

Service-roboter und intuitive Mensch-Roboter- Interaktion

Habilitationsschrift

vorgelegt der

Fakultät für Informatik und Automatisierung
der Technischen Universität Ilmenau

von

Dr.-Ing. Hans-Joachim Böhme

Oktober 2001

Danksagung

Die Arbeit stellt das Ergebnis der langjährigen Arbeit des Autors am Fachgebiet Neuroinformatik der TU Ilmenau dar. Maßgeblich zu nennen sind dabei zunächst die beiden durch das Thüringer Ministerium für Wissenschaft, Forschung und Kultur geförderten Projekte GESTIK (GESTenbasierte Interaktion und Kommunikation mit dem mobilen Robotersystem MILVA, 1996 bis 1998) und PERSES (PERSONenlokalisation und Personentracking für mobile SERVICEsysteme, 1998 bis 2001), die der Autor als Projektleiter betreute. Fachliche Beiträge der KollegInnen Anja Brakensiek, Markus Krabbes, Ulf-Dietrich Braumann, Jürgen Key und Torsten Wilhelm sind in diese Arbeit an verschiedenen Stellen eingeflossen. Die Mitarbeiter der fachgebietsinternen Projektgruppe NAASA (Neural Architectures for Acoustic Scene Analysis), die überwiegend vom DFG-Graduiertenkolleg GK164 gefördert wurden, erbrachten ebenfalls Beiträge zu dieser Arbeit.

Darüber hinaus sei an dieser Stelle allen jetzigen und ehemaligen Kollegen gedankt für das stets sehr gute Arbeitsklima und die stets konstruktive und außerordentlich angenehme Zusammenarbeit. Mein besonderer Dank gilt unserem Fachgebietsleiter, Prof.Dr. Horst-Michael Groß, für seine unermüdliche Ermunterung und die fachliche Begleitung, meinem Zimmerkollegen Dr.-Ing. Klaus Debes für die nie erschöpfende Diskussionsbereitschaft und so auch manche persönliche Aufmunterung, sowie unseren beiden Technikerinnen Sabine Schulz und Heike Groß, die stets unauffällig, aber wirkungsvoll dafür sorgten, dass immer arbeitsfähige Technik zur Verfügung stand.

Ohne unsere Studenten und deren aktive und konstruktive Mitarbeit hätte das Projekt nie den aktuellen Stand erreichen können. Genannt seien hier insbesondere Torsten Hempel, Christian Martin, Christof Schröter, Andre Liebetau und Stefan Weber.

Gäbe es meine Familie nicht, wäre diese Arbeit sicher ebenfalls nie zustande gekommen. Welche Last meiner Familie in den vergangenen Wochen und Monaten aufgebürdet wurde, lässt sich nur schwer in Worte fassen. Meiner Frau Ines und unserem Töchterchen Lena deshalb ein liebevolles Dankeschön für alle Unterstützung und das entgegengebrachte Verständnis. Ihnen möchte ich diese Arbeit widmen.

Inhaltsverzeichnis

I	Einführung	9
1	Serviceroboter - auf dem Weg ins tägliche Leben	11
1.1	Serviceroboter im Spiegel der Medien	11
1.2	Begriffsdefinition	15
1.3	Beispiele für Serviceroboter	15
1.4	Serviceroboter und Mensch-Maschine-Interaktion	18
1.5	Marktentwicklung	19
1.6	Zielstellung der Arbeit	20
1.7	Struktur der Arbeit	22
2	Basiskomponenten interaktiver Serviceroboter	25
2.1	Sensorik und Aktuatorik	25
2.1.1	Interne Sensoren	25
2.1.2	Externe Sensoren	26
2.1.3	Aktuatorik	27
2.2	Navigation	29
2.2.1	Grundproblem und Begriffsdefinition	29
2.2.2	Lokale Navigationsverfahren und Hindernisvermeidung	31
2.2.3	Verfahren zur globalen Navigation	31
2.2.4	Umgebungsmodelle	32
2.2.4.1	Koppelnavigation	35
2.2.4.2	Probabilistische Verfahren	36
2.2.5	Pfadplanung und Bewegungssteuerung	36
2.3	Steuerarchitektur und Verhaltenskoordination	37
2.3.1	Sense-Plan-Act-Paradigma	37
2.3.2	Subsumption-Architektur	38
2.3.3	Dreiebenen-Architektur	39
2.3.4	Verhaltensbasierte Architekturen	40
2.3.5	Hybride Steuerarchitekturen	41
2.4	Mensch-Roboter-Interaktion (MRI)	42
2.4.1	Klassische MRI	43
2.4.2	Intuitive, natürliche MRI	43
2.4.2.1	Sprachbasierte Interaktion	44
2.4.2.2	Visuell-basierte Interaktion	45
2.4.3	Roboterartikulation	49

2.5	Lernfähigkeit	49
2.5.1	Sensordatenverarbeitung	50
2.5.2	Direkte Sensor-Aktor-Abbildungen	50
2.5.3	Lernen von Umgebungsmodellen	50
2.5.4	Lernen und MRI	51
2.6	Szenariospezifische Serviceleistungen	52
2.7	Zusammenfassung	53
II	Szenarien und Applikationen	55
3	Tourguide-Roboter	57
3.1	Der Roboter POLLY als Laborführer	57
3.2	RHINO und MINERVA	58
4	Assistenz-Roboter	63
4.1	Nursebot-Szenarien	63
4.1.1	Care-O-Bot	65
4.1.2	FLO und PEARL	67
4.1.3	DIROKOL	68
4.1.4	Projekt MORPHA	69
4.2	Weitere Assistenzsysteme	71
4.2.1	Produktionsassistent im MORPHA-Projekt	71
4.2.2	HERMES	72
4.2.3	ARNOLD	74
4.2.4	COLIN	75
5	Resümee	79
III	PERSES - ein Intelligenter Interaktiver Mobiler Shopping-Assistent	81
6	Das BauMarkt-Szenario	83
6.1	Vorgeschichte	83
6.2	Typische Eigenschaften der BauMarktumgebung	86
6.3	Szenariospezifische Serviceleistungen	86
6.4	Die Experimentalplattform PERSES	88
6.5	Navigation	89
6.6	Mensch-Roboter-Interaktion	92
6.7	Steuerarchitektur und Verhaltenskoordination	93
6.8	Zusammenfassung	95

7	Navigation in der BauMarkt-Umgebung	97
7.1	Hindernisvermeidung und lokale Umgebungskarte	98
7.1.1	Basisrepräsentation zur Umgebungsmodellierung	98
7.1.2	Ultraschall-basiertes Verfahren zur Hinderniswahrnehmung	98
7.1.3	Visuell-basiertes Verfahren zur Hinderniswahrnehmung	100
7.2	Aufbau einer globalen Umgebungskarte	114
7.2.1	Visuelle Odometriekorrektur	115
7.2.2	Eigentlicher Kartenaufbau	118
7.2.3	Exploration versus manuelle Führung	120
7.2.4	Probleme und Limitierungen	121
7.3	Visuell-basierte Zustandsschätzung (Selbstlokalisierung)	122
7.3.1	Markov-Lokalisierung und Monte-Carlo-Verfahren	123
7.3.1.1	Markov-Lokalisierung	124
7.3.1.2	Monte-Carlo-Lokalisierung und Condensation-Algorithmus	125
7.3.2	Visuelle Monte-Carlo-Lokalisierung (MCL)	127
7.3.2.1	Visuelle Umgebungsrepräsentation	128
7.3.2.2	Visuell-basiertes Sensormodell	130
7.3.2.3	Bewegungsmodell	130
7.3.2.4	Experimentelle Ergebnisse	131
7.3.2.5	Probleme und Limitierungen	133
7.4	Pfadplanung und Bewegungssteuerung	133
7.4.1	Pfadplanung	133
7.4.1.1	Dijkstra-Algorithmus	134
7.4.1.2	Pfadoptimierung	137
7.4.2	Bewegungssteuerung	140
7.4.2.1	Potentialfeld-Verfahren	141
7.4.2.2	Vektorfeld-Histogramm	145
7.5	Zusammenfassung	150
8	Mensch-Roboter-Interaktion	155
8.1	Aufmerksamkeitssteuerung	155
8.1.1	Visuelle Aufmerksamkeit	156
8.1.2	Akustische Aufmerksamkeit	159
8.1.3	Fusion der Aufmerksamkeitsquellen	159
8.2	Personenverifikation	160
8.2.1	Pyramidale Repräsentation	161
8.2.1.1	Abstufung und Glättung	163
8.2.1.2	Unterabtastung und Interpolation	164
8.2.2	Hautfarbanalyse	165
8.2.3	Kopfschulterpartiedetektion	167
8.2.3.1	Gewinnung eines Konturmodells	167
8.2.3.2	Eigentliche Konturdetektion	168
8.2.4	Gesichtsdetektion	170
8.2.5	Fusion und Selektion	172
8.2.5.1	Kodierung der Einzelcues	172

8.2.5.2	Fusion der Einzelcues	172
8.2.5.3	Dynamische Selektion	173
8.2.6	Experimentelle Ergebnisse	177
8.3	Visuelles Personentracking	179
8.3.1	Condensation-Algorithmus	182
8.3.2	Erweiterungen	185
8.3.3	Experimentelle Ergebnisse	185
8.4	Weitere Verfahren zur visuell-basierten MRI	189
8.5	Grafisches Nutzerinterface	189
8.6	Sprachausgabe und Robotergesicht	190
8.7	Zusammenfassung	192
9	Steuerarchitektur und Interaktionszyklus	195
9.1	Koordination von Teilverhalten - Grundidee des Dynamic Approach	195
9.2	Gesamtablaufsteuerung	197
9.3	Interaktionszyklus	198
10	Zusammenfassung und Ausblick	201
10.1	Zusammenfassung und Wertung der Ergebnisse	201
10.1.1	Navigation	202
10.1.2	Mensch-Roboter-Interaktion	202
10.1.3	Steuerarchitektur	203
10.1.4	Serviceleistungen	203
10.1.5	Übergreifende Aspekte	204
10.2	Offene Probleme und Ausblick	205
	Literatur	207

Teil I
Einführung

Kapitel 1

Serviceroboter - auf dem Weg ins tägliche Leben

Serviceroboter sind in aller Munde, die Visionen über ihren (mehr oder weniger sinnvollen) Einsatz nahezu grenzenlos. Der Mensch träumt von „dienstbaren Gesellen“, die ihm sein mühsames Dasein erleichtern und rund um die Uhr verfügbar sind. Aber wie sieht der aktuelle Entwicklungsstand tatsächlich aus? Wie weit ist die Entwicklung bereits vorangeschritten, um diesem Ziel näher zu kommen? Welche Verfahren und Methoden sind bereits so robust, dass sie sich für einen praktischen Einsatz unter realen Bedingungen wirklich eignen? Und wo besteht noch akuter Forschungs- und Entwicklungsbedarf, um auf dem Weg zu den „dienstbaren Gesellen“ weiter voranzuschreiten? All dies sind Fragen, denen im Rahmen dieser Arbeit nachgegangen werden soll.

Abschnitt 1.1 analysiert zunächst die gesellschaftliche Auseinandersetzung mit dieser neuen Technologie. Daran anschließend werden im Abschnitt 1.2 einige grundlegende Begriffe definiert, die für das weitere Verständnis der Arbeit von Bedeutung sind. Anhand einiger ausgewählter Beispielapplikationen verdeutlicht Abschnitt 1.3, wie breit das Anwendungsspektrum für solche Systeme gefächert ist, bevor in Abschnitt 1.4 auf den Aspekt der Mensch-Roboter-Interaktion im Kontext von Serviceroboter-Anwendungen prinzipiell eingegangen wird. Abschnitt 1.5 beleuchtet gegenwärtige marktwirtschaftliche Entwicklungstendenzen anhand aktueller Studien. Als Resümee der vorangegangenen Abschnitte ist dann Abschnitt 1.6 der Darstellung der Zielstellung der vorliegenden Arbeit gewidmet.

1.1 Serviceroboter im Spiegel der Medien

Das zunehmende gesellschaftliche Interesse spiegelt sich in der rasant gestiegenen Zahl der Beiträge in den verschiedensten Medien wider. Fernsehsendungen berichten über neueste Entwicklungen und Szenarien, verschiedenste Printmedien veröffentlichen entsprechende Beiträge, was letztendlich Ausdruck der Auseinandersetzung der Gesellschaft mit aktuellen technologischen Entwicklungen ist.

Stellvertretend soll zunächst ein Beitrag der Zeitschrift *Automation & Qualität* aus deren Juni-Ausgabe des Jahres 2000 unter dem Titel *Von der Werkshalle ins Wohnzimmer - das Leben mit Servicerobotern* fast unverändert wiedergegeben werden [htt00]:

Am Ende der Servicewüste *Die Möglichkeiten scheinen grenzenlos zu sein. Serviceroboter schrubben Bahnhöfe, tragen Koffer und helfen Pflegebedürftigen im Alltag. Von Dienstleistungswüste keine Spur. Die Entwicklung schreitet voran. Roboter verlassen die Werkshallen und erobern den Dienstleistungssektor. Gründe sind überproportionales Wachstum des Dienstleistungsbereichs und Fortschritte bei Sensor-, Steuerungs- und Antriebstechnik. Bei der Entwicklung vom Industrieroboter zum Personalroboter sind Serviceroboter aber nur eine Zwischenstufe. Schon heute kaum mit Standardkomponenten zu verwirklichen, werden sie in Zukunft noch individueller nach Art, Umfeld und Ablauf der Aufgabe angepasst sein. Die meisten Serviceroboter sind mobil und müssen sich - im Gegensatz zu Industrierobotern - ständig in einer anderen Umgebung zurechtfinden. Ihr künftiger Haushaltsroboter sollte beispielsweise wissen, wo er beim Staubsaugen entlangfahren muss - auch wenn Sie täglich Ihre Möbel umstellen. Wie gut Serviceroboter künftig solche Aufgaben lösen, hängt davon ab, wie schnell und wie korrekt sie herausfinden, wo sie sind, wohin sie wollen und wie sie dahin kommen. Wo sich ein Roboter befindet, berechnet er aus den Umgebungsdaten, die ihm seine Sensoren liefern. Er konstruiert ein Bild seiner Umgebung - das Umweltmodell. Neben den einfach zu handhabenden und kostengünstigen Ultraschallsensoren kommen hier immer häufiger optische Laserscanner zum Einsatz, die die Position eines Objektes sehr genau bestimmen können. Um Messfehler zu vermeiden, arbeiten Entwickler derzeit intensiv daran, verschiedene Sensoren miteinander zu kombinieren und die Daten sinnvoll miteinander zu verrechnen. Bewegungsplanung und Bewegungssteuerung gewährleisten, dass der Roboter sich auf dem richtigen Weg befindet. Mit Hilfe des Umweltmodells erzeugt die Bewegungsplanung Wegpunkte, die anzusteuern sind. Die Bewegungsführung berechnet die Bahn des Roboters zwischen den Wegpunkten und ermöglicht Ausweichmanöver, falls unvorhergesehene Hindernisse wie sich bewegende Personen den Weg versperren. Um die Orientierung des Roboters auf seinem Weg zu bestimmen, wurden neue Sensorsysteme entwickelt, die schwingende Strukturen enthalten. Aktuelle Arbeiten befassen sich mit der Miniaturisierung dieser Strukturen, die sich dann auf dem gleichen Chip wie die Auswertungselektronik unterbringen lassen.*

