wegs zwischen den Landmarken. “This information, while not essential, may be important for keeping the traveler confidently on track” [85], p. 169.

Die Routenbeschreibungen, die Menschen üblicherweise geben, sind Sequenzen elementarer Teilstücke, die aus den eben aufgezählten Bestandteilen bestehen.

Diese elementaren Teilstücke werden so lange aneinandergereiht, bis das Ziel des Teilstücks das Ziel der Route ist. Da meist der Startpunkt eines Segments das Ziel des vorigen ist, wird der Startpunkt oft weggelassen.

Um die Kommunikation zwischen dem menschlichen Benutzer und dem Rollstuhl zu erleichtern, sollte der Rollstuhl Routenbeschreibungen verstehen, die aus Elementen wie den oben geschilderten bestehen, und seinen Weg in einem Gebäude aufgrund solcher Beschreibungen finden.

Qualitative Routenbeschreibungen

Wenn der Endpunkt eines Teilstücks der Routenbeschreibung gleichzeitig Startpunkt des nächsten Stücks ist, kann ein solches Teilstück folgendermaßen formalisiert werden:

$$\langle [\{ controlmarks \} router] reorientation \rangle,$$

wobei

- eine *reorientation* eine Richtungsangabe ist, wie Menschen sie in Routenbeschreibungen [86] häufig benutzen, z. B. „TurnLeft“ (biege links ab), „EnterRightDoor“ (biege rechts in eine Tür ein), oder „FollowCorridor“ (folge dem Korridor).
- ein *router* eine Landmarke ist, an der eine Richtungsänderung stattfindet; der letzte Router ist das Ziel der Route.
- *Controlmarks* Landmarken sind, die passiert werden, ohne daß eine Richtungsänderung stattfindet; sie dienen der Kontrolle, ob der Weg noch stimmt.

Ein Tupel kann lediglich aus einer Reorientierung bestehen. Sobald allerdings Kontrollmarken enthalten sind, muß auch ein Router vorhanden sein. Eine Routenbeschreibung ist eine Sequenz aus solchen Tupeln. Startpunkt der Route ist immer der aktuelle Standpunkt, Endpunkt ist der letzte Router.

Erweiternd sollten in solche Tupel als optionales Element noch qualitative Distanzen aufgenommen werden, die die Entfernung der Router voneinander beschreiben. Dies ist bereits vorbereitet, wurde allerdings noch nicht in die Software des Rollstuhls implementiert. Qualitative Distanzen erleichtern dem Roboter die Suche nach den Routern. Wenn bekannt ist, daß längere Zeit kein Router auftaucht, kann die Sensitivität der Landmarkenerkennung (die ja bei einem lediglich mit Ultraschall ausgerüsteten Roboter nicht trivial ist) heruntergeschraubt werden, um weniger „falsch positive“ Landmarkenerkennungen zu erhalten, wohingegen in der Nähe des vermuteten Routers die Sensitivität wieder hochgeschaltet werden kann, um den Router nicht zu übersehen.

Wie die Landmarkenerkennung mit Hilfe von Ultraschall funktioniert, ist in [60] nachzulesen. Wegen der Schwierigkeit der Aufgabe können Türen in dieser Implementierung nicht als Landmarken benutzt werden, da sie zu schwierig zu erkennen sind. Vorwiegend werden Korridorkreuzungen oder -einbiegungen sowie die Form des Wegs, wie sie durch die in Abschnitt 8.1.1 dargestellte Generalisierung erkannt wird, als Landmarke benutzt.

Die qualitative Routenbeschreibung *A*

$$\begin{aligned} &\langle \text{RightHandBend TurnRight} \rangle \\ &\langle \text{CorridorRight CorridorLeft TurnLeft} \rangle \\ &\langle \text{CorridorRight DeadEnd Stop} \rangle \end{aligned}$$

korrespondiert zu der in Abbildung 8.4 als gestrichelte Linie dargestellten Route. Dabei wird angenommen, daß der Rollstuhl nicht nur am Ausgangspunkt steht, sondern auch bereits in der richtigen Orientierung, und deswegen geradeaus losfahren kann, bis er an den ersten Router gelangt. Falls der Rollstuhl nicht in der richtigen Orientierung steht, muß ein weiteres Tupel vorne an die Routenbeschreibung angefügt werden, das nur aus einer *reorientation* besteht wie z. B. „TurnRound“, die dazu geeignet ist, den Rollstuhl in die richtige Orientierung zu bringen.

Da der Rollstuhl *reorientations* wie „TurnRight“ nicht direkt ausführen kann, müssen diese zunächst in Grundverhalten übersetzt werden. Diese werden gestartet, wenn der Router entdeckt wurde, und beendet, nachdem sie erfolgreich passiert wurden.

Als erster Schritt wird also die Routenbeschreibung in eine Repräsentation übersetzt, die auf den Grundverhalten basiert:

```
< FollowRightWall RightHandBend>
< FollowLeftWall CorridorRight CorridorLeft >
< FollowLeftWall CorridorRight DeadEnd >
< Stop >
```

Ein Problem, das mit dieser Repräsentation immer noch bleibt, besteht darin, den Rollstuhl davon abzuhalten, in einen Korridor einzubiegen, der lediglich als Kontrollmarke oder gar nicht in der Routenbeschreibung vorkommt. Ist als Grundverhalten „FollowLeftWall“ eingestellt, und vor dem Router geht ein weiterer Korridor links ab, so würde der Rollstuhl automatisch in diesen einbiegen. Dies muß bei Verfolgung der Route beachtet und verhindert werden: Wenn solch eine interferierende Landmarke, also eine Landmarke, die zu einer falschen Einbiegung führen würde, als Kontrollmarke vorkommt, muß bis zum Passieren dieser Landmarke ein Verhalten benutzt werden, das dazu führt, daß der Rollstuhl an dieser Landmarke geradeaus vorüberfährt. Kommt die Landmarke in der Routenbeschreibung allerdings gar nicht vor, kann das falsche Einbiegen mit der momentanen Implementierung gar nicht verhindert werden. Hier müßten als zusätzliche Kontrolle noch qualitative Distanzen in die Routenbeschreibung mit aufgenommen werden.

Generierung von Fahrkommandos

Der Navigationsalgorithmus funktioniert wie folgt. Das erste Tupel T der Routenbeschreibung wird ausgewählt. Je nach Inhalt von T , kann man drei Fälle unterscheiden:

T enthält lediglich eine Reorientierung. Dann muß das Grundverhalten ein intrinsisches Ende haben und der Rollstuhl in der Lage sein zu erkennen, wann der Endzustand erreicht ist. Dies ist z. B. bei „Stop“ oder „TurnRound“ der Fall. Der Algorithmus wartet

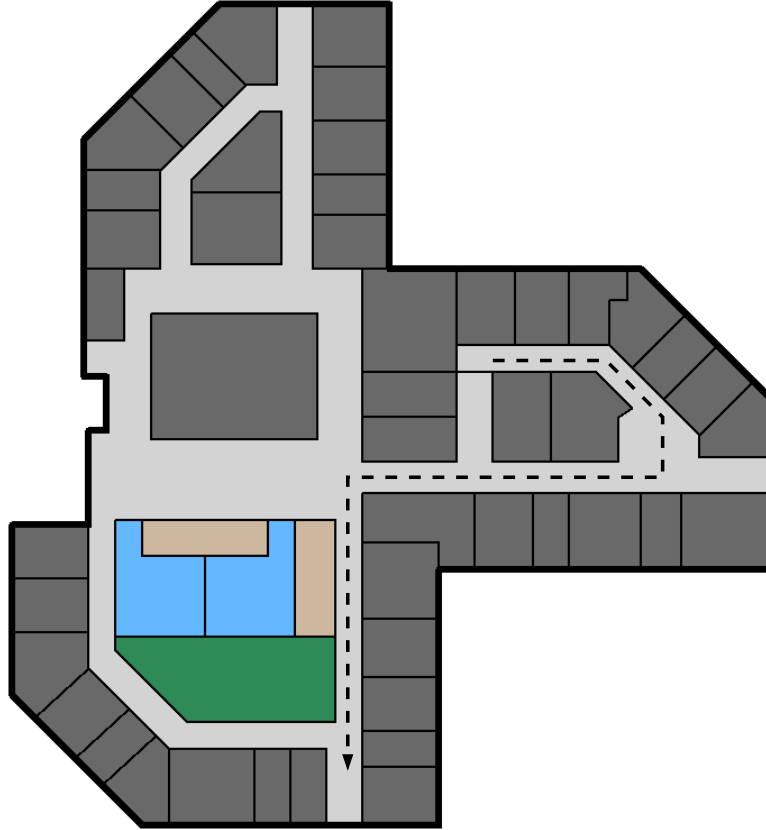


Abbildung 8.4: Plan des 2. Stocks des MZH-Gebäudes in der Bremer Universität und der Route A (gestrichelte Linie, ca. 65 m). Abbildung aus [60].

dann, bis das Verhaltensmodul des Rollstuhls signalisiert, daß das Verhalten beendet ist. Anschließend entfernt der Algorithmus T und arbeitet das nächste Tupel ab.

T **enthält Kontrollmarken.** An diesen Kontrollmarken muß der Rollstuhl geradeaus vorbeifahren, um den Router zu erreichen, an dem eine Reorientierung stattfinden soll. Deshalb wird mit jedem Kontrollmarken-Typ das Default-Verhalten assoziiert, das notwendig ist, damit der Rollstuhl dort nicht abbiegt. Nachdem die Kontrollmarke erfolgreich passiert wurde, wird sie aus T entfernt und der Algorithmus arbeitet mit dem restlichen Tupel weiter.

T enthält keine Kontrollmarken, entweder, weil in der Routenbeschreibung keine angegeben wurden oder, weil sie alle erfolgreich passiert und aus T entfernt wurden. Dann muß der Rollstuhl nach dem Router suchen. Im Unterschied zur Kontrollmarke soll der Rollstuhl hier gerade nicht davon abgehalten werden, abzubiegen. Wenn der Router z. B. eine Korridor Kreuzung ist, und das Grundverhalten „FollowLeftWall“, so biegt der Rollstuhl dort korrekt links ein. Wenn der Router erfolgreich passiert wurde, wird T aus der Routenbeschreibung gelöscht und der Algorithmus arbeitet mit dem nächsten Tupel weiter.

Wenn die Routenbeschreibung leer ist, ist das Ziel erreicht und der Algorithmus stoppt den Rollstuhl.

Experimentelle Ergebnisse

Der Algorithmus wurde zunächst in einer Simulation in Java implementiert und dann von Thomas Röfer und Kollegen leicht verändert auf den Bremer Rollstuhl portiert. Experimente auf dieser Plattform ergaben, daß der Algorithmus praxistauglich ist. In Abbildung 8.5 ist ein erfolgreicher Versuch dargestellt, die Route A abzufahren.

Ein Problem der momentanen Implementierung sind die eher unzuverlässigen und teilweise schwer zu interpretierenden Daten der Ultraschallsensoren, die dazu führen, daß Landmarken häufig nicht erkannt werden oder dort falsch erkannt werden, wo gar keine sind. Dieser Ansatz kann nur dann gut funktionieren, wenn Landmarken zuverlässig erkannt werden.

8.2 Von Bewegung zu Routenbeschreibungen

Im letzten Abschnitt wurden Möglichkeiten besprochen, qualitative Beschreibungen zur Roboternavigation zu benutzen. Dazu wurden Teile der entwickelten Repräsentationen und Algorithmen benutzt. Für die umgekehrte Möglichkeit, nämlich aus beobachteten oder aufgezeichneten Bewegungsdaten eine abstrakte, qualitative Routenbeschreibung automatisch zu erzeugen, braucht man die meisten der in der Arbeit beschriebenen Techniken: Generalisierung, Segmentierung und Klassifikation der Bewegungsdaten.

Um Routenbeschreibungen zu generieren, muß allerdings eine wichtige Komponente in die QMVs mit aufgenommen werden: Landmarken. Wie schon in Abschnitt 8.1.2 dargestellt, kann zwar die Form eines Wegabschnitts auch als Landmarke verwendet werden. Dies ist aber in Routenbeschreibungen von untergeordneter Bedeutung. Landmarken in Routenbeschreibungen sind üblicherweise bekannte und/oder weithin sichtbare Gebäude, Plätze, Straßen, etc.

Es ist klar, daß derartige Landmarken nicht punktförmig, sondern ausgedehnt sind. Wenn

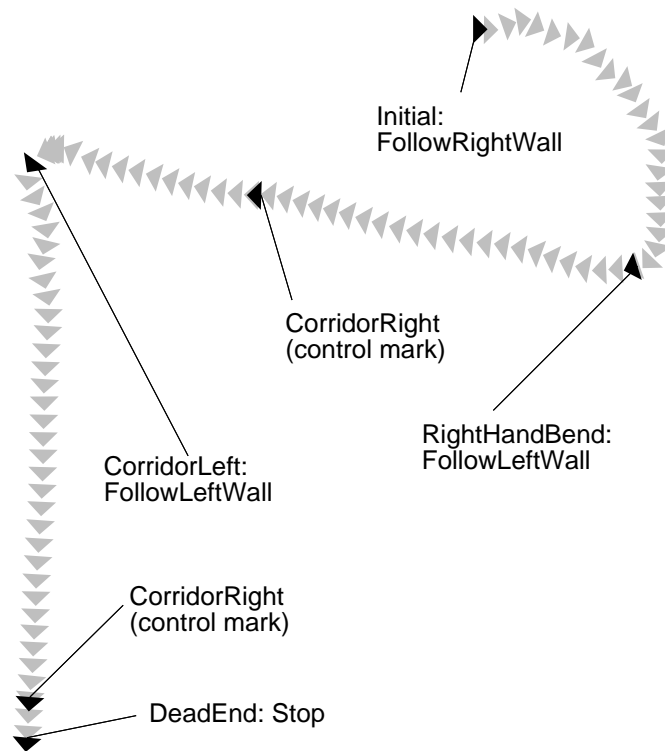


Abbildung 8.5: Pfad des Rollstuhls bei einem erfolgreichen Versuch, die Route A abzufahren. Die Dreiecke geben Position und Orientierung des Rollstuhls wieder in Intervallen von ungefähr .25 s (aus [60]).