Bewährt: Scheuern und Saugen mit Robotern *Als Staubsauger allein sind Reinigungsroboter überqualifiziert - zum Beispiel die lernfähige Scheuer-Saugmaschine „BR 700 Robot“ von Kärcher. Der Roboter eignet sich speziell zur Reinigung großer Flächen etwa auf Flughäfen, Bahnhöfen und in Produktionshallen. Programmiert wird er durch einmaliges Abfahren von bis zu sechs Reinigungsweegen. Der Roboter benötigt keine künstlichen Markierungen auf der Reinigungsstrecke, da er die Umgebung mit Ultraschallsensoren erfasst und eventuelle Wegabweichungen korrigiert. Zum Schutz von Personen oder Gegenständen auf der Fahrbahn dienen verschiedene Sicherheitssysteme: vorne ein Bumper, seitlich und hinten Kontaktleisten, die den Roboter gegebenenfalls sofort anhalten und alle Funktionen abschalten können. Eine Hinderniserkennung über Ultraschallsensoren ist nach vorn (Auffahrschutz), oben (Unterfahrschutz) und unten (Treppenerkennung) wirksam.*

Der Roboter als Altenpfleger Für Reinigungsaufgaben im häuslichen Bereich ist „Care-O-bot“ vom Fraunhofer Institut für Produktionstechnik und Automatisierung geeignet. Dies allerdings nur am Rande. Das Care-Home-System soll nämlich älteren, unterstützungs- und pflegebedürftigen Menschen ein generell selbständigeres Leben in vertrauter Umgebung ermöglichen. Der eigenständig und auf Zuruf agierende Roboter kommuniziert zum Beispiel mit Arzt und Behörden, steuert die häusliche Infrastruktur wie Heizung, Fenster, Licht und Alarmanlage, bietet durch seitlich angebrachte Arme Aufsteh-, Stütz- und Gehhilfe und kann durch Überwachung von Vitalfunktionen und Notfallalarmierung zur persönlichen Sicherheit beitragen. Für die Bewegung des Roboters mit bis zu 1,5 m/s sorgen zwei bürstenlose Servomotoren, Differentialantrieb, zwei Vollgummi-Antriebsräder und vier Stützräder. Die 46 Ah-Akkus lassen sich in 10 min aufladen. Messräder, Laserscanner und Stereokamera erfassen die Umgebung. Seine Meldungen gibt der Roboter in gesprochener Form und auf einem 14-Zoll-TFT-Display aus.

Digitaler Zimmerservice Roboterbutler „Mortimer“ wird in Kürze Hotelgästen das Leben erleichtern. Der Serviceroboter entstand am Institut für Prozessrechen-technik und Robotik (IPR) der Universität Karlsruhe in Zusammenarbeit mit dem ebenfalls in Karlsruhe ansässigen Hotel Kübler. Der Prototyp soll mit Zimmerservice und Gepäcktransport typische Serviceaufgaben übernehmen. Unübersichtliche, weite Fahrstrecken, veränderliche Umweltgeometrie und Hindernisse wie sich bewegende Menschen sind die Regel. Laserscanner, Ultraschallsensoren und Kreiselkompass sorgen für sichere Navigation und Bewegung. Zentrales Entwicklungsziel ist die Integration des Roboters in die Serviceumgebung, das heißt Kommunikation mit der Haustechnik, Zugriff auf Fahrstühle oder Türen und Interaktion mit Menschen.

Briefträger aus Stahl „MoPS“ ist der Name des Roboter-Briefträgers am Institut für Robotik der Eidgenössischen Technischen Hochschule (ETH) Zürich. Das mobile Postverteilungssystem trägt die eingehende Post von der Zentrale zu den einzelnen Instituten im Gebäude. Dazu nimmt es die verschiedenen Behälter auf und fährt zu den entsprechenden Sekretariaten, die sich in unterschiedlichen Stockwerken befinden. Dort sammelt es die ausgehende Post ein und stellt sie in der zentralen Poststelle im Erdgeschoss ab. Der vorgegebene Tourenplan kann über das Internet abgeändert werden. Mit dem Warenaufzug gelangt der Roboter von einem in das andere Stockwerk. Infrarotsignale übernehmen die Kommunikation zwischen Roboter, Aufzug und automatischen Türöffnern. Personen auf seinem Weg weicht der Roboter aus. Bei Problemen schickt er dem zuständigen Personal eine E-Mail. Der „Briefträger“ passt seine Geschwindigkeit den äußeren Umständen an: Wenig Hindernisse und breite Gänge bedeuten höhere Geschwindigkeiten. Die durchschnittliche Geschwindigkeit liegt bei 0,5 m/s. Sind die Batterien des Roboters leer, fährt er selbständig eine Ladestation an und nimmt dort neue Batterien auf. Die wichtigsten Sensoren sind zwei Laserscanner vorn und hinten am „Briefträger“ zur groben Positionsabschätzung, sicheren Navigation und Hinderniserkennung. Für eine hohe Positioniergenauigkeit etwa zum Batteriewechsel sind zusätzlich zwei Infrarot-Triangulationssensoren vorgesehen.

Den Roboter im eigenen Leib „Wo bin ich?“ ist eine Frage, über die sich Operationsroboter nicht den „Kopf zerbrechen“ müssen. Sie sind an einem festen Ort installiert und tragen vom Arzt ferngesteuerte Instrumente. Erstmals in der Geschichte der Herzchirurgie gelang es 1999 am Universitätsklinikum Frankfurt, einen Defekt der Vorhofscheidewand des Herzens - ein sogenanntes Loch im Herzen - ohne eine Öffnung des Brustkorbs zu operieren - ausschließlich unter Einsatz des Operationsroboters des Klinikums. Durch drei kleine Schnitte von etwa einem Zentimeter Länge ließen sich die Roboterarme mit den chirurgischen Instrumenten und die 3-D-Kamera in den Körper einführen. Ein Austausch der Instrumente ist auch während der Operation innerhalb von 30 s möglich. Die Bewegungen des Chirurgen werden mit Hilfe einer variabel skalierbaren Übersetzung auf die „Hände“ des Roboters übertragen. Der Chirurg kann den Eingriff mit einer bis zu zehnfachen Vergrößerung kontrollieren. Neben der dreidimensionalen optischen Darstellung des Operationsfeldes ist das frei drehbare „Handgelenk“ an den Roboterarmen eine wesentliche Innovation. So können innerhalb des Brustkorbes alle Handbewegungen des Chirurgen nachvollzogen werden.

Roboter erfinden sich selbst Vielleicht erfinden sich Serviceroboter in nicht allzu ferner Zukunft selbst. US-Forscher von der Brandeis University bei Boston entwickelten die ersten polymorphen Roboter, die ihre Form veränderten Aufgabenstellungen anpassen können. Prototyp ist ein einfacher Roboter mit thermoplastischem Gehäuse. Um sich seiner Situation anzupassen, nutzt der Roboter das Verfahren des Rapid Prototyping. Er stapelt ein vom Computer berechnetes Modell mit Hilfe des 3-D-Druckers Schicht für Schicht aus dem thermoplastischen Material aufeinander. Die Forscher beauftragten ein neuronales Netzwerk mit der Aufgabe, einen Roboter zu konstruieren, der sich mit nur einem Bein und einem Motor bewegen kann. Der Computer berechnete viele verschiedene Möglichkeiten. Einige Varianten des entstandenen Prototyps kriechen wie ein Fisch auf dem Trockenen, andere krabbeln beispielsweise krabbenähnlich seitwärts. Doch noch müssen die Motoren vom Menschen in das Kunststoffgehäuse eingesetzt werden, die 3-D-Drucker sind noch groß und unhandlich. Ziel ist es aber, die Roboter völlig unabhängig zu machen. So könnten sie Teile ihres Körpers dem jeweiligen Bedarf anpassen - zum Beispiel einen Arm umformen, um ein neues Instrument für eine neue Situation nutzen zu können.

Obwohl auch in diesem Artikel (wie das leider üblich ist) Vision und aktueller Realisierungsstand gleitend ineinander übergehen, so werden doch die typischen Einsatzfelder von Servicerobotern mit ihren jeweiligen Charakteristika und Zielstellungen sehr treffend beleuchtet. Einige der o.g. Systeme werden sich entsprechend auch in dieser Arbeit an verschiedenen Stellen, entweder als Beispiel für eine mögliche Realisierung oder bei der Diskussion von Anwendungsszenarien, wiederfinden.

Weiterer Ausdruck der gesellschaftlichen Reflexion aktueller und zukünftiger Technologien sind Medienberichte, die sich mit dem Sinn und Unsinn von Roboterapplikationen beschäftigen (siehe z.B. [Ble01]). Daraus lässt sich schlussfolgern, dass letztendlich nur solche Applikationen erfolgreich und marktwirksam werden, die die notwendige gesellschaftliche Akzeptanz erfahren.

1.2 Begriffsdefinition

Was versteht man nun konkret unter einem Serviceroboter? In [Fra94] findet man folgende Definition:

Ein Serviceroboter ist eine freiprogrammierbare Bewegungseinrichtung, die teil- oder vollautomatisch Dienstleistungen verrichtet. Dienstleistungen sind dabei Tätigkeiten, die nicht der direkten industriellen Erzeugung von Sachgütern, sondern der Verrichtung von Leistungen an Menschen und Einrichtungen dienen.

Als wesentliche Teilkomponenten von Servicerobotern werden Handhabungsarm, Endeffektor, mobile Plattform, Antriebstechnik, Steuerung, Sensorik, Mensch-Maschine-Schnittstelle und Sicherheitssystem angeführt. Die konkrete Ausprägung der einzelnen Komponenten hängt vom dann letztlich vom vorliegenden Anwendungsfall, der vorliegenden Hardware und den jeweils verfügbaren bzw. notwendigen informationstechnischen Methoden ab. Diese Definition besitzt aus aktueller Sicht noch volle Gültigkeit. Deshalb erfolgt im Kontext der vorliegenden Arbeit keine Adaption dieser Definition, sondern lediglich eine Beschränkung auf interaktive mobile Serviceroboter, die gemeinsam und in kontinuierlicher Interaktion mit ihrem menschlichen Benutzer ihre Dienstleistung vollbringen sollen. Es werden also vorrangig Systeme eine Rolle spielen, die ihre eigentliche Servicefunktionalität maßgeblich aus dieser Interaktion beziehen.

Nachdem der Begriff des Serviceroboters definiert ist, muss nun geklärt werden, was in dieser Arbeit unter einem interaktiven mobilen Serviceroboter verstanden werden soll. Im Blickpunkt stehen mobile Serviceroboter, also wie auch immer geartete Systeme, die sich selbständig und mehr oder weniger autonom in ihrer Einsatzumgebung bewegen können. Unter deren Interaktionsfähigkeit kann man sowohl die Interaktion zwischen mehreren solcher Systeme als auch die Interaktion zwischen System und menschlichem Benutzer verstehen, wobei letzterem in dieser Arbeit besondere Aufmerksamkeit gewidmet werden wird.

Sicherlich am schwierigsten zu definieren ist der Begriff des intelligenten Serviceroboters, allein schon deshalb, weil sich für Intelligenz als solches sicher keine einheitliche Definition angeben lässt. Eher pragmatisch soll deshalb unter einem intelligenten System ein System verstanden werden, welches die Fähigkeit besitzt, ein dem Beobachter oder dem Interaktionspartner stets sinnfällig erscheinendes Verhalten an den Tag zu legen. Letztlich wird es dem menschlichen Interaktionspartner, dem Benutzer, immer darum gehen, ein System vorzufinden, welches sich abhängig von der konkreten Situation weitestgehend so verhält, wie man das von einem menschlichen Interaktionspartner auch erwarten würde. Damit einher geht die notwendige Fähigkeit dieser Systeme, aktuelle Wahrnehmungen in den aktuellen Verhaltenskontext einzuordnen und daraus flexibel entsprechende Aktionen abzuleiten. Dieser hier vertretenen Sichtweise kommt auch der Begriff der „Kognitiven Robotik“ recht nahe, der die Integration von Wahrnehmung und Aktionsgenerierung auf einer physischen Plattform, die mit ihrer Umwelt interagiert, beschreibt.

1.3 Beispiele für Serviceroboter

Wie weit das Spektrum möglicher Realisierungen von Servicerobotern reicht, mögen die beiden folgenden Beispiele zeigen: Ein an einem Rollstuhl angebrachter Manipulatorarm zur Unter-

stützung älterer oder behinderter Personen stellt ebenso einen Serviceroboter dar wie etwa der SONY-Hund *Aibo*, sofern man Entertainment auch als Service am Menschen auffasst (Abbildung 1.1).



Abbildung 1.1: Rollstuhl mit Manipulator (MANUS-Arm) und der Roboterhund Aibo von SONY als mögliche Realisierungen von Servicerobotern.

Die soeben erwähnten Beispiele verdeutlichen die Unmöglichkeit, möglichst alle bestehenden Systeme bzw. die entsprechenden Forschungs- und Entwicklungsprojekte zu beleuchten. An dieser Stelle sei auf die entsprechende Literatur verwiesen. Einen guten Überblick zu aktuellen Entwicklungen und Trends findet man beispielsweise in [SS98], [VDI00], [Bis98], [BGG98] oder auch [KPBI96].

Entsprechend der bereits definierten Intention dieser Arbeit sollen vorrangig Systeme und Anwendungsszenarien behandelt werden, die ihre Leistungs- bzw. Tragfähigkeit durch eine (möglichst intuitive und natürliche) Interaktion zwischen Serviceroboter und dem menschlichen Benutzer erlangen.

So müssen Systeme, die perspektivisch in häuslichen Umgebungen zur Unterstützung älterer oder behinderter Menschen eingesetzt werden sollen (Home-Care-Systeme, ausführliche Diskussion dazu in Kapitel 4), über die entsprechenden Interaktionsfähigkeiten verfügen. In dem genannten Anwendungsbereich geht man teilweise sogar soweit und argumentiert, dass solche Roboter neben ihrer Fähigkeit zur natürlichen und intuitiven Interaktion auch einen anthropomorphen (humanoiden) Aufbau aufweisen sollten, der möglichst menschenähnliche Bewegungen zulässt (Abbildung 1.2). Der Grund dafür ist neben der unmittelbaren Interaktion mit Menschen insbesondere darin zu suchen, dass diese Systeme in der gleichen privaten Umgebung operieren und über ähnliche (beispielsweise haptische) Fähigkeiten verfügen müssen wie ihr Interaktionspartner auch.