Straßen oder Plätze als Landmarken fungieren, dann sind die Bewegungen, die innerhalb dieser Regionen stattfinden, für die Routenbeschreibungen meist irrelevant. Falls diese Bewegungen doch von Bedeutung sind, ist es oft so, daß diese Regionen lokale Referenzrahmen definieren und es deswegen nützlich ist, wenn man zwischen diesem lokalen Referenzsystem und dem in Routenbeschreibungen weitgehend üblichen egozentrischen Referenzsystem wechseln kann.

Auch eine Anpassung der Granularität für Teile der Routenbeschreibung kann nützlich sein. So ist es z. B. sinnvoll, in Bereichen, in denen man sich leicht verlaufen kann (weil z. B. unübersichtliche Kreuzungen mit vielen Abbiegemöglichkeiten vorhanden sind), eine feinere Granularität zu wählen als für den Rest der Routenbeschreibung.

Für eine automatische Generierung von Routenbeschreibungen aus beobachteten oder aufgezeichneten (Eigen-) Bewegungen ist also folgendes nützlich:

- Einbinden von Landmarken: Der einfachste und exakteste Weg, Landmarken in QMV-Sequenzen einzubinden, ist, sie als Komponenten in die QMVs selbst mitaufzunehmen. In einem QMV ist dann nicht nur die Information über Länge, Richtung und Geschwindigkeit des Bewegungsverlaufs enthalten, sondern auch an welchen Landmarken das bewegte Objekt vorbeigekommen ist. Je nach räumlicher Ausdehnung der Landmarke und Geschwindigkeit des Bewegungsverlaufs, kommt ein und dieselbe Landmarke dann in mehreren aufeinanderfolgenden QMVs vor. So erhält die Landmarke im Bewegungsverlauf auch eine *zeitliche* Ausdehnung. Landmarken können ihren eigenen lokalen Referenzrahmen definieren, wie bereits in Kapitel 6.3 besprochen. Trotz der Einbindung von Landmarken findet die Segmentierung und Klassifizierung der QMV-Sequenz in eine Sequenz von MOTION SHAPES unabhängig von diesen statt, weil die Form eines Bewegungsverlaufs schließlich selbst eine Landmarke sein kann. Dies haben wir im Roboternavigationsbeispiel in Abschnitt 8.1.2 bereits gesehen. Wenn an einer Landmarke kein Richtungswechsel stattfindet, hat diese nur Kontrollfunktion. Deshalb ergibt es keinen Sinn, an dieser Landmarke zu segmentieren.
- Wechsel des Referenzsystems: Da Landmarken eine zeitliche, nicht nur räumliche Ausdehnung erhalten, ist es möglich, eine ganze Bewegungssequenz mit einer Landmarke zu assoziieren. Dies kann der Fall sein, wenn die Landmarke eine größere Region ist, wie z. B. der Marienplatz in München. Diese Landmarken können ein lokales Referenzsystem definieren, und es ist möglich, die Bewegungssequenz auf dieses lokale Referenzsystem umzurechnen. So erhält man Routenbeschreibungen mit gemischten Referenzsystemen, wobei eine zusätzliche Angabe des allozentrischen Referenzsystems, falls von Anfang an bekannt, immer möglich ist. Falls das allozentrische Referenzsystem aus anfänglicher allozentrischer Richtung und egozentrischer Messung erschlossen werden muß, sind natürlich die in Kapitel 6 besprochenen Einschränkungen zu beachten.
- Wechsel der Granularität: Routen können ja auf verschiedenen Granularitäten beschrieben werden; und auch in unserem Ansatz ist es möglich, die Granularitäten zu wechseln. So kann z. B. eine Routenbeschreibung in der grobgranularen Form mit MOTION SHAPES und Landmarken gegeben werden, und näher am Ziel oder an unübersichtlichen Stellen kann dann auf die feinere Granularität der QMVs gewechselt werden.

Beispiel:

In unserem Beispiel gehen wir von einem Bewegungsverlauf aus, der gemessen, generalisiert und in QMVs umgewandelt wurde.

```
<far east fast [lm1]> <medium-dist northeast medium-vel []>
```

```
<medium-dist east slow []> <far south fast [lm2]> <close east slow [lm3]> <close southeast very-slow [lm3]> <close east very-slow [lm3]> <close west very-slow [lm3]>
```

Die Landmarken werden hier einfach durch Nummern identifiziert. Durch Generalisierung kann es passieren, daß einem QMV mehrere Landmarken zugeordnet werden¹, weswegen hier jedem QMV eine Liste von Landmarken zugeordnet ist. Die Ordnung der Liste entspricht auch der zeitlichen Abfolge der Landmarken. In diesem Beispiel ist mit der Landmarke 3 eine längere Subsequenz assoziiert, was hier den Einsatz eines lokalen Referenzsystem möglich macht.

Aus der allozentrisch gemessenen Sequenz können wir natürlich automatisch die egozentrischen Richtungen berechnen:

```
<far east forward fast [lm1]> <medium-dist northeast forward-left medium-vel []> <medium-dist east forward-right slow []> <far south right fast [lm2]> <close east left slow [lm3]> <close southeast forward-right very-slow [lm3]> <close east forward-left very-slow [lm3]> <close west backward very-slow [lm3]>
```

Der Segmentierungs- und Klassifizierungsalgorithmus erzeugt nun eine Sequenz von MOTION SHAPES und die jeweils korrespondierende Liste von Landmarken:

straight-line	Landmarks passed: [lm1]
half-left-turn	Landmarks passed: []
three-quarter-right-turn	Landmarks passed: []
straight-line	Landmarks passed: [lm2]
left-turn	Landmarks passed: [lm3]
half-right-turn	Landmarks passed: [lm3]
half-left-turn	Landmarks passed: [lm3]
backward-turn	Landmarks passed: [lm3]
straight-line	Landmarks passed: [lm3]

Auf einer größeren Granularitätsstufe können wir den Bewegungsverlauf, der mit Landmarke 3 assoziiert ist, ignorieren. Möglicherweise ist dies für die Routenbeschreibung nicht relevant, weil sowieso alles klar ist, sobald man diese Region erreicht hat, z. B. weil das Ziel von dort gut zu sehen ist, oder Landmarke 3 selbst das Ziel darstellt. Dann reicht es aus, folgende Beschreibung zu geben:

¹Das kann natürlich so auch passieren, wenn man annimmt, daß man mehrere Landmarken gleichzeitig entdecken kann, was wir hier allerdings der Einfachheit halber ausschließen. In Kapitel 9 werden wir ansprechen, wie sich Landmarken strukturierter in die QMV-Sequenzen einbinden lassen.

straight-line	Landmarks passed: [lm1]
half-left-turn	Landmarks passed: []
three-quarter-right-turn	Landmarks passed: []
straight-line	Landmarks passed: [lm2]
left-turn	Enter Landmark 3

Weil wir uns die jeweilige QMV-Sequenz, die zu jedem MOTION SHAPE gehört, gemerkt haben, können wir die Beschreibung aber auch expandieren und zu einer feineren Granularität wechseln:

```

straight-line – Landmarks passed: [lm1]
half-left-turn – Landmarks passed: [ ]
three-quarter-right-turn – Landmarks passed: [ ]
straight-line – Landmarks passed: [lm2]
left-turn – Enter Landmark 3:
<close southeast forward-right very-slow><close east forward-
left very-slow><close west backward very-slow>

```

Wenn das lokale Referenzsystem der Landmarke 3 bekannt ist, können die allozentrischen Richtungsangaben der Subsequenz, die mit dieser Landmarke assoziiert ist, in den intrinsischen Referenzrahmen, der von ihr erzeugt wird, umgerechnet werden. In unserem Beispiel ist die allozentrische Orientierung der intrinsischen Referenzachse der Landmarke 3 *east*, und die lokal-intrinsischen Richtungen werden mit *forward-i* (= *east*), *forward-right-i*, *right-i*, *backward-right-i*, *backward-i*, *backward-left-i*, *left-i*, *forward-left-i* bezeichnet:

```

<close forward-right-i forward-right very-slow lm3> <close
forward-i forward-left very-slow lm3> <close backward-i back-
ward very-slow lm3>.