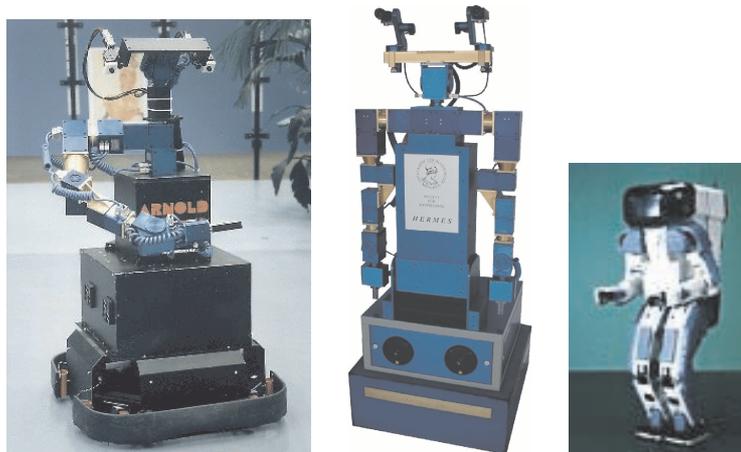


Abbildung 1.2: (v.l.n.r.) Anthropomorphe Roboter ARNOLD am Institut für Neuroinformatik der Ruhr-Universität Bochum ([BBD⁺ 97]), HERMES am Institut für Messtechnik der Universität der Bundeswehr München ([Bis00a]) sowie der Roboter P2 der Firma Honda.

Ein weiteres typisches Anwendungsszenario für Systeme der o.g. Art sind mobile Informationskioske, die in Museen, auf Messen oder in Supermärkten operieren (ausführliche Diskussion in Kapitel 3). Während man bei Servicerobotern zur Unterstützung älterer oder behinderter Menschen zumindest davon ausgehen kann, dass sich der konkrete Interaktionspartner zumindest partiell auf die „Eigenheiten“ seines maschinellen Gefährten einstellt, müssen mobile Informationskioske mit beliebigen Personen intuitiv interagieren können. Hier kann nicht davon ausgegangen werden, dass der Benutzer über Kenntnisse bezüglich der technischen oder informationstechnischen Möglichkeiten seines maschinellen „Gegenüber“ verfügt, was entsprechend bei der Konzeption der Mensch-Roboter-Schnittstelle berücksichtigt werden muss (siehe dazu auch Abschnitt 2.4). Beispiele für diese Art von Serviceroboter-Applikationen sind die drei Infotainment-Roboter des Instituts für Produktionstechnik und Automatisierung der Fraunhofer-Gesellschaft (FhG-IPA) in Stuttgart, die im Foyer des Museums für Kommunikation in Berlin für Information und Unterhaltung sorgen oder der Roboter MINERVA, der im Smithsonian's National Museum of American History in Washington im Rahmen eines Tourguide-Projektes eingesetzt wurde (Abbildung 1.3)

Heute findet man bereits kommerzielle Systeme (siehe Abbildung 1.4), die selbständig Bontendienste in Büroumgebungen oder Krankenhäusern (Helpmate, USA) erledigen, die autonom Supermärkte während der Öffnungszeit reinigen (Hefter, <http://www.hefter.de>) oder in häuslicher Umgebung autonom staubsaugen (Robosoft, Frankreich).

Weitere Beispiele für kommerzielle Serviceroboter-Systeme sind Reinigungssysteme für Flugzeuge, Inspektionsroboter für Unterwasser-Anwendungen und Roboter, die im Baugewerbe eingesetzt werden (siehe auch [VDI00]). Einen aktuellen Überblick über verschiedenste Serviceroboter-Anwendungen gibt ebenfalls die entsprechende Datenbank des Instituts für Produktionstechnik und Automatisierung der Fraunhofer-Gesellschaft (FhG-IPA, [htt01]). Je leistungsfähiger solche Systeme werden, und je mehr diese Systeme Schritt für Schritt in die verschiedensten Bereiche des täglichen Lebens vordringen, umso weniger wird es gelingen, „typische“ Anwendungsfelder zu charakterisieren bzw. zu klassifizieren.



Abbildung 1.3: Infotainment-Roboter des FhG-IPA im Foyer des Museums für Kommunikation in Berlin (links, siehe auch [GBTS00]) und Roboter MINERVA im Smithsonian's National Museum of American History [TBB⁺99a] [TBB⁺00].

1.4 Serviceroboter und Mensch-Maschine-Interaktion

Analysiert man aktuelle Roboter-Anwendungen im allgemeinen, so entsteht das Bild einer Pyramide, deren Basis Industrieroboter bilden und deren Spitze durch (intelligente interaktive) mobile Serviceroboter repräsentiert wird (siehe Abbildung 1.5). Sehr interessant ist in diesem Zusammenhang die Frage, welche Rolle dem Menschen im Sinne der Mensch-Roboter-Interaktion (MRI) auf den verschiedenen Pyramidenstufen zukommt. Erfüllt der Mensch an der Pyramidenbasis ausschließlich die Funktion des Systemingenieurs, eine Funktion, die auf allen Pyramidenebenen anzusiedeln ist, so nimmt seine aktive Rolle als Interaktionspartner nach oben hin kontinuierlich zu. In der Pyramidenmitte übernimmt der Mensch zusätzlich die Funktion des Bedienungspersonals, während Systeme aus der Pyramidenspitze mit beliebigen Personen interagieren können sollen. Damit nimmt einerseits der Grad der Kenntnis der Funktionsweise der Systeme für den eigentlichen Benutzer von unten nach oben kontinuierlich ab, andererseits steigen die Anforderungen an die Fähigkeiten der Systeme, insbesondere hinsichtlich der Mensch-Maschine-Schnittstellen, in derselben Richtung drastisch an. Benötigt der Systemingenieur lediglich ein funktionell hinreichendes Interface, um Programme eines Industrieroboters zu aktualisieren, so steht für den Nutzer eines Serviceroboters die eigentliche Robotertechnologie völlig im Hintergrund. Umschrieben wird dies oft mit dem „Come-as-you-are“-Anspruch, der nichts anderes ausdrückt als die angestrebte Fähigkeit eines Roboters, auch in Interaktion mit völlig uninstruierten Benutzern seine Funktion hinreichend gut zu erfüllen. Dies setzt jedoch völlig neue, natürliche und intuitiv verständliche Formen der Mensch-Roboter-Interaktion voraus (siehe dazu auch Abschnitt 2.4).

Künftige Serviceroboter werden oft in Umgebungen mit geringem Technisierungsgrad eingesetzt werden. Dies bedeutet, dass sie von ungeschultem oder nicht unterwiesenem Personal bedient oder besser benutzt werden. Da dabei der direkte Kontakt zwischen Benutzer und System im Vordergrund steht, wird der Grad der Zufriedenheit der Benutzer neben allen wirtschaftlichen Gesichtspunkten *das* Maß für den Erfolg der jeweiligen Applikation sein. Ausschlaggebend dafür ist, inwieweit das Servicesystem mit seiner Dienstleistung auf den in-



Abbildung 1.4: Beispiele für verschiedene prototypische Serviceroboter. (v.l.n.r.) HelpMate (HelpMate Robotics, Inc. bzw. Pyxis Corporation, USA), Autonomer Reinigungsroboter VARIOTECH der Firma Hefter (Deutschland), autonomer Staubsauger der Firma ROBOSOFT (Frankreich) sowie der Postverteilungsroboter MoPS an der ETH Zürich.

dividuellen Bedarf des Kunden eingehen kann, wie flexibel die Anpassung an die jeweilige Situation und die sich möglicherweise kontinuierlich ändernde Einsatzumgebung erfolgt. Die eben genannten Fähigkeiten eines Serviceroboters bestimmen entscheidend, inwieweit solche Systeme dem Anspruch eines intelligenten Verhaltens gerecht werden können.

1.5 Marktentwicklung

Generell lässt sich während der letzten Dekade eine Entwicklung beobachten, die die gesamte Robotik von der Industrieautomation mehr und mehr in verschiedenste Bereiche des täglichen Lebens führen wird. Im von der United Nations Economical Commission for Europe (EN/ECE) gemeinsam mit der International Federation of Robotics (IFR) erarbeiteten statistischen Bericht World Robotics 2000 [Kar00] wird neben einem weiteren Anstieg im Bereich Industrieroboter eine ähnliche Entwicklung auf dem Sektor der Serviceroboter prognostiziert. Zum Stand 1999 geht man von mindestens 6600 weltweit installierten Servicerobotern aus, die sich zu etwa 50% auf häusliche Anwendungen, 14% auf Unterwasser-Roboter, 12% auf Medizin-Roboter, 6% auf Reinigungsroboter und 23% auf sonstige Anwendungen aufteilen. Für den Zeitraum bis zum Jahr 2003 wird ein weltweiter Anstieg um 49400 Serviceroboter-Systeme vorausgesagt, wobei davon 40000 Systeme auf Anwendungen im häuslichen Umfeld (autonome Staubsaugroboter nicht mit eingeschlossen) sowie 5000 Systeme auf medizinische Anwendungen entfallen. Der gesamte potentielle Markt für den Einsatz von Servicerobotern im häuslichen Bereich wird im Zeitraum bis 2003 mit mehr als 300000 Systemen angegeben, vorausgesetzt, dass diese zu einem entsprechend angemessenen Preis angeboten werden können. Weiterhin wird längerfristig dem Sektor der sogenannten Home-Care-Systeme, also Robotern, die der Unterstützung älterer und hilfsbedürftiger Menschen dienen (siehe auch Kapitel 4), ein enormes Marktpotential beigemessen, insbesondere vor dem Hintergrund der allgemeinen demoskopischen Entwicklung in

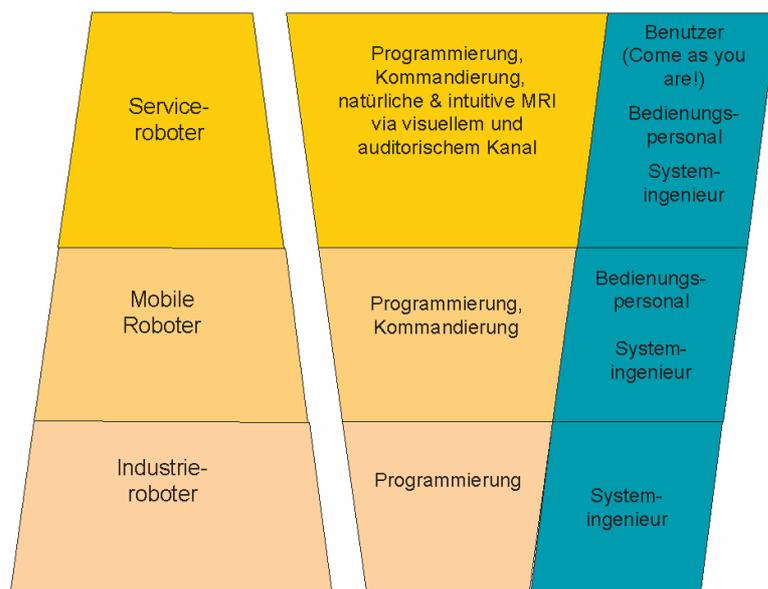


Abbildung 1.5: Wechselspiel zwischen Roboter-Anwendungen und der Rolle des Menschen als Interaktionspartner.

den Industriestaaten.

1.6 Zielstellung der Arbeit

Ausdrücklich betont werden soll an dieser Stelle, dass es *nicht* Anliegen dieser Arbeit ist, visionäre Zukunftsszenarien für Serviceroboter-Applikationen zu entwerfen. Vielmehr soll es darum gehen, anhand der gegenwärtig verfügbaren wissenschaftlich-technischen Methoden laufende bzw. unmittelbar anstehende Entwicklungen aufzuzeigen, zu diskutieren und zu bewerten.

Wie bereits mehrfach betont, wird der Schwerpunkt klar auf Szenarien liegen, in denen der Serviceroboter sehr eng mit seinem menschlichen Interaktionspartner interagiert. Im Rahmen der Arbeit wird stets versucht, die Teilfunktionen, die ein solcher Serviceroboter realisieren muss, im Kontext der angestrebten Servicefunktionalität als eine untrennbare Einheit zu betrachten. Dies rührt aus der Tatsache, dass es für diese Teilfunktionen kaum universelle Lösungen gibt. In diesem Kontext ist ebensowenig davon auszugehen, dass es den universell einsetzbaren Serviceroboter in absehbarer Zeit geben wird. Das folgende Beispiel, das dem eigenen Forschungsprojekt entlehnt ist, soll diese Einheit veranschaulichen: Man stelle sich einen Roboter vor, dessen Serviceleistung u.a. darin besteht, Baumarktkunden zu einem benutzerdefinierten Marktbereich zu lotsen. Für die Erfüllung dieser Aufgabe sind auf der Navigationsseite zunächst eine Reihe von Voraussetzungen notwendig. Der Roboter benötigt Wissen über seine Einsatzumgebung (globale Karte), er muss in der Lage sein, seine aktuelle Position innerhalb dieser Karte zu bestimmen, und er benötigt Mechanismen, die ihm die Planung eines Weges von der aktuellen Position zur benutzerdefinierten Zielposition gestatten. Um mit dem Baumarktkunden in Interaktion treten zu können, benötigt der Serviceroboter entsprechende Ein- und Ausgabegeräte sowie Methoden, um z.B. die Anforderung des Kunden, möglichst schnell zu einem Artikelstandort zu gelangen, in die entsprechenden Steueranweisungen (Auffinden des Artikels in der Umgebungsrepräsentation, daran anschließende Pfadplanung und Fahrt zu diesem Standort) umsetzen zu können. Weiterhin ist die möglichst intuitive und natürliche Inter-

aktionsfähigkeit des Serviceroboters für die Akzeptanz solcher Systeme enorm wichtig. Der Serviceroboter sollte in der Lage sein, eine Person in seiner Einsatzumgebung aktiv wahrzunehmen, von sich aus den Kontakt zu seinem menschlichen Interaktionspartner zu suchen und sich nicht zuletzt intuitiv verständlich machen zu können. All dies sind Voraussetzungen dafür, dass der Serviceroboter nicht Gefahr läuft, das gleiche traurige Dasein zu führen, welches den oftmals anzutreffenden stationären Informationsterminals beschieden ist, die endlos ihre Informationen präsentieren, völlig unabhängig davon, ob sich überhaupt jemand dafür interessiert. Aufgabe der Steuerarchitektur zur Verhaltenskoordination wiederum ist es, u.a. dafür zu sorgen, dass der Kontakt zu dem aktuell eingebuchten Kunden während der Fahrt zu diesem Standort aufrecht erhalten werden kann. Dies bedeutet beispielsweise, dass sich im System je nach aktueller Situation die beiden Modi Personentracking und Zielfahrt wechselseitig beeinflussen müssen, um zu gewährleisten, dass der Kunde dem System bei der Zielfahrt auch wirklich folgt.

Sollen Serviceroboter einen Siegeszug ähnlich dem der Industrieroboter seit den 70er Jahren antreten, so sind dazu eine Vielzahl von technischen und insbesondere informationstechnischen Voraussetzungen notwendig, an denen in verschiedenen Forschungsgruppen und Institutionen mit Hochdruck gearbeitet wird. Neben der Wahrnehmung und Modellierung der Einsatzumgebung betrifft dies insbesondere den Aspekt der Interaktionsfähigkeit: Sollen Serviceroboter gemeinhin als „dienstbare Geister“ eine breite Akzeptanz erfahren, so müssen diese langfristig über dem Menschen adäquate Interaktionsfähigkeiten verfügen. Deshalb wird dem Aspekt der intuitiven Mensch-Roboter-Interaktion eine zentrale Rolle beigemessen, die jedoch stets im Kontext der vorliegenden Aufgabenstellung betrachtet wird und werden muss. Während Industrieroboter in hochtechnisierten Umgebungen zu finden sind, werden Serviceroboter vorrangig dort zum Einsatz kommen, wo ein eher geringer Technisierungsgrad vorherrscht. Dies bedeutet gleichzeitig, dass Serviceroboter mit Personen interagieren, die a-priori nur wenig über die zugrunde liegende Technik / Informationstechnik und die von einem solchen System realistisch zu erwartende Funktionalität wissen. Dennoch soll nicht der Anschein erweckt werden, dass für Serviceroboter-Applikationen generell ausschließlich die Fähigkeit zur natürlichen und intuitiven Interaktion mit Menschen maßgeblich ist. So benötigt ein autonomer Staubsauger sicherlich keine leistungsfähige und intelligente Mensch-Maschine-Schnittstelle. Allerdings lassen sich nur wenige Szenarien absehen, in denen ein Serviceroboter nahezu vollständig autonom agieren kann und, wie im Falle des autonomen Staubsaugers, die Interaktionsfähigkeiten des Systems eine eher untergeordnete Rolle spielen.

Eine tragende These der Arbeit ist, dass die Kenntnis des Roboters über den aktuellen Interaktionsstatus seines menschlichen Partners für eine anzustrebende intuitive Mensch-Roboter-Interaktion (MRI) enorm wichtig ist. Dies soll wiederum bezogen auf das o.g. konkrete Einsatzbeispiel im BauMarkt illustriert werden: Der Roboter verfügt über verschiedene Möglichkeiten, sich seinem menschlichen Gegenüber mitzuteilen. Dies kann über eine Ausgabe auf dem Display, eine Sprachausgabe oder über die Bewegung des Gesichts erfolgen. Um diese Ausgabemodalitäten sinnvoll einsetzen zu können, benötigt der Roboter Kenntnis darüber, ob sich die Person, mit der er gerade interagiert, ihm zuwendet, weil im gegenteiligen Fall eine Ausgabe auf dem Display ebenso wie ein intuitiv verständlicher Gesichtsausdruck sinnlos wären.

Vor diesem Hintergrund werden vor allem im dritten Teil der Arbeit, dessen Gegenstand die eigenen Forschungsarbeiten sind, visuell-basierte Methoden zur MRI motiviert und entwickelt, die einen Beitrag zur intuitiven MRI darstellen. Auch der Einsatz der bidirektionalen

natürlichsprachlichen Interaktion wird diskutiert, auch wenn diese im Kontext der eigenen Forschungsarbeiten noch keine Rolle spielt.

Der Vorteil beim Einsatz visuell-basierter Methoden beschränkt sich jedoch nicht allein auf die MRI. Auch für Navigationsaufgaben können entsprechende Verfahren nutzbringend eingesetzt werden. Im Kontext der eigenen Beiträge betrifft dies insbesondere die robuste Hinderniswahrnehmung und die Zustandsschätzung des Roboters in Bezug auf eine gelernte Karte der Einsatzumgebung.

Weiteres zentrales Anliegen der Arbeit ist es, die Notwendigkeit der parallelen bzw. kombinierten Nutzung verschiedener Sensormodalitäten herauszustellen. Dies betrifft alle Basisfunktionen des Systems und bildet eine wesentliche Voraussetzung, dass der Roboter in seiner typischerweise nicht robotergerecht gestaltbaren Umgebung robust operieren kann.

Abschließend noch der explizite Hinweis, dass die vorliegende Arbeit nicht den Anspruch erhebt, alle Aspekte der intuitiven MRI im Kontext von Servicerobotik-Szenarien zu behandeln. Ebenso soll betont werden, dass es sich bei der Darstellung der eigenen Beiträge zum Wissenschaftsgebiet bei weitem nicht um endgültige Lösungen handelt. Vielmehr wird versucht, vor dem Hintergrund einer konkreten Serviceroboter-Applikation die dabei notwendigen Teilleistungen schrittweise umzusetzen und insbesondere mit den hier vorgestellten Verfahren einen Schritt in Richtung intuitive MRI für Serviceroboter zu erbringen.

1.7 Struktur der Arbeit

Im Kapitel 2 werden elementare Basiskomponenten interaktiver mobiler Serviceroboter als *Navigation, Mensch-Roboter-Interaktion, Steuerarchitektur / Verhaltenskoordination* definiert. Ebenso wird der Frage nachgegangen, wie die angestrebte Servicefunktionalität diese Basiskomponenten determiniert. Ein weiterer Abschnitt widmet sich hier auch der Frage der entsprechenden Sensorik und Aktuatorik.

Einige typische Anwendungsszenarien und Projekte für interaktive mobile Serviceroboter werden im Teil II der Arbeit in den Kapiteln 3 (Tourguide-Roboter) und 4 (Assistenzroboter) behandelt. Hier steht weniger die detaillierte Beschreibung im Vordergrund, sondern es wird versucht, aus einer etwas abstrakteren Sicht grundlegende Zielstellungen und applikationsübergreifende Aspekte herauszuarbeiten (Kapitel 5).

Der dritte Teil der Arbeit ist einem aktuellen Forschungsprojekt am Fachgebiet Neuroinformatik der TU Ilmenau gewidmet. Kapitel 6 gibt einen kurzen Einblick in die Vorgeschichte des BauMarkt-Szenarios und die angestrebte Entwicklung eines intelligenten interaktiven mobilen Shopping-Assistenten. Ausgehend von den Eigenschaften dieser BauMarkt-Umgebung als realem Einsatzszenario werden dann die notwendigen Teilleistungen dieses speziellen Service-systems abgeleitet. Die nachfolgenden Kapitel widmen sich ausführlich der Realisierung dieser Teilleistungen. Kapitel 7 beschreibt zunächst die Navigationsleistungen des Shopping-Assistenten. Die Darstellung der Methoden zur intuitiven MRI findet man in Kapitel 8, bevor in Kapitel 9 die Aspekte der Steuerarchitektur im Kontext der Verhaltenskoordination betrachtet werden. Kapitel 9 verdeutlicht ebenso einen Interaktionszyklus, wie er mit den aktuell realisierten Methoden umgesetzt werden kann. Kapitel 10 diskutiert Stärken und Schwächen des momentanen Realisierungsstandes, gibt einen kurzen Ausblick auf aktuell laufende Forschungsarbeiten

und widmet sich nicht zuletzt den Schlussfolgerungen, die sich aus den bisherigen Erfahrungen und Ergebnissen ergeben. Insbesondere wird hier der Frage nachgegangen, welche Ergebnisse generischer Natur sind und sich möglichst auch auf weitere Anwendungsszenarien übertragen lassen.

Kapitel 2

Basiskomponenten interaktiver Serviceroboter

Aus aktueller Sicht lassen sich eine Reihe konkreter technischer und informationstechnischer Basiskomponenten definieren, die sich im wesentlichen um vier Problemkreise ranken: Navigation, Mensch-Roboter-Interaktion, die Steuerarchitektur zur Koordination des Gesamtverhaltens und mit allem untrennbar verbunden die Umsetzung der applikationsspezifischen Serviceleistungen selbst. In den folgenden Abschnitten werden diese vier Kernpunkte zunächst anwendungsübergreifend betrachtet. Neben der Darstellung der wesentlichen methodischen Aspekte erfolgt auch eine kurze Diskussion der notwendigen Sensorik und Aktuatorik, bevor im anschließenden Teil II dieser Arbeit einige ausgewählte Beispielapplikationen behandelt werden.

2.1 Sensorik und Aktuatorik

Der Begriff des interaktiven mobilen Serviceroboters impliziert unmittelbar die Notwendigkeit von entsprechender Sensorik und Aktuatorik. Das System muss einerseits in der Lage sein, Informationen über sich selbst (propriozeptiv) und seine Umwelt zu akquirieren, andererseits muss es sich in seiner Einsatzumgebung aktiv bewegen und sich gegenüber seinem menschlichen Interaktionspartner (oder weiteren Servicerobotern) mitteilen können.

Wie in Abschnitt 2.6 noch ausführlich diskutiert wird, determiniert das konkrete Anwendungsszenario maßgeblich auch die sensorische und aktuatorische Ausstattung eines Serviceroboters. In diesem Abschnitt sollen jedoch wieder anwendungsübergreifend die typischerweise verwendeten sensorischen und aktuatorischen Systeme kurz vorgestellt werden.

2.1.1 Interne Sensoren

Odometriesensoren, die meist in Form von Inkrementalgebern (optisch, magnetisch, induktiv, kapazitiv etc.) realisiert sind, liefern Informationen über die Bewegung(sänderung) des Roboters (siehe auch [BEF96] und Abschnitt 2.2.4.1). Typischerweise werden Messwerte dieser Sensoren akkumuliert, um die aktuelle Position des Roboters relativ zu seiner Startposition oder die Gesamtbewegung (Trajektorie) eines Manipulatorarms und damit den Zustand des Roboters bezüglich eines globalen Koordinatensystems (z.B. in einer Umgebungskarte) zu ermitteln,

was unter dem Begriff Odometrie bzw. Koppelnavigation zusammengefasst wird. Nahezu jedes mobile Robotersystem verfügt heute über Odometriesensoren, und die eigentliche Odometrie bildet das Rückgrat vieler Navigationssysteme (siehe dazu auch Abschnitt 2.2.4.1).

Für die Bestimmung der Orientierung des Roboters, entweder bezüglich einer Referenzorientierung oder bezüglich des Erdmagnetfeldes, bieten sich Kreisel (Gyroskop) oder Kompass an.

Weitere interne Sensoren liefern z.B. Informationen über die momentane Stellung von Gelenken, kontrollieren den Ladezustand der Batterien, Messen die Stromaufnahme der Motoren oder überwachen die Strombilanz und liefern somit eine Beschreibung des internen Zustands des Roboters.

2.1.2 Externe Sensoren

Abstandsmessende Sensoren Für die Wahrnehmung der Umgebung des Roboters kommen wiederum verschiedene Sensorsysteme in Betracht. Mit Abstand am häufigsten verwendet werden entfernungsmessende Sensoren, die auf der Basis eines vom Roboter selbst ausgesendeten Signals (aktive Sensoren) den Abstand zu Hindernissen in der Senderichtung bestimmen und entweder optisch (Laser, Infrarot) oder akustisch arbeiten (Ultraschall). Die Gründe dafür liegen einerseits, zumindest beim Ultraschallsensor, im günstigen Anschaffungspreis und andererseits in der, zumindest im Vergleich zu Kamerabildern, niedrigen Dimensionalität der anfallenden Daten, die relativ einfach und vor allem effizient verarbeitet werden können. Genauigkeit und Reichweite dieser Sensoren hängen vom physikalischen Wirkprinzip und dem jeweiligen konkreten technischen Aufbau ab (siehe auch [BEF96]). Bezüglich der Anordnung der Sensoren wird immer das Ziel darin bestehen, den unmittelbaren Umgebungsbereich des Roboters, gerade auch im Hinblick auf eine robuste Hindernisvermeidung, möglichst vollständig (dreidimensional) zu erfassen. In jüngster Zeit werden auch verstärkt das Prinzip des GPS (Global Positioning System) für die Roboternavigation eingesetzt. Dabei ist zu beachten, welche Orts- bzw. Positionsaufösung erreicht werden soll, und dass bei Indoor-Anwendungen damit immer die Installation einer entsprechenden technischen Infrastruktur in der Einsatzumgebung verbunden ist.

Taktilen Sensoren, die Kollisionen des Roboters mit Hindernissen detektieren, kommt in erster Linie eine Schutzfunktion zu. Sie lösen meist unmittelbar einen Notstop des Roboters aus, spielen für die eigentliche Funktion des Serviceroboters im Sinne des angestrebten Zurechtfindens in der Einsatzumgebung eine untergeordnete Rolle.

Visuelle Sensoren Während noch vor etwa 5 Jahren mit Kameras ausgestattete Systeme eher die Ausnahme bildeten, gehört heute Bildverarbeitungstechnik fast schon zur Standardausstattung eines Serviceroboters, was zum einen auf den Preisverfall der entsprechenden Komponenten zur Bilderfassung und den dramatischen Anstieg in der Leistungsfähigkeit der informationsverarbeitenden Hardware zurückzuführen ist. Der entscheidende Vorteil der kamerabasierten Umgebungserfassung liegt darin, dass die für verschiedenste Teilaufgaben notwendige Information quasi auf einen Schlag wahrgenommen werden kann. Allerdings gehen mit der jetzt verfügbaren Informationsflut auch Probleme einher, die insbesondere die geeignete und effiziente Extraktion der für die jeweilige Aufgabe relevanten Informationen aus diesem hochdimensionalen Datenstrom betreffen. Da Serviceroboter in ihren Szenarien typischerweise keine sta-

tionären und vorhersagbaren Umgebungsbedingungen (Beleuchtung, Szeneninhalte) vorfinden, spielt dieser Aspekt gerade für die Bildverarbeitung eine wesentliche Rolle. So stellt beispielsweise die robuste Detektion und das Verfolgen von Personen im Kamerabild eines sich selbst bewegenden Roboters unter variablen Umgebungsbedingungen nach wie vor eine wissenschaftliche Herausforderung dar.

Neben der Bedeutung für die Mensch-Roboter-Interaktion spielt der visuelle Kanal zunehmend auch für Navigationsaufgaben eine entscheidende Rolle. So lassen sich beispielsweise Situationen, die rein distanzbasiert nicht voneinander unterschieden werden können, erst durch Zuhilfenahme von visueller Information auseinanderhalten, was eine unter Umständen notwendige Selbstlokalisierung erleichtern kann oder aber überhaupt erst ermöglicht.

Eine visuell-basierte Hindernisvermeidung erweist sich insbesondere dann als notwendig, wenn die gesamte relevante Umgebung des Roboters nicht vollständig mittels abstandsmessender Sensoren abgedeckt wird. Ebenso rückt in Anwendungsszenarien, in denen kaum Annahmen über die geometrischen Gegebenheiten der Einsatzumgebung getroffen werden können, die visuell-basierte Hindernisvermeidung mehr und mehr in den Blickpunkt des Interesses.