```


9 Schlußbemerkungen

9.1 Zusammenfassung

In der vorliegenden Arbeit wurde eine qualitative Repräsentation von Bewegungsverläufen vorgestellt, die aus zwei Schichten besteht. Die untere Schicht, die QMV-Repräsentation, basiert auf einer Diskretisierung des Raumes in Intervalle, die mit natürlichsprachlichen Bezeichnern versehen werden. Es wurden unterschiedliche Möglichkeiten der Diskretisierung des Raumes vorgestellt und teilweise aus der Kognition motiviert. Die QMV-Repräsentation ist modular in der Hinsicht, daß sie mit allen Kombinationen von Diskretisierungen funktioniert, denn diese führen lediglich zu unterschiedlichen Parkettierungen des Raums. Die obere Schicht, die SHAPE-Repräsentation, benutzt die Form von Teilen des Bewegungsverlaufs als primitives Element der Repräsentation. Dies spiegelt die große Bedeutung des Formaspekts für die Bewegungswahrnehmung wider und war zu Anfang als Anforderung an eine kognitiv plausible Repräsentation von Bewegungsverläufen identifiziert worden. Auch die SHAPE-Repräsentation ist modular, sowohl was das SHAPE-Vokabular betrifft, als auch in den verwendeten Klassifizierungsalgorithmen und -regelbasen. Das ermöglicht es, weitere Erkenntnisse zum Formaspekt in der Wahrnehmung eines einzelnen bewegten Punktes auch nachträglich noch einfließen zu lassen.

Die beiden Schichten repräsentieren Bewegungsverläufe auf unterschiedlichen Granularitäten. Die Darstellung kann problemlos gemischt werden, wie in Abschnitt 8.2 deutlich wurde. Die QMV-Repräsentation funktioniert mit egozentrischen, allozentrischen und lokal-intrinsischen Referenzsystemen, die ebenfalls gewechselt bzw. gemischt werden können. Auch dies war eine wichtige Anforderung an eine qualitative Repräsentation von Bewegungsverläufen, die, z. B. in der Benutzerinteraktion, intuitiv sein soll.

Für die automatische Konstruktion der QMV-Repräsentation aus numerischen Daten wurden verschiedene Möglichkeiten vorgestellt: Direkte Abbildung in die Quantity Spaces, Generalisierung auf numerischen Daten mit anschließender Abbildung in die Quantity Spaces, und Generalisierung auf fuzzy-numerischen Daten mit anschließender Abbildung in die qualitativen Intervalle. Außerdem wurden mehrere Möglichkeiten des Rechnens mit QMVs vorgestellt.

Generalisierungsalgorithmen abstrahieren von unnötigen Details des Bewegungsver-

laufs, um die Form besser extrahieren zu können. Zwei solcher Algorithmen wurden im Rahmen dieser Arbeit entwickelt. Beide funktionieren mit QMVs, exakten oder fuzzy-numerischen Daten, so daß numerische Daten und qualitative Benutzereingaben gleichermaßen mit ihnen verarbeitet werden können. Zur Konstruktion der SHAPE-Repräsentation werden die QMV-Sequenzen segmentiert und in basale SHAPES umgewandelt, die dann mittels verschiedener Klassifizierungsmethoden zu komplexeren SHAPES zusammengefaßt werden. Die resultierende SHAPE-Sequenz stellt eine äußerst kompakte Repräsentation von Bewegungsverläufen dar, mit Schwerpunkt auf dem Formaspekt, die geeignet ist, bei Menschen eine gute Vorstellung vom beschriebenen Bewegungsverlauf zu evozieren.

Überlegungen zu Vergleichbarkeit und Komplexität der so repräsentierten Bewegungsverläufe runden die Betrachtungen ab. An einigen Stellen wurde versucht, menschliche kognitive Leistungen mit Hilfe der Repräsentation zu modellieren. So gingen z. B. menschliche Richtungsdiskriminationsleistungen in die Modellierung der entsprechenden Quantity Spaces ein, und menschliche Reproduktions- und Gedächtnisleistungen in die Klassifizierung der SHAPE-Sequenzen mittels Memory Buffer. Eine experimentelle Validierung dieser Modellierung steht noch aus und wird in der kommenden Projektphase durchgeführt werden.

Die Praxistauglichkeit der entwickelten Repräsentationen und Algorithmen wurde in einer Roboteranwendung unter Beweis gestellt. In der Navigation aufgrund von aufgezeichneter Eigenbewegung kam einer der entwickelten Generalisierungsalgorithmen zum Einsatz und es mußten Bewegungssequenzen verglichen werden. In der Navigation mit groben, qualitativen Routenbeschreibungen wurde die Form des Bewegungsverlaufs aus der generalisierten Spur extrahiert und als zusätzliche Landmarke benutzt. Diese Formlandmarken können nur aufgrund der Odometrie und ganz ohne Sensoren erkannt werden. Ein Verfahren, das auf die Detektion von Landmarken angewiesen ist, wird durch die Verwendung derartiger Landmarken robuster. Die Routenbeschreibung selbst enthält Ausdrücke wie „TurnRight“, „TurnLeft“ und „TurnRound“, die äquivalent sind zu den basalen MOTION SHAPES bei einer Diskretisierung in vier Richtungen, allerdings wegen des Anweisungscharakters der Routenbeschreibung in Befehlsform gebracht wurden. Die letzte vorgestellte Anwendung ist der duale Fall zur Navigation mittels Routenbeschreibungen, nämlich die Erzeugung von Routenbeschreibungen aus aufgezeichneter Bewegung. Hier konnten die meisten der entwickelten Repräsentationen und Algorithmen zum Einsatz gebracht werden

9.2 **Ausblick**

Trotz der lückenlosen Beschreibung des Wegs von gemessenem Bewegungsverlauf zu einer qualitativen Repräsentation, die auf der Form dieses Bewegungsverlaufs basiert,

ist diese Arbeit natürlich keineswegs erschöpfend. Bei der Entwicklung der vorgestellten Repräsentationen und Algorithmen begegneten uns immer wieder interessante Probleme und Ideen, zu deren Lösung bzw. Realisierung wir noch keine Gelegenheit hatten.

Ein offensichtliches Problem ist die Kinetik des Bewegungsverlaufs, die vielerorts noch nicht stark genug berücksichtigt wird. So werden Stillstände momentan von den Generalisierungsalgorithmen weggeneralisiert und stehen dem Segmentierungsalgorithmus nicht mehr als Merkmale, an denen eine Segmentierung sinnvoll ist, zur Verfügung. Hier ist eine stärkere Einbeziehung der Besonderheiten der Kinetik sicher noch notwendig!