Ein Nachteil der visuellen Informationsverarbeitung liegt in der Tatsache, dass aus einem zweidimensionalen Eingangsbild zunächst keine Entfernungsinformation unmittelbar abgeleitet werden kann. Diese kann monokular nur bei Eigenbewegung des Roboters oder aber durch die Verwendung von Mehrkamerasystemen extrahiert werden, wobei beide Fälle auf die Lösung eines Korrespondenzproblems hinauslaufen, dessen robuste Lösung sich unter realen Umgebungsbedingungen als ausgesprochen schwierig erweisen kann (für eine ausführliche Diskussion siehe auch Abschnitt 7.1.3).

Während die Verwendung von (in Fahrtrichtung montierten) Frontalkameras oder kompletten Stereosystemen den klassischen Einsatzfall bildet, finden in jüngster Zeit omnidirektionale Kameras zunehmend Verbreitung. Deren Vorteil besteht in der Möglichkeit, die gesamte Umgebung des Roboters quasi auf einen Blick zu erfassen. Diesen Vorteil erkauft man sich jedoch mit einer typischerweise einhergehenden geringen Detailauflösung, da ein im Vergleich normalen Kamerasystemen größerer Bereich der Umgebung auf die gleiche Sensorfläche abgebildet wird.

Akustische Sensoren Soll der Roboter Geräusche wahrnehmen oder auf sprachliche Äußerungen reagieren können, benötigt er auch dafür entsprechende Sensoren. Hier bietet sich die Verwendung von Mikrofonen an, die entweder auf dem Roboter montiert oder als Funkmikrofone in einem Headset integriert sind, was sich für die Spracherkennung als sehr vorteilhaft erweist, aber auch entsprechende Einschränkungen mit sich bringt (siehe Abschnitt 2.4).

Abbildung 2.1 gibt einen systematischen Überblick über die verschiedenen Sensorsysteme.

2.1.3 Aktuatorik

Die Aktuatorik des Serviceroboters sichert die aktive Bewegung des Systems innerhalb seiner Einsatzumgebung und stellt weiterhin die Artikulation des Roboters sicher. Nur wenn der Roboter in der Lage ist, seinen aktuellen Zustand gegenüber seinem menschlichen Interaktionspartner mitzuteilen, kann sich ein natürlicher und möglichst intuitiver Interaktionszyklus aufbauen.

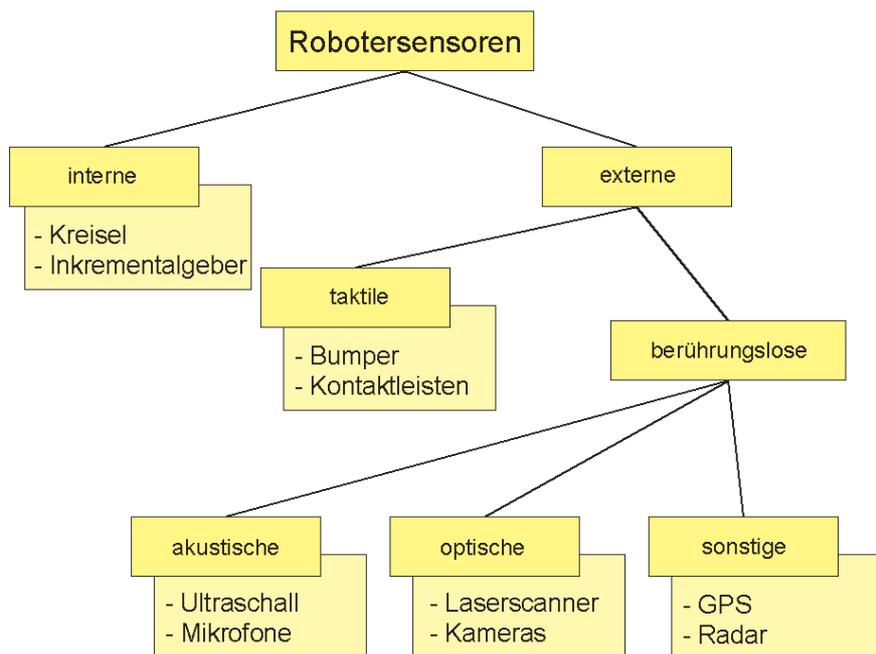


Abbildung 2.1: Schematische Übersicht über verschiedene Sensorsysteme (adaptiert nach [Kni91]).

Antrieb und Manipulation Die Mehrzahl der derzeit bekannten Systeme verfügt über radbasierte Antriebe, die die Fortbewegung auf einem ebenen Untergrund ermöglichen. Abhängig vom Anwendungsszenario werden gegebenenfalls zusätzliche Manipulatoren, die z.B. das Greifen und Übergeben von Objekten realisieren, eingesetzt. Für spezielle Einsatzszenarien wie das Operieren auf unwegsamem Terrain kommen spezielle Antriebskonzepte zum Einsatz, wie z.B. Laufmaschinen.

Artikulation Für die Artikulation, die hier ebenfalls zur Aktuatorik gerechnet werden soll, werden oft Sprachausgabesysteme verwendet, die meist vorgefertigte Sprachausgaben, entweder direkt oder über einen Sprachsynthetisierer, situationsabhängig ausgeben. Weiterhin kann die Artikulation des Roboters über Ausgaben auf einem grafischen Display erfolgen, was allerdings voraussetzt, dass der Benutzer diese von seiner aktuellen Position aus auch einsehen kann.

Zunehmend wird auch der Aspekt einer „robotischen“ Körpersprache bedeutsam, gerade wenn es um eine möglichst natürliche und intuitiv verständliche Mensch-Roboter-Artikulation geht. Dazu verfügt der Serviceroboter dann meist über ein ansteuerbares Gesicht (siehe MINERVA in Abschnitt 3.2 oder auch PERSES in Abschnitt 8.6) oder es wird versucht, über typische Bewegungen den aktuellen Status des Roboters zu transportieren, wobei hier der gesamte Roboter als Ausgabegerät angesehen werden kann.

Abbildung 2.2 fasst die verschiedenen Aktorsysteme überblicksartig zusammen.

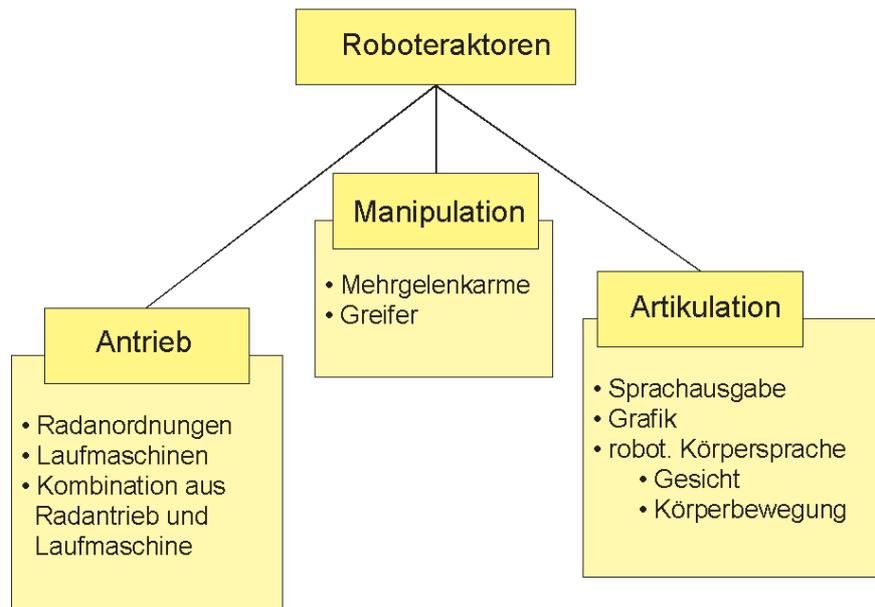


Abbildung 2.2: Schematische Darstellung verschiedener Aktorsysteme, über die ein intelligenter interaktiver mobiler Serviceroboter je nach Anwendungsszenario verfügt.

2.2 Navigation

2.2.1 Grundproblem und Begriffsdefinition

Für jedes mobile Servicesystem ist eine sichere und robuste Navigation im Einsatzfeld unerlässlich. Der Begriff *Navigation* leitet sich aus dem Lateinischen ab und hat seinen Ursprung in der Seefahrt. *Navis* bedeutet *Schiff*, *agere* bedeutet *lenken*. Er beschreibt den Prozess, ein Schiff zu einem Zielpunkt zu führen, der sich aus drei Schritten zusammensetzt, die wiederholt ausgeführt werden [SD91]: zunächst bestimmt der Navigator die Position des Schiffes in einer Karte; danach wird die aktuelle Schiffsposition zur Zielposition, Referenzpunkten und möglichen Gefahrenstellen in Relation gesetzt; abschließend wird vom Navigator der neue Kurs festgelegt. In die Robotik wurde diese Form des Navigationsbegriffs nahezu unverändert übernommen. Die typischen Problemstellungen während der Navigation eines Roboters sind demnach [LL90]:

- Wo befindet sich der Roboter gerade?
- Wo befinden sich andere Objekte oder Orte in der Karte in Relation zum Roboter?
- Wie kommt der Roboter von seiner aktuellen Position zu anderen Orten der Karte?

Algorithmen zur Lösung dieser drei genannten Problemstellungen bilden den Kern vieler Navigationssysteme in der Robotik (siehe z.B. [KBM98] [Thr99]). Einen exzellenten Überblick über bestehende Systeme und Techniken zur Roboternavigation gibt BORENSTEIN [BEF96].

Was bedeutet nun die Lösung von Navigationsaufgaben für einen intelligenten interaktiven Serviceroboter? Aus pragmatischen Gründen soll unter dem Oberbegriff Navigation auch die Hindernis- oder Kollisionsvermeidung behandelt werden, obwohl Navigation im eigentlichen Sinne immer eine Zielposition voraussetzt. Die robuste Vermeidung von Hindernissen bildet jedoch eine elementare Voraussetzung für jede Art von Navigation.

Alle Verfahren zum Erreichen eines unmittelbar detektierbaren Zieles werden unter dem Begriff *lokale Navigation* zusammengefasst. Fasst man die Hindernisvermeidung als das Ziel des Roboters auf, immer den größten detektierbaren Freiraum zu befahren, so kann diese ebenfalls als lokale Navigationsleistung bezeichnet werden, die implizit die Zielposition bzw. -richtung beeinflusst. Verfahren zur lokalen Navigation benötigen weder die Kenntnis der eigenen Position noch eine explizite Modellierung der Einsatzumgebung, da Orte oder Objekte außerhalb des aktuellen sensorischen Horizonts nicht intern repräsentiert werden müssen. Allerdings sind auf dieser Basis lediglich solch elementare Verhaltensweisen wie das zufällige Umherfahren realisierbar. Die *globale Navigation* beinhaltet die Methoden, die dem System die Bewegung von seiner aktuellen Position zu einem beliebigen Ort innerhalb der Einsatzumgebung gestatten. Im Gegensatz zur lokalen Navigation spielt hier neben der Kenntnis der aktuellen eigenen Position die Frage der (meist expliziten) Umgebungsmodellierung in Verbindung mit effizienten Algorithmen zur Pfadplanung eine zentrale Rolle. Eine interne Repräsentation dieser Orte und der Beziehungen zwischen diesen erlaubt es dem Roboter, auch solche Orte anzufahren, die sich außerhalb des aktuellen sensorischen Horizonts befinden.

Während lokale Navigationsverfahren in Verbindung mit robuster Hindernisvermeidung für jedes mobile Servicesystem realisiert werden müssen, hängt die Notwendigkeit globaler Navigationsleistungen sowie die Art und Weise deren Umsetzung von der konkreten Applikation ab.

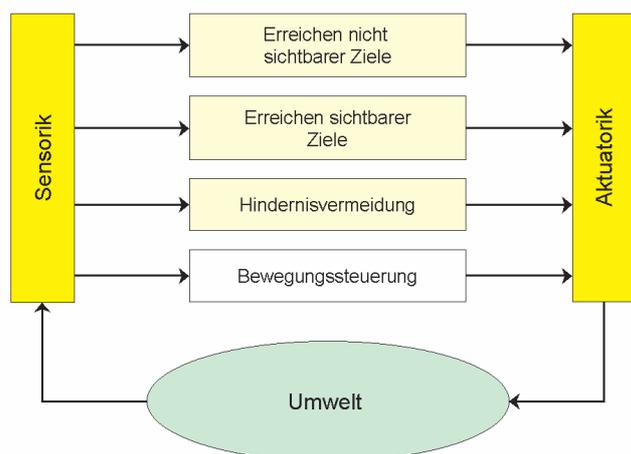


Abbildung 2.3: Mögliche Hierarchie an Navigationsleistungen aus einer verhaltensorientierten Sicht (siehe auch Abbildung 2.8). Die Bewegungssteuerung stellt im eigentlichen Sinne dabei keine Verhaltensleistung dar.

Folgt man einer verhaltensorientierten Sichtweise (siehe auch Abschnitt 2.3), so ergibt sich die in Abbildung 2.3 dargestellte Hierarchie an Navigationsverhalten, in der lokale Navigationsverfahren die unterste Stufe darstellen. Alle hierarchisch darüber angesiedelten Teilleistungen sind dann entsprechend der globalen Navigation zuzuordnen.

2.2.2 Lokale Navigationsverfahren und Hindernisvermeidung

Aufgabe der lokalen Navigation im Sinne der Hindernisvermeidung ist es, dem Roboter eine kollisionsfreie Fahrt zu ermöglichen. Dabei müssen sowohl Ungenauigkeiten der Bewegungsplanung bzw. -steuerung als auch dynamische, unvorhergesehen auftretende Hindernisse berücksichtigt werden. Dabei sollte der Roboter in einem sicheren Zustand bleiben, also grundsätzlich auf einer Bahnkurve gehalten werden, die physikalisch und kinematisch realisiert werden kann. Zur Hindernisvermeidung wurden verschiedenste Ansätze vorgeschlagen, die nahezu ausschließlich als *reaktive* Verfahren (im Gegensatz zu planenden Verfahren) bezeichnet werden können. Maßgeblich mitgeprägt wurde die Entwicklung hier von einer Reihe so genannter biomimetischer Navigationsalgorithmen [FM00], die versuchen, zur Lösung von Navigationsaufgaben Anleihen bei biologischen Systemen zu nehmen. Biomimetische Ansätze betrachten im Gegensatz zur klassischen Roboternavigation Navigation als Verhaltensleistung eines biologischen Systems. Für eine Reihe von Navigationsleistungen eines biologischen Systems spielt z.B. die Frage der aktuellen Position innerhalb eines globalen Weltmodells nicht die zentrale Rolle; vielmehr geht es um die Frage, wie ein bestimmtes Ziel erreicht oder ein Gesamtverhalten realisiert werden kann. So findet man in [FM00] die folgende Definition: „Navigation is the process of determining and maintaining a course or trajectory to a goal location.“ Einige dieser stark biologisch orientierten Ansätze werden bereits praktisch eingesetzt. So wurde das in [Sri92] experimentell untersuchte und neuronal modellierte Verfahren zur visuellen Navigation der Biene in [SCW⁺99] und [BTN00] zur Roboternavigation implementiert. Allerdings sind die biomimetischen Ansätze zur Roboternavigation noch weit davon entfernt, die mittlerweile eher als klassisch anzusehenden Ansätze in vollem Umfang zu ersetzen. Der Vielzahl an experimentellen Arbeiten steht derzeit eine noch vergleichsweise geringe Zahl an praktischen Implementierungen gegenüber. Dennoch liefern die biomimetischen Navigationsverfahren wertvolle Ideen, die in praktische Robotikszenerarien mit einbezogen werden können und sollten.

2.2.3 Verfahren zur globalen Navigation

Die globale Navigation mobiler Roboter wurde in den zurückliegenden Dekaden intensiv bearbeitet. Reflektiert man die historische Entwicklung (siehe auch [Bro99]), so erfolgte eine ähnliche Evolution wie sie auch im Abschnitt 2.3 bezüglich der Problematik Steuerarchitektur und Verhaltenskoordination skizziert wird.

Etwa Mitte der 70er Jahre entstanden Verfahren, die ausschließlich auf der Verwendung interner Sensorik (Odometriesensoren) basierten und exakte, vollständige (vom Designer entworfene) Umgebungsmodelle voraussetzten. Aufgrund der notwendigen exakten Modelle war deren Funktionalität jedoch nur für extrem stark eingeschränkte Umgebungen gegeben. Infolgedessen wurden ab Mitte der 80er Jahre unter dem Schlagwort *verhaltensorientierte Robotik* [Bro86] so genannte reaktive Verfahren entwickelt, die die Notwendigkeit expliziter Modelle negierten. Allerdings konnte der Nachweis der Funktionalität nur für Navigationsleistungen im Low-Level-Bereich (siehe Abbildung 2.3 und Abschnitt 2.2.2), wie beispielsweise die Hindernisvermeidung, erbracht werden. Dies wiederum leitete die Entwicklung hybrider Ansätze ein, die vornehmlich seit den 90er Jahren Einzug in die Robotik hielten und bis heute auch den größten Teil der Navigationskonzepte ausmachen (siehe auch Abschnitt 2.3). Parallel dazu werden seit Mitte der 90er Jahre verstärkt probabilistische Verfahren in die Robotik eingeführt (siehe

Abschnitt 2.2.4.2), deren wesentlichstes Merkmal die explizite Einbeziehung aller Arten von Unsicherheit, die in Realwelt-Szenarien auftreten können, ist.

Stellvertretend soll hier lediglich auf zwei grundlegende Verfahren eingegangen werden. Während die Koppelnavigation (Abschnitt 2.2.4.1) ausschließlich auf der Auswertung der internen Odometriesensoren beruht, stellen die in Abschnitt 2.2.4.2 skizzierten probabilistischen Verfahren quasi den Kontrapunkt dazu dar.

Abbildung 2.4 gibt einen groben Überblick über die entsprechende historische Entwicklung der verschiedenen Verfahren.

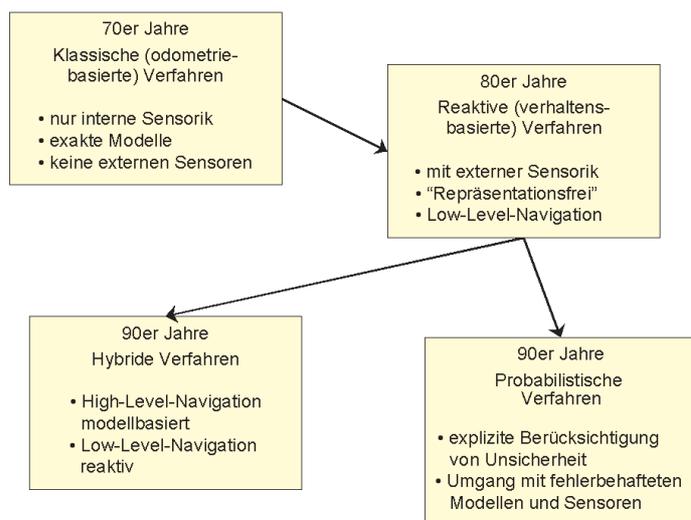


Abbildung 2.4: Historie der Entwicklung verschiedener Verfahren zur globalen Roboternavigation.

Da für alle globalen Navigationsverfahren ein Modell der Umgebung, in der der Roboter operiert, unumgänglich ist, soll im nachfolgenden Abschnitt zunächst auf einige typische Formen der Umgebungsrepräsentation eingegangen werden.

2.2.4 Umgebungsmodelle

Umgebungsmodelle stellen quasi das Weltwissen des Roboters dar. Für einen autonomen Serviceroboter ist weiterhin typisch, dass dieses Weltwissen nicht a priori zur Verfügung gestellt (z.B. in Form eines Bauplanes), sondern vom Roboter selbstständig erworben wird (eine ausführliche Diskussion dazu findet man in Kapitel 7).

Zur Kategorisierung der verschiedenen Modellierungsvarianten kommen verschiedene Kriterien in Betracht. Auch bestimmen die Art der vom Serviceroboter zu lösenden Aufgaben (siehe Abschnitt 2.6), die Charakteristika der verwendeten Sensoren und die Randbedingungen der Umwelt maßgeblich die Art des Umgebungsmodells.

Unterteilt man die verschiedenen Modelle nach ihrem Abstraktionsgrad, so ergibt sich die in Abbildung 2.5 dargestellte Hierarchie (siehe auch [Edl97]).

Das metrische (oder auch rasterorientierte) Umweltmodell (siehe auch Abschnitt 7.2, [Nil69] [Elf89]) basiert auf einer zweidimensionalen Gitterstruktur, welche wahlweise mit fester Gittergröße oder auch hierarchisch (z.B. als Quadtree) organisiert werden kann. Im einfachsten Fall wird die Umwelt auf ein Gitternetz projiziert und für jede Zelle des Gitters vermerkt, ob

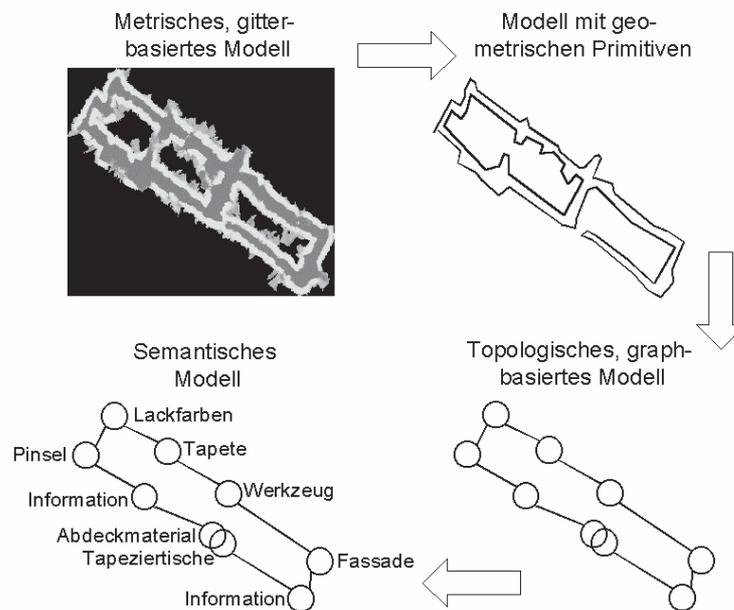


Abbildung 2.5: Typische Umgebungsmodelle, deren Abstraktionsgrad in der dargestellten Hierarchie in Pfeilrichtung zunimmt. Das Beispiel wurde stark vereinfacht und dem im Teil III dieser Arbeit beschriebenen Forschungsprojekt, einem interaktiven mobilen Shopping-Assistenten für den Einsatz im BauMarkt, entlehnt.

sich dort ein Hindernis befindet oder nicht. Dabei sind auch Zwischenwerte in Form von Belegungswahrscheinlichkeiten denkbar. Vorteil einer solchen Umweltrepräsentation ist die maßstabgetreue Abbildung der Umwelt und die einfache Handhabung dieser Repräsentation. Ein entscheidender Nachteil ist, dass eine genaue Abbildung der Umwelt meist nicht ohne Informationsverlust möglich ist, abgesehen von Umgebungen, in denen nur rechte Winkel vorkommen und Hindernisse, deren Grenzen genau auf Rasterkanten fallen. Weiterhin gestaltet sich der Aufbau solcher metrischer Karten aufgrund von Fehlern in der Positionsbestimmung des Roboters als sehr fehleranfällig. Weitere Nachteile sind der hohe Speicherverbrauch und der geringe Abstraktionsgrad sowie daraus resultierende Probleme bei der Pfadplanung und Selbstlokalisierung.

Eine abstraktere und u.U. besser an die Umweltstruktur angepasste Modellierung kann mit der Verwendung von geometrischen Primitiven erreicht werden [IAY89]. Abhängig vom Abstraktionsgrad können Linien, Polygone, Flächen oder auch Körper verwendet werden. Für ein solches Umweltmodell ist es allerdings notwendig, dass die Umwelt eine weitgehend geradlinige Struktur aufweist. Dies ist im Indoor-Bereich (z.B. für Büroumgebungen) oft gegeben, trifft aber für den Outdoor-Bereich nur bedingt zu. Dieses Modell hat gegenüber dem zuvor besprochenen rasterorientierten Modell den Vorteil, dass zumindest im Indoor-Bereich eine recht genaue Abbildung der Umwelt möglich ist. Der Nachteil dieses Modells liegt in der Schwierigkeit, die geometrischen Primitive aus den Sensordaten zu extrahieren. Problematisch sind dabei fehlerbehaftete oder zu ungenaue Messungen der Sensorik, die eine entsprechende Vorverarbei-

tung erfordern.

Beim topologischen Umweltmodell [KB88] [NS90] verzichtet man auf eine maßstabsgetreue Abbildung der Umgebung und wählt eine abstraktere Repräsentation. So wird die Umwelt in Form eines Graphen repräsentiert. Die Knoten eines solchen Graphen stellen dabei so genannte Landmarken bzw. Plätze dar. Verbindungen zwischen diesen Landmarken werden durch die Kanten des Graphen repräsentiert. Topologische Umweltmodelle enthalten (meist) keine Informationen über Entfernungen. Dadurch haben sie den Vorteil, relativ robust gegenüber Fehlern in der Positionsbestimmung des Roboters beim Kartenaufbau und bei der Selbstlokalisierung zu sein. Weitere Vorteile sind der geringe Speicherplatzbedarf und die effizientere Pfadplanung. Ein großer Nachteil ist die aufwendige Generierung eines topologischen Umweltmodells. Eine Möglichkeit ein topologisches Modell zu erhalten ist, die Freiräume der Umgebung in konvexe Polygone zu unterteilen und in den Knoten abzuspeichern. Verbindungen zwischen den einzelnen Polygonen würden dann als Kanten gespeichert.

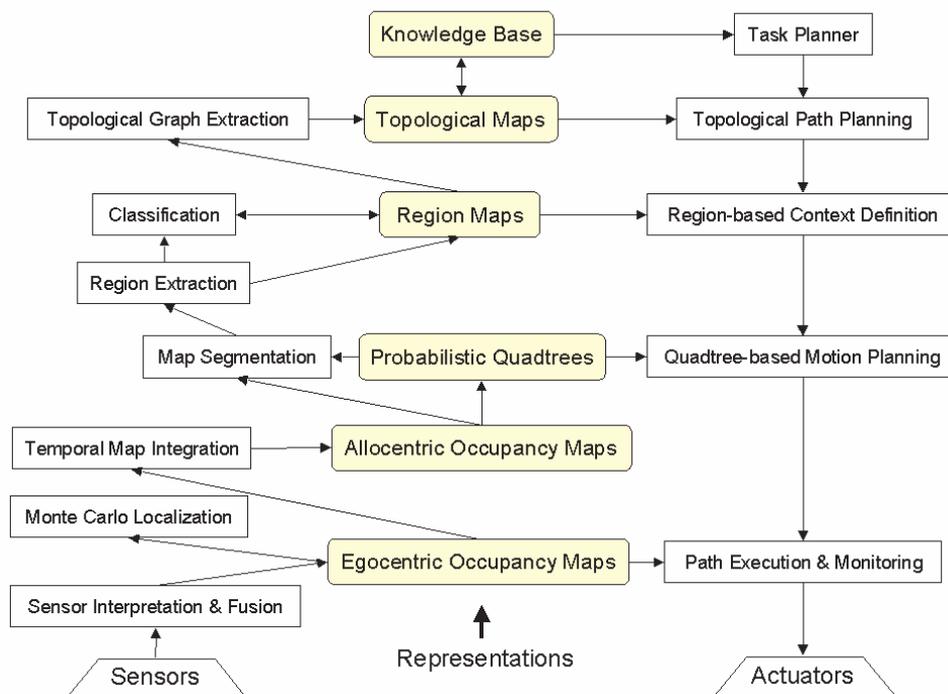


Abbildung 2.6: Typisches Beispiel für die Anwendung einer hierarchischen Umgebungsrepräsentation (nach [KSE⁺00]). Dargestellt sind links sind die Verarbeitungsstufen, die die jeweiligen Repräsentationen erzeugen, sowie rechts die Verarbeitungsstufen, die auf den jeweiligen Repräsentationen entsprechende Planungsverfahren realisieren.

Das semantische Umweltmodell stellt die höchste Stufe der Abstraktion dar. Hierbei wird Objekt (Hindernis) oder Orten der Umgebung eine Bedeutung (Semantik) zugeordnet. Verbindungen zwischen diesen Objekten können durch einfache Beziehungen, wie z.B. „zwischen Raum A und Raum B gibt es eine Verbindung“, repräsentiert werden. Für den praktischen Einsatz kann es ebenfalls sinnvoll sein, ein metrisches oder topologisches Umweltmodell mit semantischen Informationen zu attributieren, worauf im Abschnitt 7.4 bei der Darstellung der

eigenen Forschungsarbeiten noch ausführlicher eingegangen wird. Ein solches Modell vereinfacht die Formulierung von Aufträgen in Bezug auf die Mensch-Maschine-Schnittstelle. Dadurch sind z.B. Anweisungen wie „Fahre zu den Lackfarben“ direkt interpretierbar. Weiterhin bietet ein solches Modell den Vorteil, dass mit der Semantik verbundene Informationen, wie die Dynamik von Objekten (z.B. Türen oder Fahrstühlen) bei der Aktionsplanung berücksichtigt werden können. Eine autonome Erstellung eines solchen Modells durch den Roboter ist mit dem heutigen Stand der Technik jedoch noch außerordentlich schwierig, weshalb semantische Umgebungsmodelle typischerweise mit menschlicher Hilfe aufgebaut werden.

Wie bereits herausgearbeitet wurde, kann und wird es *das* Umweltmodell nicht geben. Hinzu kommt, dass nicht selten mehrere der eben diskutierten Repräsentationen in einer mehrstufigen (hierarchischen) Darstellung zum Tragen kommen, was eine eindeutige Zuordnung zu einem bestimmten Konzept unmöglich macht. Im Kontext hierarchischer Umgebungsmodelle soll stellvertretend der von KRAETZSCHMAR in [KSE⁺00] vorgestellte Ansatz genannt werden, dessen Struktur in Abbildung 2.6 illustriert ist.