Auch bei der Integration der Landmarken gibt es noch Ergänzungsmöglichkeiten. Im vorgestellten Ansatz merken wir uns lediglich, welche Landmarken passiert wurden, aber nicht, wo sie ungefähr im Verhältnis zum QMV lokalisiert sind. Darum können wir momentan auch nur für ausgedehnte Landmarken wie die Landmarke 3 aus Abschnitt 8.2, die *betreten* werden, das intrinsische Referenzsystem benutzen. Bei den anderen Landmarken wissen wir nicht, ob sie rechts oder links von Bewegungsverlauf passiert wurden. Hier könnte eine Repräsentation wie Freksas Doppelkreuzkalkül die QMV-Repräsentation erweitern: Jedes Bewegungssegment enthält dann nicht nur die Landmarken, die passiert wurden, sondern auch entfernte, aber sichtbare Landmarken (modelliert evtl. durch die „Einflußgebiete“, siehe [47]), und deren Lage relativ zum QMV als Doppelkreuzrelation. Der Doppelkreuzkalkül läßt dann auch Inferenzen zu, die bei einer Generalisierung sehr nützlich sein können: Das große Problem bei der Generalisierung eines Bewegungsverlaufs mit Landmarken ist, daß der Winkel, in dem man Landmarken passiert, sich dabei signifikant ändern kann. Wenn man die Lage der Landmarke zum generalisierten Bewegungsverlauf nicht mitinferiert, bekommt man unter Umständen völlig falsche Informationen darüber. Bei der Generierung natürlichsprachlicher Routenbeschreibungen haben die Doppelkreuzrelationen den Vorteil, daß sich Bezeichnungen wie „Links vorne“ oder „Rechts“ einfach aus ihnen erzeugen lassen; die Relationen wiederum lassen sich aus dem Meßdaten einfach berechnen.

Bislang wurden in unserer Repräsentation nur Punkte betrachtet, die keine intrinsische Orientierung haben. Die Orientierung des Objekts relativ zur Bewegungsrichtung kann aber leicht in die Repräsentation mit aufgenommen werden. Dies kann auf QMV-Ebene einfach durch eine zusätzliche Vektorkomponente geschehen, wobei sich interessante Aspekte bezüglich des Referenzsystems ergeben. So kann die Orientierung z.B. allozentrisch „absolut“ oder relativ zur Bewegungsrichtung codiert werden. Hier stellt sich die Frage, welches Referenzsystem für welche Zwecke besonders geeignet ist. Diese Erweiterung der QMV-Repräsentation ermöglicht es dann auch, eine Änderung der Orientierung im Raum ohne Bewegung durch den Raum zu beschreiben, wodurch die Repräsentation an Ausdrucksstärke gewinnt.

Eine interessante Frage im Zusammenhang mit orientierten Objekten ist, was diese Erweiterung für Auswirkungen auf das SHAPE-Vokabular hat: Zwei Bewegungen, die dieselbe Trajektorie beschreiben, können verbal unter Umständen unterschiedlich beschrieben

werden, wenn die Orientierung des sich bewegenden Objekts verschieden ist (z.B. „abbiegen“ vs. „seitwärts fahren“, cf. Abb. 9.1 (II) und (I)).

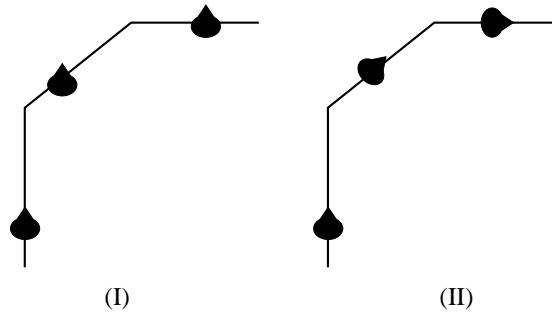


Abbildung 9.1: Trajektorien mit orientierten Objekten

In der vorliegenden Arbeit wird nur die Bewegung eines einzelnen Objekts relativ zu einem ruhenden Bezugssystem betrachtet. Die Bewegung eines Objekts relativ zu einem zweiten, sich bewegenden Objekt kann noch nicht repräsentiert werden. Dies ist aber nicht nur vom theoretischen Standpunkt her interessant. Gerade auch in der natürlichsprachlichen Roboterinstruktion können Kommandos wie “Verfolge die Person” oder “Überhole die Person” nützlich sein. Während auf der höheren Ebene der Zwei-Schicht-Repräsentation diese Bewegungen einfach durch die entsprechenden Verben repräsentiert werden können, stellt sich auf dem unteren Level die Frage nach der Repräsentation einer Bewegung in einem sich selbst bewegenden Bezugssystem. Eine weitere Frage ist die Operationalisierung solcher Instruktionen in konkrete Fahrkommandos, z.B. für den Bremer Rollstuhl. Ebenfalls von Interesse ist die Kombination von Bewegungsteilen mit ruhendem Referenzsystem wie bisher und dem neuen, bewegten Referenzsystem, z.B. wenn in der Roboterinstruktion von “Verfolge die Person” zu “Biege hier rechts ab” übergegangen wird.

Ein weiteres Thema, das bisher nur am Rande gestreift wurde, ist die Prädiktion weiterer Bewegungen aus einem gegebenen Bewegungsverlauf. Wenn dafür gar kein Vorwissen außer dem bisherigen Bewegungsverlauf vorhanden ist, kann natürlich nur die Analyse desselben zur Prädiktion benutzt werden. Hypothese ist hierbei, daß man aus der Vergangenheit des Bewegungsverlaufs auf seine Zukunft schließen kann. Dies ist z.B. der Fall, wenn es sich um eine Bewegung mit sich immer wieder wiederholenden Mustern handelt. In [87] wird gezeigt, daß auch Menschen solche Regularitäten schnell entdecken und aufgrund dieser eine Antizipation des Ereignisverlaufs vornehmen. Es scheint also legitim, bei der Prädiktion von Bewegungsverläufen von ähnlichen Voraussetzungen auszugehen. Dazu müssen natürlich derartige Regularitäten erkannt werden können. Die SHAPE-Repräsentation kann dabei schon als eine Art Vorverarbeitung des Bewegungsverlaufs angesehen werden, die durch ihr Zusammenfassen einzelner Ereignisse (wie *left*,

left, left, left) zu einer höheren Struktur (wie *loop-left*) die Komplexität reduziert und somit Auffinden von Regularitäten, die wiederum zur Prädiktion dienen können, erleichtert. In einem Systementwicklungsprojekt [49] wurden hierzu schon erste Vorarbeiten geleistet.

Literaturverzeichnis

- [1] ALLEN, JAMES F.: *Maintaining Knowledge about Temporal Intervals*. Communications of the ACM, 26(11):832–843, November 1983. [2.2.2](#), [2.4.2](#)
- [2] ALLEN, JAMES F. und J. A. KOOMEN: *Planning Using a Temporal World Model*. In: *Proceeding of the Eighth International Joint Conference in Artificial Intelligence*, Seiten 741–747, Karlsruhe, 1983. [11](#)
- [3] ANDRÉ, ELISABETH, GUIDO BOSCH, GERD HERZOG und THOMAS RIST: *Characterizing Trajectories of Moving Objects Using Natural Language Path Descriptions*. In: *Proceedings of the 7th ECAI*, Band 2, Seiten 1–8, Brighton, UK, 1986. [2.2](#)
- [4] ASHER, N. und L. VIEU: *Toward a Geometry of Common Sense: A Semantics and Complete Axiomatization of Mereotoplogy*. In: *Proc. IJCAI-95*, Seiten 847–852. Morgan Kaufman, 1995. [2.2.3](#), [2.4.2](#)
- [5] DEL BIMBO, A. und E. VICARIO: *A Logical Framework for Spatio Temporal Indexing of Image Sequences*. In: CHANG, S. K., E. JUNGERT und G. TORTORA (Herausgeber): *Intelligent Image Database Systems*, Seiten 165–196. World Scientific Publishing, Singapur, 1996. [2.4.2](#)
- [6] DEL BIMBO, A., E. VICARIO und D. ZINGONI: *A Spatio-Temporal Logic for Image Sequence Coding and Retrieval*. In: *Proceedings IEEE VL'92 Workshop on Visual Languages*, 1992. [2.2](#)
- [7] DEL BIMBO, A., E. VICARIO und D. ZINGONI: *Sequence Retrieval by Contents through Spatio Temporal Indexing*. In: *Proceedings IEEE VL'93 Workshop on Visual Languages*, Seiten 88–92, 1993. [2.2](#), [2.4.2](#)
- [8] DEL BIMBO, A., E. VICARIO und D. ZINGONI: *Symbolic Description and Visual Querying of Image Sequences Using Spatio-Temporal Logic*. IEEE Transactions on Knowledge and Data Engineering, 7(4):609–622, August 1995. [2.2](#), [2.4.2](#)
- [9] BÖHME, GERT: *Fuzzy-Logik*. Springer, Berlin, Heidelberg, New York, 1993. [4.2](#), [4.2](#), [4.2.2](#)

- [10] DO CARMO, MANFREDO P.: *Differentialgeometrie von Kurven und Flächen*. Vieweg, Braunschweig, Wiesbaden, 1993. [2.3](#)
- [11] CLARKE, B. L.: *A Calculus of Individuals Based on Connection*. Notre Dame Journal of Formal Logic, 2(3), 1981. [2.2.3](#)
- [12] CLEMENTINI, ELISEO, PAOLINO DI FELICE und DANIEL HERNÁNDEZ: *Qualitative representation of positional information*. Artificial Intelligence, 95(2):317–356, 1997. [2.2.4](#), [2.5](#), [3.2.3](#), [4](#), [4.1.2](#), [4.1.4](#)
- [13] COHN, ANTHONY G.: *Qualitative Spatial Representation and Reasoning Techniques*. In: BREWKA, G., C. HABEL und B. NEBEL (Herausgeber): *Proceedings of KI-97*, Nummer 1303 in *Lecture Notes in Artificial Intelligence*, Seiten 1–30, Berlin, Heidelberg, New York, 1997. Springer. [1.3](#), [2.2](#)
- [14] COHN, ANTHONY G. und ACHILLE C. VARZI: *Connection Relations in Mereotopology*. In: PRADE, HENRI [70], Seiten 150–154. [2.2.3](#)
- [15] COHN, ANTHONY G. und ACHILLE C. VARZI: *Modes of Connection*. In: FREKSA, CHRISTIAN und DAVID M. MARK [31], Seiten 299–314. [2.2.3](#)
- [16] DEAN, THOMAS (Herausgeber): *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, San Francisco, CA, August 1999. Morgan Kaufman Publishers, Inc. [62](#), [76](#), [83](#)
- [17] DOUGLAS, D. H. und T. K. PEUCKER: *Algorithms for reduction of the number of points required to represent a digitized line or its caricature*. The Canadian Cartographer, 10(2):112–122, 1973. [5.2](#)
- [18] EGENHOFER, M. und J. HERRING: *A Mathematical Framework for the Definition of Topological Relationships*. In: *Fourth International Symposium on Spatial Data Handling*, Zürich, Juli 1990. [12](#)
- [19] EGENHOFER, M., J. SHARMA und D. MARK: *A Critical Comparison of the 4-Intersection and 9-Intersection Models for Spatial Relations: Formal Analysis*. In: MCMASTER, R. und M. ARMSTRONG (Herausgeber): *Autocarto 11*, Minneapolis, MN, Oktober 1993. [12](#)
- [20] EGENHOFER, MAX J. und ROBERT D. FRANZOSA: *Point-Set topological spatial relations*. International Journal of Geographical Information Systems, 5(2):161–174, 1991. [2.2.3](#)
- [21] EISENKOLB, ANDREAS, ALEXANDRA MUSTO, KERSTIN SCHILL, D. HERNÁNDEZ und WILFRIED BRAUER: *Representational Levels for the Perception of the Courses of Motion*. In: FREKSA, C. et al. [28], Seiten 129–155. [2.3](#)

-
- [22] FALTINGS, B.: *Qualitative Kinematics in mechanisms*. In: *Proc. IJCAI-87*, Seiten 436–442, Detroit, 1987. [2.4.1](#)
- [23] FORBUS, K.D., PAUL NIELSEN und BOI FALTINGS: *Qualitative Kinematics: A Framework*. In: WELD, DANIEL S. und JOHAN DE KLEER (Herausgeber): *Readings in Qualitative Reasoning about Physical Systems*, Seiten 562–567, San Mateo, California, 1990. Morgan Kaufmann Publishers Inc. [2.4.1](#)
- [24] FORBUS, KENNETH D.: *Qualitative reasoning about space and motion*. In: GENTNER, DEDRE und ALBERT L. STEVENS [[37](#)], Seiten 53–73. [2.4.1](#)
- [25] FORBUS, KENNETH D., PAUL NIELSEN und BOI FALTINGS: *Qualitative spatial reasoning: The CLOCK project*. *Artificial Intelligence*, 51:417–471, 1991. [2.2.1](#)
- [26] FRANK, ANDREW U. und WERNER KUHN (Herausgeber): *Spatial Information Theory. A Theoretical Basis for GIS. European Conference, COSIT'95*, Band 988 der Reihe *Lecture Notes in Computer Science*, Berlin, Heidelberg, New York, September 1995. Springer. [34](#), [35](#), [81](#)
- [27] FREKSA, C., W. BRAUER, C. HABEL und K. WENDER (Herausgeber): *Spatial Cognition II*, Band 1849 der Reihe *Lecture Notes in Artificial Intelligence*. Springer, 2000. [60](#), [63](#)
- [28] FREKSA, C., C. HABEL und K. WENDER (Herausgeber): *Spatial Cognition. An Interdisciplinary Approach to Representing and Processing Spatial Knowledge*, Band 1404 der Reihe *Lecture Notes in Artificial Intelligence*. Springer, 1998. [21](#), [48](#), [85](#), [88](#), [89](#)
- [29] FREKSA, CHRISTIAN: *Temporal reasoning based on semi-intervals*. *Artificial Intelligence*, 54:199–227, 1992. [11](#), [2.4.2](#), [2](#)
- [30] FREKSA, CHRISTIAN: *Using Orientation Information for Qualitative Spatial Reasoning*. In: FRANK, ANDREW U., I. CAMPARI und FORMENTINI U. (Herausgeber): *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, Nummer 639 in *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, New York, 1992. [2.2.4](#)
- [31] FREKSA, CHRISTIAN und DAVID M. MARK (Herausgeber): *Spatial Information Theory. Cognitive and Computational Foundations of Geographic Information Science. International Conference COSIT'99*, Band 1661 der Reihe *Lecture Notes in Computer Science*, Berlin, Heidelberg, New York, August 1999. Springer. [15](#), [47](#), [86](#)