2.2.4.1 Koppelnavigation

Die auch unter den Begriffen *dead reckoning* bzw. Odometrie bekannte Koppelnavigation ist den bereits in den 70er Jahren entwickelten Navigationsverfahren (siehe Abbildung 2.4) zuzuordnen und stellt das mit Abstand am häufigsten verwendete, wenn auch nie das alleinige Navigationsverfahren dar. Bekanntermaßen liefert die Odometrie über kurze Wegstrecken meist eine gute Genauigkeit und gestattet eine hohe Abtastrate. Das Grundprinzip besteht letztlich jedoch in der Integration der Bewegungsinformation über einen längeren Zeitraum, was bei alleiniger Verwendung dieses Verfahrens zwangsläufig zu großen Fehlern in der Positionsschätzung führt. Um diesen Nachteil zumindest zu mildern, kommen verschiedene Verfahren zur Fehlerreduktion zum Einsatz (insbesondere zur Reduktion des systematischen Odometriefehlers).

Obwohl bei der Koppelnavigation erhebliche Fehler auftreten können und man die durch die Odometriesensoren gelieferte Information (zumindest theoretisch) auch ausschließlich mittels externer Sensoren durch die Beobachtung der Umgebung gewinnen kann, ist derzeit kein Servicerobotik-Szenario bekannt, welches explizit auf Odometriesensoren verzichtet. Der Grund dafür liegt in der Tatsache, dass die mittels Odometriesensoren gewonnene Information, und sei diese noch so fehlerbehaftet, zumindest in erster Näherung gute Schätzwerte für den aktuellen Zustand des Robotersystems (Position und Orientierung in Bezug zu einem globalen Weltmodell) liefert, was die Ermittlung des tatsächlichen (wahrscheinlichsten) Zustands mittels externer Sensorik wenn auch nicht überflüssig macht, so aber zumindest erheblich beschleunigt. Da auch die externe Sensorik typischerweise fehlerbehaftet ist und hier Störungen auftreten können, die beispielsweise die sichere Wahrnehmung von Landmarken für einen bestimmten Zeitraum unmöglich machen, kann zumindest kurzzeitig ausschließlich mittels Odometrie navigiert werden. Weiterhin besteht die Möglichkeit, Odometriefehler anhand extern gewonnener sensorischer Information zu korrigieren, wie dies z.B. in dem in Abschnitt 7.2.1 beschriebenen Verfahren, das den eigenen Forschungsarbeiten entstammt, der Fall ist. Für detailliertere Ausführungen zum Thema odometriebasierte Navigation sei auf [BEF96] verwiesen.

2.2.4.2 Probabilistische Verfahren

Seit etwa Mitte der 90er Jahre halten verstärkt probabilistische Modellierungsverfahren Einzug in die Robotik [Thr00a]. Wie bereits betont besteht deren wesentliches Merkmal in der expliziten Berücksichtigung aller Formen von Unsicherheit, die für Robotikanwendungen in realen, nicht oder nur teilweise vorhersehbaren Umgebungen typisch sind.

Die theoretischen Wurzeln dieses Modellierungsrahmens gehen bis in die 60er Jahre zurück und basieren maßgeblich auf dem Bayes'schen Theorem und der Repräsentation von Information in Form von Wahrscheinlichkeitsdichtefunktionen. THRUN charakterisiert den Kerngedanken der probabilistischen Verfahren in der Robotik folgendermaßen [Thr00a]:

- **Probabilistische Wahrnehmung:** Roboter sind prinzipiell unsicher bezüglich des Zustandes ihrer Umgebung. Diese Unsicherheit rührt von limitierten Sensoren, Rauschen und der Tatsache her, dass reale Umgebungen, zumindest bis zu einem gewissen Grad, nicht vorhersehbar sind. Durch die Repräsentation der Sensorinformation als Wahrscheinlichkeitsverteilung über alle möglichen (mit diesem Sensor erfassbaren) Zustände der Welt wird die alleinige Verwendung einer momentan besten (wahrscheinlichsten) Sensorinterpretation vermieden. Dies führt letztlich dazu, dass der Roboter robust mit temporären Fehlern und Mehrdeutigkeiten umgehen und Sensordaten konsistent integrieren kann. Der Roboter „weiß“ sozusagen um die Unzulänglichkeit seiner eigenen Wahrnehmung.
- **Probabilistische Steuerung:** Autonome Roboter müssen unter Unsicherheit agieren, eine Tatsache, die sich direkt aus der unsicheren Wahrnehmung ableitet. Eine probabilistische Aktionsauswahl berücksichtigt diese Unsicherheit. Einige Verfahren betrachten lediglich die aktuelle Unsicherheit, andere antizipieren in Richtung zukünftiger erwarteter Unsicherheit. Anstatt lediglich die wahrscheinlichste Situation (für die Aktionsauswahl) zu betrachten, versuchen viele probabilistische Verfahren, ein entscheidungstheoretisches Optimum zu berechnen, in dem alle möglichen Alternativen Berücksichtigung finden.

Mittlerweile existieren eine Reihe von Algorithmen (z.B. Methoden zur adaptiven und spärlichen Abtastung komplexer Dichtefunktionen wie die Monte-Carlo-Lokalisation, siehe dazu auch [FBT99] [FDBT99]), die eine effiziente und akkurate Umsetzung dieser probabilistischen Verfahren ermöglichen, so dass deren ursprüngliches Defizit, die enorme Berechnungskomplexität, mehr und mehr in den Hintergrund tritt.

Der Nachweis der robusten Funktionalität probabilistischer Verfahren zur Robotersteuerung konnte im Rahmen der Tourguide-Projekte RHINO und MINERVA [BCF⁺98] [FBT98] [FBT99] [TBB⁺98] [TBB⁺99b] (siehe auch Abschnitt 3.2) erbracht werden.

2.2.5 Pfadplanung und Bewegungssteuerung

Letztlich muss der Roboter in die Lage versetzt werden, seine Bewegung so zu steuern, dass ein vorgegebenes Ziel unter Kollisionsvermeidung erreicht werden kann. An dieser Stelle fließen die Aspekte der lokalen ebenso wie die der globalen Navigation zusammen, da für die Bewegungssteuerung sowohl die momentane lokale Hindernissituation als auch die vom globalen Planer vorgegebene Richtung zum Ziel berücksichtigt werden müssen.

Die Pfadplanung lässt sich auf das Problem der Suche nach einem kürzesten Weg in einem Graphen abbilden, wobei die Struktur des Graphen vom verwendeten Umgebungsmodell abhängt. Einen guten Überblick über verschiedene Algorithmen zur Pfadplanung findet man in [BKOS97], wobei der DIJKSTRA-Algorithmus das bekannteste Verfahren darstellt. Abhängig vom konkreten Anwendungsfall, der vorliegenden Roboterkinematik und der Charakteristik der Einsatzumgebung lassen sich weitere Randbedingungen formulieren, die während der Pfadplanung als Nebenbedingungen mit berücksichtigt werden können.

Ein typisches Verfahren zur Bewegungssteuerung stellt die Potentialfeld-Methode dar [BK89], die den Roboter näher zum (globalen) Ziel bringt und dabei Kollisionen mit Hindernissen vermeidet. Kraftvektoren repräsentieren dabei sowohl die Richtung zum Ziel als auch im sensorischen Erfassungsbereich befindliche Hindernisse, wobei von Hindernisse abstoßende Kräfte ausgehen, während die Richtung zum Ziel auf den Roboter als Anziehungskraft wirkt. Eine Erweiterung des Potentialfeld-Ansatzes bildet die Verwendung sogenannter Vektorfeld-Histogramme, die ebenfalls in [BK89] beschrieben wird und einige Nachteile der Potentialfeld-Methode umgeht.

Auf alle genannten Verfahren wird im Rahmen der in Teil III beschriebenen eigenen Forschungsarbeiten (Abschnitt 7.4) noch ausführlicher eingegangen.

2.3 Steuerarchitektur und Verhaltenskoordination

Unter Steuerarchitektur und Verhaltenskoordination werden all die Methoden zusammengefasst, die für das korrekte Wechselspiel aller auf dem Serviceroboter implementierten Teilleistungen oder Teilverhalten verantwortlich sind. Mit zunehmender Komplexität des Gesamtsystems kommt der Steuerarchitektur und der darin realisierten Verhaltenskoordination eine immer bedeutendere Rolle zu.

2.3.1 Sense-Plan-Act-Paradigma

Betrachtet man die historische Entwicklung (siehe dazu insbesondere [Ark98] und [KBM98]), so begann diese mit dem in Abbildung 2.7 skizzierten pipelineorientierten Sense-Plan-Act-Paradigma, das eine universelle Wahrnehmung der Welt annimmt, die Wahrnehmung vollständig von der Verhaltensgenerierung trennt und den Schwerpunkt ganz klar auf den Planungsaspekt legt.

Betont werden sollte die diesem Ansatz innewohnende universelle Repräsentation der Welt, die damit ausschließlich sensorisch geprägt ist (siehe dazu auch [Moe96]), Aspekte einer aktiven, handlungsgetriebenen Wahrnehmung vollständig vernachlässigt und sich stark an das Informationsverarbeitungsparadigma nach MARR [Mar82] mit seiner Trennung von Wahrnehmung und Verhaltensgenerierung anlehnte. Neben der Annahme dieser universellen Repräsentation der Welt weist die kettenförmige Abfolge der Verarbeitungsstufen in Form einer horizontal organisierten Architektur weitere gravierende Nachteile auf:

- Die Performanz des Gesamtsystems kann nur so gut sein wie das schwächste Glied innerhalb einer solchen Kette.

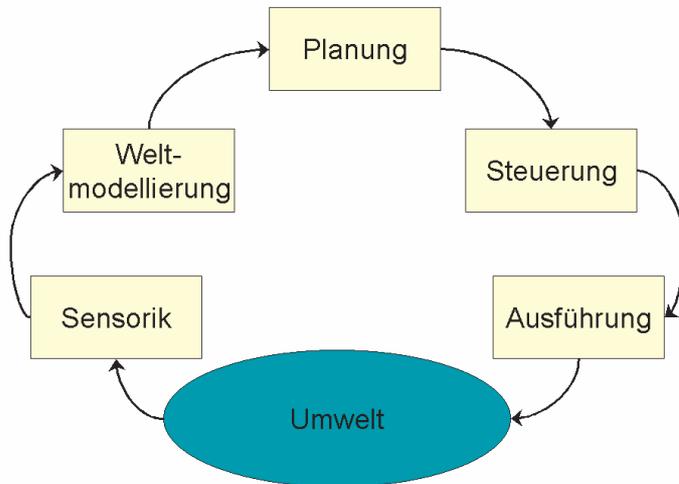


Abbildung 2.7: Sense-Plan-Act-Paradigma.

- Das Gesamtsystem kann seine Funktion erst dann aufnehmen, wenn alle Kettenglieder fertiggestellt sind.
- Die Architektur ist nur schwer erweiterbar, da neue Teilkomponenten in den gesamten informationsverarbeitenden Ablauf eingebunden werden müssen.

2.3.2 Subsumption-Architektur

Aufgrund der genannten Schwachstellen des Sense-Plan-Act-Paradigmas wurden verschiedene Architekturen vorgeschlagen, die eine Dekomposition in verschiedene Verhaltensmodule vornehmen, diese parallel und teilweise konkurrierend zueinander implementieren, und deren Wechselwirkung durch eine Kontrollstruktur übernommen wird, die den Zugriff der verschiedenen Verhaltensmodule auf die Aktorseite steuert. Das wohl bekannteste Beispiel ist die in Abbildung 2.8 dargestellte Subsumption-Architektur von BROOKS [Bro86], die die Verwendung von internen Repräsentationen (zumindest solche auf symbolischem, expliziten Niveau) komplett ablehnt und die Welt als deren beste Repräsentation auffasst.

Genau genommen ist die im Vergleich zum horizontal organisierten Sense-Plan-Act-Paradigma vertikal organisierte Subsumption-Architektur jedoch nicht wirklich repräsentationsfrei, sondern es existieren sensorische (implizite) Repräsentationen innerhalb der verschiedenen Verhaltensmodule. Im Gegensatz zum Sense-Plan-Act-Paradigma liegt hier eine Repräsentation der Welt verteilt in den verschiedenen Verhaltensmodulen vor. Charakteristisch ist hier weiterhin die angestrebte enge und möglichst unmittelbare Kopplung zwischen Wahrnehmung und Aktion, was zur Bezeichnung als reaktive Systeme führte (im Gegensatz zu planenden Systemen).

Neben der Umorientierung hinsichtlich der Steuerarchitekturen vollzog sich parallel ein Paradigmenwechsel bezüglich der Wahrnehmung im allgemeinen (siehe auch [Bro99] [Moe96]), der weg von der universellen Wahrnehmung hin zu einer zweckbestimmten, aktionsorientierten Wahrnehmung führte. Das jeweilige Verhaltensziel bestimmt, welcher Aspekt der Umwelt wahrgenommen bzw. modelliert wird. Dieser Paradigmenwechsel wurde maßgeblich beein-

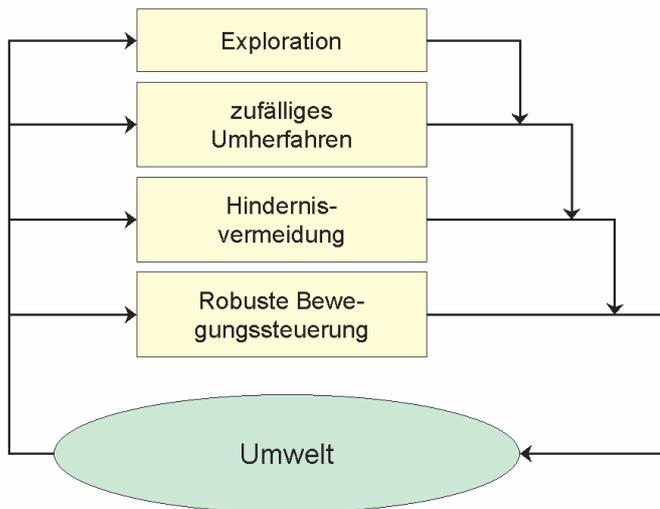


Abbildung 2.8: Subsumption-Architektur.

flusst durch die Forschungen auf den Gebieten der Ethologie und Psychophysik. Als typische Vertreter sind die Arbeiten im Bereich des sogenannten Active-Vision (siehe auch [AWB87] [Baj88] [Bal91] [MS99]) anzusehen, die enorme praktische Konsequenzen für den Umgang mit sensorischen Datenströmen mit sich brachten und letztlich auch das Paradigma der selektiven Aufmerksamkeit definierten, in dem der Datenstrom hinsichtlich der Informationen untersucht wird, die für die aktuell zu lösende Aufgabe relevant sind, was einer sehr engen Verkopplung zwischen Wahrnehmung und Aktionsgenerierung gleichkommt. Damit einher geht eine enorme Reduktion der Dimensionalität des Eingaberaums, was wiederum zu einer Effizienzsteigerung der eigentlichen Informationsverarbeitung führt.