- [32] FREKSA, CHRISTIAN und RALF RÖHRIG: *Dimensions of Qualitative Spatial Reasoning*. In: *Proc. of QUARDET (Qualitative Reasoning in Decision Technologies)*, Barcelona, Juni 1993. [1.3](#)
- [33] FREKSA, CHRISTIAN und KAI ZIMMERMANN: *On the Utilization of Spatial Structures for Cognitively Plausible and Efficient Reasoning*. In: *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Chicago, Oktober 1992. [2.2.4](#)
- [34] FRITSCH, EMMANUEL und JEAN PHILIPPE LAGRANGE: *Spectral Representations of Linear Features for Generalisation*. In: FRANK, ANDREW U. und WERNER KUHN [\[26\]](#), Seiten 157–171. [5.2](#)
- [35] GALTON, ANTONY: *Towards a Qualitative Theory of Movement*. In: FRANK, ANDREW U. und WERNER KUHN [\[26\]](#), Seiten 377–396. [2.4.2](#)
- [36] GALTON, ANTONY: *Continuous Change in Spatial Relations*. In: HIRTLE, STEPHEN C. und ANDREW U. FRANK (Herausgeber): *Spatial Information Theory. A Theoretical Basis for GIS*, Nummer 1329 in *Lecture Notes in Computer Science*, Berlin, Heidelberg, New York, 1997. Springer. [2.4.2](#)
- [37] GENTNER, DEDRE und ALBERT L. STEVENS (Herausgeber): *Mental Models*. Lawrence Erlbaum, Hillsdale, NJ, 1983. [3](#), [24](#), [57](#)
- [38] GOTTS, NICK M.: *How Far Can We ‘C’? Defining a Doughnut Using Connection Alone*. In: DOYLE, J., E. SANDEWALL und P. TORASSO (Herausgeber): *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourth International conference (KR94)*, Seiten 246–257. Morgan Kaufmann, 1994. [2.4.1](#)
- [39] GROS, BRYAN L., RANDOLPH BLAKE und ERIC HIRIS: *Anisotropies in visual motion perception: A fresh look*. *Journal of the Optical Society of America A*, 15(8):2003–2011, August 1998. [4.1.2](#)
- [40] GRÜSSER, OTTO-JOACHIM: *Multimodal structure of the extrapersonal space*. In: HEIN, A. und M. JEANNEROD (Herausgeber): *Spatially oriented behavior*, Seiten 327–352. Springer, Berlin, Heidelberg, New York, 1983. [4.1.1](#)
- [41] HAYES, PATRICK J.: *The Naive Physics Manifesto*. In: MICHIE, DONALD (Herausgeber): *Expert Systems in the Micro Electronic Age*, Seiten 242–270. Edinburg University Press, Edinburgh, UK, 1979. [2.2.1](#)
- [42] HERNÁNDEZ, DANIEL: *Qualitative Representation of Spatial Knowledge*, Band 804 der Reihe *Lecture Notes in Artificial Intelligence*. Springer, Berlin, Heidelberg, New York, 1994. [11](#), [2.2.4](#), [3.2.2](#), [4](#), [4.1.2](#), [5.3.3](#)

-
- [43] JOHANSSON, GUNNAR: *Visual perception of biological motion and a model for its analysis*. Perception & Psychophysics, 14(2):201–211, 1973. 1, 7, 2.3
- [44] JOHANSSON, GUNNAR: *Visual Motion Perception*. Scientific American, 232(4):76–88, 1975. 1, 7, 2.3
- [45] JOHANSSON, GUNNAR: *Spatio-Temporal Differentiation and Integration in Visual Motion Perception*. Psychological Research, (38):379–393, 1976. 1, 7, 2.3
- [46] KÄSTNER, ERICH: *Emil und die Detektive*. Cecilie Dressler Verlag, Berlin, ohne Jahr. 1.1
- [47] KETTANI, DRISS und BERNARD MOULIN: *A spatial model based on the notions of spatial conceptual map and of object's influence areas*. In: FREKSA, CHRISTIAN und DAVID M. MARK [31], Seiten 401–416. 9.2
- [48] KLATZKY, ROBERTA L.: *Allocentric and egocentric spatial representations: Definitions, distinctions and interconnections*. In: FREKSA, C. et al. [28], Seiten 1–18. 6.1
- [49] KRAPP, LORENZ: *Qualitative Repräsentation von Bewegungsabläufen*. Institut für Informatik der TU München, 1998. Systementwicklungsprojekt. 9.2
- [50] KUIPERS, BENJAMIN: *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. MIT Press, Cambridge, Mass., 1994. 2.2.1
- [51] LEVINSON, S. C.: *Frames of reference and Molyneux's question*. In: BLOOM, PAUL, MARY A. PETERSON, LYNN NADEL und MERRILL F. GARRETT (Herausgeber): *Language and Space*, Seiten 109–169. MIT Press, Cambridge, MA, 1996. 24
- [52] LIGOZAT, GÉRARD: *Weak Representations of Interval Algebras*. In: *Proc. of AAAI-90*, Seiten 715–720, 1990. 11
- [53] LIGOZAT, GÉRARD: *On generalized interval calculi*. In: *Proc. of AAAI-91*, Seiten 234–240, 1991. 11
- [54] LIGOZAT, GÉRARD und HÉLÈNE BESTOUGEFF: *On Relations Between Intervals*. Information Processing Letters, 32:177–182, 1989. 11
- [55] LIU, JIMING: *A Method of Spatial Reasoning Based on Qualitative Trigonometry*. Artificial Intelligence, 98(1-2):137–168, 1998. 4.1.1
- [56] LUDWIG, RALF: *Kant für Anfänger. Die Kritik der reinen Vernunft*. Deutscher Taschenbuch Verlag, München, 1995. 10

- [57] McCloskey, Michael: *Naive Theories of Motion*. In: GENTNER, DEDRE und ALBERT L. STEVENS [37], Seiten 299–324. 2.2.1
- [58] MULLER, P.: *A Qualitative Theory of Motion Based on Spatio-temporal Primitives*. In: COHN, A.G., L.K. SCHUBERT und S.C. SHAPIRO (Herausgeber): *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixth International Conference (KR'98)*. Morgan Kaufmann, 1998. 2.4.2
- [59] MULLER, P.: *Space-Time as a Primitive for Space and Motion*. In: *Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS98)*, Frontiers in Artificial Intelligence and Applications. IOS Press, 1998. 2.4.2
- [60] MÜLLER, R., T. RÖFER, A. LANKENAU, A. MUSTO, K. STEIN und A. EISENKOLB: *Coarse Qualitative Descriptions in Robot Navigation*. In: FREKSA, C. et al. [27], Seiten 265–276. 2, 8.1.2, 8.4, 8.5
- [61] MUSIL, ROBERT: *Die Verwirrungen des Zöglings Törleß*. Rowohlt Taschenbuch Verlag, Hamburg, 1959. 1.1
- [62] MUSTO, ALEXANDRA, KLAUS STEIN, ANDREAS EISENKOLB und THOMAS RÖFER: *Qualitative and Quantitative Representations of Locomotion and their Application in Robot Navigation*. In: DEAN, THOMAS [16], Seiten 1067–1073. 2, 8.1, 8.1.1, 8.2, 8.3
- [63] MUSTO, ALEXANDRA, KLAUS STEIN, ANDREAS EISENKOLB, KERSTIN SCHILL, THOMAS RÖFER und WILFRIED BRAUER: *From Motion Observation to Qualitative Motion Representation*. In: FREKSA, C. et al. [27], Seiten 115–126. 1, 2, 7, 2.3, 2.3
- [64] NEBEL, BERNHARD: *Solving Hard Qualitative Temporal Reasoning Problems: Evaluating the Efficiency of Using the ORD-Horn Class*. In: *Proceedings of the 12th European Conference on Artificial Intelligence (ECAI'96)*, Seiten 38–42, 1996. 2.2.4
- [65] NEBEL, BERNHARD und HANS-JÜRGEN BÜRCKERT: *Reasoning about Temporal Relations: A Maximal Tractable Subclass of Allen's Interval Algebra*. Journal of the ACM, 42(1):43–66, 1995. 2.2.4
- [66] NUNES, JOAN: *General Concepts of Space and Time*. In: FRANK, ANDREW (Herausgeber): *GIS-Material for a Post-Graduate Course*, Band 1, Seiten 7–34. TU Wien, 1995. 2.1
- [67] O'KEEFE, JOHN und LYNN NADEL: *The hippocampus as a cognitive map*. The Clarendon Press, Oxford, 1978. 24

-
- [68] PERSSON, JONAS und ERLAND JUNGERT: *Generation of multi-resolution maps from run-length-encoded data*. International Journal of Geographical Information Systems, 6(6):497–510, 1992. [5.2](#), [5.2.2](#)
- [69] PÖPPEL, E.: *A hierarchical model of temporal perception*. Trends in Cognitive Sciences, 1(2):312–320, 1997. [3.2.1](#)
- [70] PRADE, HENRI (Herausgeber): *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*. John Wiley & Sons, 1998. [14](#), [72](#)
- [71] PRITCHARD, R. M.: *Stabilized images on the retina*. Scientific American, (204):72–78, 1961. [7](#)
- [72] PUGET, JEAN-FRANCOIS: *Constraint Programming: A Great AI Success*. In: PRADE, HENRI [70], Seiten 698–705. [4](#), [2.2.4](#)
- [73] RANDELL, D. A., Z. CUI und A. G. COHN: *A Spatial Logic Based on Regions and Connection*. In: *Proceedings 3rd International Conference on Knowledge Representation and Reasoning*, Seiten 165–176, San Mateo, CA, 1992. Morgan Kaufmann. [2.2.3](#)
- [74] REICHARDT, WERNER: *Autokorrelations-Auswertung als Funktionsprinzip des Zentralnervensystems*. Zeitschrift für Naturforschung, (12b):448–457, 1957. [2.3](#)
- [75] RENZ, JOCHEN: *A Canonical Model of the Region Connection Calculus*. In: COHN, A.G., L. SCHUBERT und S.C. SHAPIRO (Herausgeber): *Principles of Knowledge Representation and Reasoning, Proceedings of the 6th International Conference (KR'98)*, Seiten 330–341, 1998. [2.2.4](#)
- [76] RENZ, JOCHEN: *Maximal Tractable Fragments of the Region Connection Calculus: A Complete Analysis*. In: DEAN, THOMAS [16], Seiten 448–454. [2.2.4](#)
- [77] RENZ, JOCHEN und BERNHARD NEBEL: *On the Complexity of Qualitative Spatial Reasoning: A Maximal Tractable Fragment of the Region Connection Calculus*. In: *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI'97)*, Seiten 522–527, 1997. [2.2.4](#)
- [78] RÖFER, THOMAS und AXEL LANKENAU: *Architecture and Applications of the Bremen Autonomous Wheelchair*. In: WANG, P. P. (Herausgeber): *Proceedings of the Fourth Joint Conference on Information Systems*, Band 1, Seiten 365–368. Association of Intelligent Machinery, 1998. [8.1](#)
- [79] SADALLA, E.K., W.J. BURROUGHS und L.J. STAPLIN: *Reference Points in Spatial Cognition*. Journal of Experimental Psychology: Human Learning and Memory, 6:516–528, 1980. [3](#), [2](#)

- [80] SCHILL, KERSTIN und CHRISTOPH ZETZSCHE: *A model of visual spatio-temporal memory: The icon revisited*. Psychological Research, 57:88–102, 1995. [5.3.2](#)
- [81] SCHLIEDER, CHRISTOPH: *Reasoning about Ordering*. In: FRANK, ANDREW U. und WERNER KUHN [26], Seiten 341–349. [2.2.4](#)
- [82] STEIN, KLAUS: *Generalisierung und Segmentierung von qualitativen Bewegungsdaten*. Diplomarbeit, Institut für Informatik an der TU München, 1998. [4.1.4](#), [5.2.2](#), [5.3.1](#), [5.3.1](#)
- [83] STOCK, OLIVIERO: *Was the Title of this Talk generated automatically? Prospects on Intelligent Interfaces and Language*. In: DEAN, THOMAS [16], Seiten 1412–1419. [2.2](#)
- [84] TVERSKY, BARBARA: *Cognitive Maps, Cognitive Collages, and Spatial Mental Models*. In: FRANK, ANDREW U. und IRENE CAMPARI (Herausgeber): *Spatial Information Theory. A Theoretical Basis for GIS. European Conference, COSIT'93*, Band 716 der Reihe *Lecture Notes in Computer Science*, Seiten 14–24, Berlin, Heidelberg, New York, September 1993. Springer. [3](#)
- [85] TVERSKY, BARBARA und PAUL U. LEE: *How Space Structures Language*. In: FREKSA, C. et al. [28], Seiten 157–177. [8](#), [8.1.2](#)
- [86] TVERSKY, BARBARA und PAUL U. LEE: *Pictorial and verbal tools for conveying routes*. In: FREKSA, CHRISTIAN und DAVID M. MARK [31], Seiten 51–64. [8.1.2](#)
- [87] VERFAILLIE, KARL und GÉRY D'YDEWALLE: *Representational Momentum and Event Course Anticipation in the Perception of Implied Periodical Motions*. Journal of Experimental Psychology, 17(2):302–313, 1991. [9.2](#)
- [88] WERNER, STEFFEN, CHRISTINA SAADE und GERD LÜER: *Relations Between the Mental Representation of Extrapersonal Space and Spatial Behavior*. In: FREKSA, C. et al. [28], Seiten 107–127. [1](#)
- [89] ZIMMER, HUBERT D., HARRY R. SPEISER, JÖRG BAUS, ANSELM BLOCHER und EVA STOPP: *The Use of Locative Espressions in Dependence of the Spatial Relation between Target and Reference Object in Two-Dimensional Layouts*. In: FREKSA, C. et al. [28], Seiten 223–240. [4.1.2](#)