2.3.3 Dreiebenen-Architektur

Nachdem die Subsumption-Architektur den radikalen Bruch mit dem klassischen, pipelineorientierten Informationsverarbeitungsparadigma vollzogen hatte, fand diese zunächst viele Anhänger, die versuchten, diese Art der Sensor-Aktor-Kopplung für verschiedene Robotikszenerien umzusetzen. Dabei stellte sich jedoch heraus, dass der völlige Verzicht auf jegliche Arten interner Repräsentationen eine Reihe von Nachteilen mit sich bringt: so bedeutet dies, dass keinerlei explizite interne Planung möglich ist, da diese notwendigerweise ein wie auch immer geartetes internes Modell der externen Welt erfordert.

Dies führte letztlich zur Entwicklung einer Architekturfamilie, deren bekanntester Vertreter die so genannte 3T-Architektur [BKW97] ist. Dieses wiederum vertikal organisierte Architekturkonzept (eine sehr gute Charakterisierung findet man in [KBM98], Kapitel 8), dessen prinzipielle Struktur Abbildung 2.9 zeigt, trägt der Notwendigkeit von reaktiven und planenden Komponenten innerhalb einer globalen Steuerarchitektur Rechnung. Während schnelle reaktive Komponenten (Skills) mit ihren typischen unmittelbaren Sensor-Aktor-Kopplungen ohne Zuhilfenahme interner Zustandsinformation auf der unteren Hierarchieebene (Skill Management) zu finden sind, stellt die mittlere Ebene (Sequencer), deren Algorithmen von interner Zustandsinformation abhängen, die aber selbst noch keine Planung im Sinne von Suchverfahren realisiert, quasi eine Vermittlungsfunktion zwischen reaktiver unterer Ebene und planender

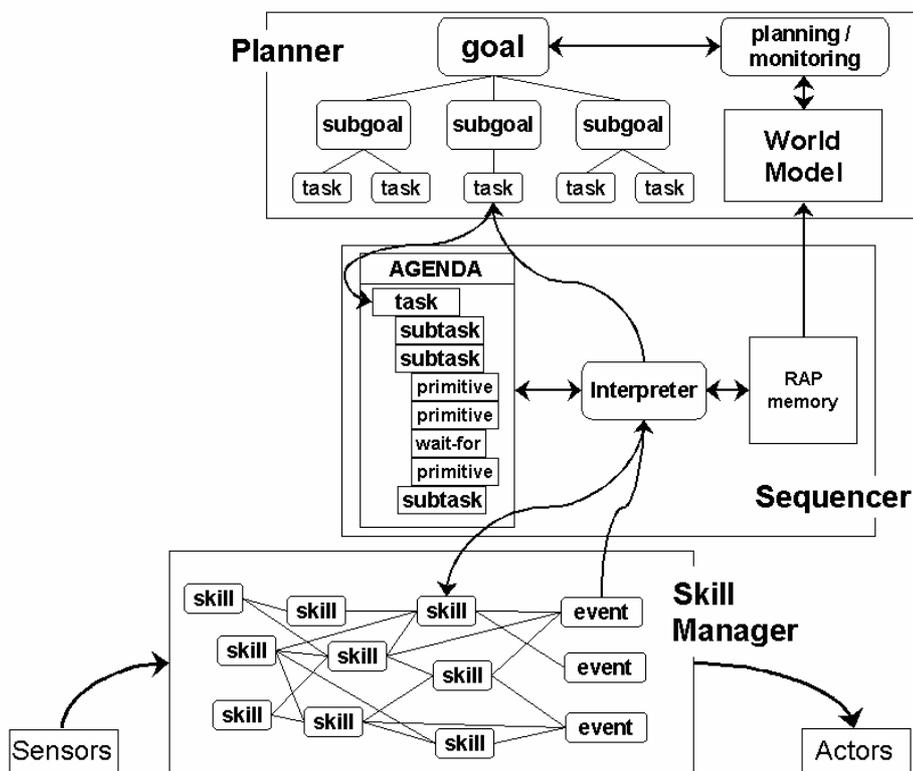


Abbildung 2.9: Schematischer Aufbau der 3T-Architektur (Anlehnung an [BKW97]).

oberer Hierarchieebene dar. Sie transformiert die vom Planer kommenden Informationen in sogenannte *Reactive Action Packages* (RAP's), die von der Ebene des Skill Management umgesetzt werden können. Die letztgenannte Ebene (Planner), die wiederum interne Zustandsinformation nutzt, beinhaltet alle berechnungs- und damit zeitintensiven Methoden wie beispielsweise Planungs- und Suchverfahren.

2.3.4 Verhaltensbasierte Architekturen

Verhaltensbasierte Steuerarchitekturen, wie sie beispielsweise bei ARKIN [Ark98] umfassend beschrieben und diskutiert werden, wurden sehr stark durch die Forschungen auf den Gebieten Ethologie und Psychophysik geprägt. Sie sind nach [Ark98] u.a. gekennzeichnet durch eine enge Sensor-Aktor-Kopplung, die Vermeidung expliziter symbolischer Repräsentationen, eine Dekomposition in sinnvolle Verhaltenseinheiten (behaviors) und den verwendeten Koordinationsmechanismus (konkurrierend versus kooperierend). Aus einer eher ingenieurwissenschaftlichen Sicht lassen sich diese Architekturen als eine mögliche Ausprägung der unteren beiden Ebenen der 3T-Architektur auffassen.

2.3.5 Hybride Steuerarchitekturen

Analysiert man die Vielfalt der heute vorgeschlagenen Steuerarchitekturen, so stellen diese meist eine Mixtur aus den bislang behandelten prinzipiellen Konzepten unter Berücksichtigung der realen Gegebenheiten der jeweiligen Applikation dar, da mittlerweile Einklang darüber herrscht, dass es *die* ultimative Steuerarchitektur nicht gibt und in absehbarer Zeit wohl auch nicht geben wird. Die zur Verhaltenskoordination entwickelten Verfahren tragen aus heutiger Sicht im wesentlichen zwei grundlegenden Aspekten Rechnung: Zum einen versucht man, der Komplexität des Gesamtsystems durch geeignete Zerlegung (Dekomposition) des Gesamtverhaltens des Systems in sinnvoll abgrenzbare Teilverhalten und deren Integration in eine problemangepasste Steuerarchitektur zu begegnen. Zum anderen wird zunehmend versucht, die auftretenden Unsicherheiten sowohl beim Verhalten des Roboters selbst als auch hinsichtlich der Umgebungseigenschaften mit geeigneten Modellierungsparadigmen zu berücksichtigen, da bei realen Systemen in realen Umgebungen meist nur in sehr begrenztem Maße deterministisches Verhalten vorausgesetzt werden kann.

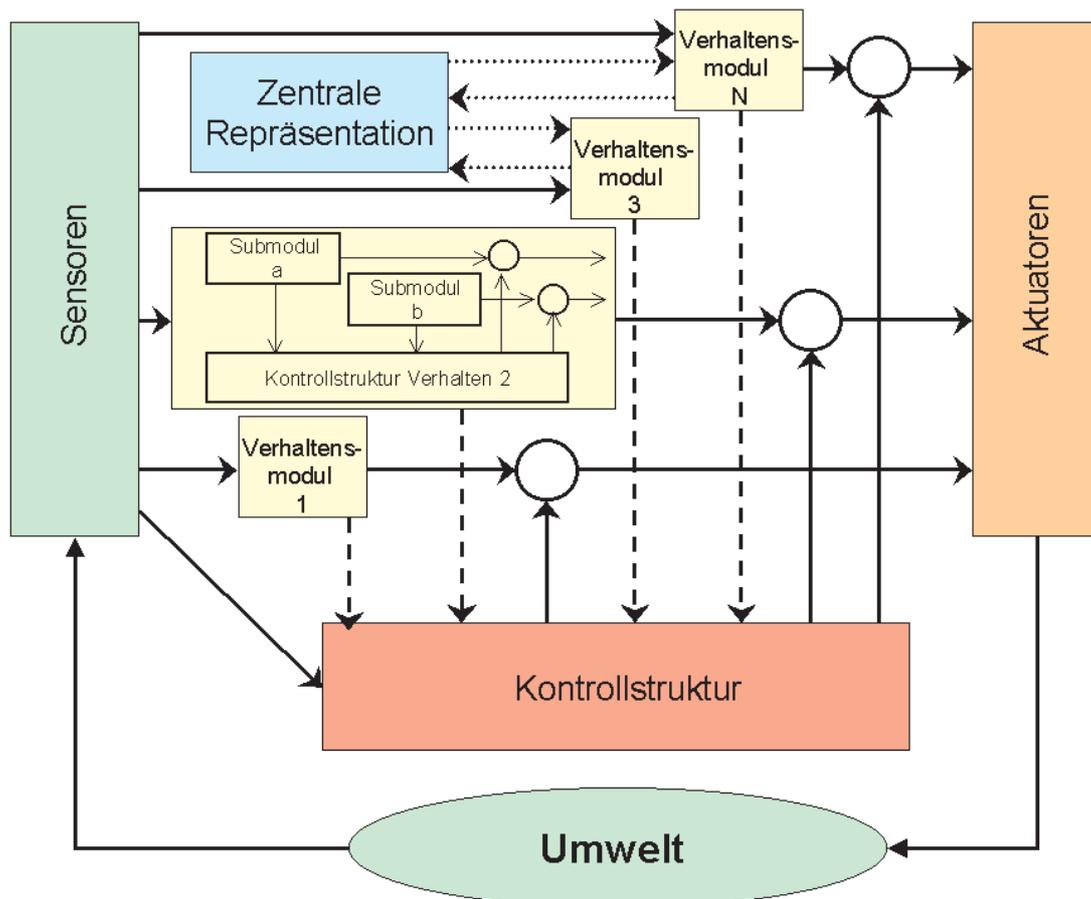


Abbildung 2.10: Eine möglichst universell gehaltene, hybride Steuerarchitektur.

Mit der Abbildung 2.10 wurde der Versuch unternommen, eine möglichst allgemein gehaltene, verhaltensbasierte Steuerarchitektur wiederzugeben, wie sie in den meisten aktuellen Forschungsprojekten im Bereich Servicerobotik vorzufinden ist. Um der Komplexität des Ge-

samtsystems Rechnung zu tragen, erfolgt typischerweise eine Zerlegung in elementare Verhaltensweisen, die aus dem konkreten Einsatzszenario abgeleitet und zunächst relativ unabhängig voneinander implementiert werden können. Typisch ist weiterhin die enge Sensor-Aktor-Kopplung verschiedener Verhaltensmodule, wobei auch Teilverhalten definiert werden können, die nicht direkt auf die Aktuatorik des Systems zugreifen, sondern lediglich eine Zustandsänderung des Systems bewirken (z.B. Verhalten 3 in Abbildung 2.10). Eine Kontrollstruktur übernimmt dann die Koordination der Teilverhalten und entscheidet, welches Teilmodul Zugriff auf die Aktorik des Systems erhält. Die Kontrollstruktur (siehe z.B. Abschnitt 9.1) erhält sowohl sensorische Information als auch Statusmeldungen von den zu koordinierenden Verhaltensmodulen. Eine hierarchische Steuerung liegt dann vor, wenn auch innerhalb der einzelnen Verhaltensmodule bereits Kontrollmechanismen zum Einsatz kommen, wie dies bei Verhalten 2 in Abbildung 2.10 angedeutet wurde. Weiterhin zu nennen sind die typischerweise auftretenden unterschiedlichen Zeitskalen, auf denen die einzelnen Verhaltensmodule operieren, sowie die verteilte Repräsentation des Weltwissens in Form aufgabenspezifischer Repräsentationen in den einzelnen Modulen. Ebenso ist es jedoch u.U. sinnvoll, zentrale Repräsentationen vorzusehen, die von verschiedenen Verhaltensmodulen generiert bzw. von diesen genutzt werden (z.B. globale Umgebungskarte). Der Vorteil der Gliederung in verschiedene Verhaltensmodule ist, dass die Architektur sehr flexibel und vor allem leicht erweiterbar ist. Letzteres ist insbesondere dann gegeben, wenn die Teilverhalten ausschließlich über eine zentrale Kontrollstruktur koordiniert werden und keine Wechselwirkungen zwischen den Teilverhalten vorliegen. Weiterhin bleibt das Gesamtsystem damit auch beim Ausfall einzelner Verhaltensmodule immer noch, wenn auch eingeschränkt, funktionsfähig.

Zusammenfassend soll eine Aussage von ERANN GAT (zitiert aus [KBM98], Kapitel 8) wiedergegeben werden, die sich zwar eigentlich auf die 3T-Architektur bzw. deren Spielarten bezieht, aber dennoch die heutige Sicht auf das Problem der Steuerarchitekturen sehr gut widerspiegelt:

In retrospect, in the story of the three-layer architecture there may be more to be learned about research methodology than about robot control architectures. For many years, the field was bogged down in the assumption that planning was sufficient for generating intelligent behavior in situated agents. That it is not sufficient clearly does not justify the conclusion that planning is therefore unnecessary. A lot of effort has been spent defending both of these extreme positions. Some of this passion may be the result of a hidden conviction on the part of AI researchers that at the root of intelligence lies a single, simple, elegant mechanism. But if, as seems likely, there is no One True Architecture, and intelligence relies on a hodgepodge of techniques, then the three-layer architecture offers itself as a way to help organize the mess.

2.4 Mensch-Roboter-Interaktion (MRI)

Wie im Kapitel 1 bereits herausgearbeitet wurde, spielt die Frage der Mensch-Roboter-Interaktion für zukünftige erfolgreiche Applikationen von Servicerobotern eine wesentliche, wenn nicht *die zentrale* Rolle. Dies trifft insbesondere dann zu, wenn der Serviceroboter in einem

relativ gering technisierten Bereich eingesetzt werden und mit Nutzern interagieren soll, die uneingewiesen sind und keine Kenntnisse bezüglich Robotertechnologien besitzen, wobei die notwendige Interaktionsfähigkeit natürlich auch von der konkret zu erbringenden Serviceleistung des Roboters abhängt. Für einen Reinigungsroboter, der einen Supermarkt während der Öffnungszeiten reinigen soll, stellen Menschen in erster Linie dynamische Hindernisse dar, die es zu umfahren gilt. Demgegenüber muss ein System, das als mobiler Informationskiosk in der gleichen Einsatzumgebung operieren soll, über völlig andere Möglichkeiten der Mensch-Roboter-Interaktion verfügen, da von dieser Interaktion maßgeblich die Erfüllung der Serviceaufgabe abhängt.

In diesem Abschnitt soll der Frage nachgegangen werden, welche Formen der Mensch-Roboter-Interaktion prinzipiell möglich sind, welche Methoden für deren Umsetzung zur Verfügung stehen und wie weit eine praktische Realisierung für reale Einsatzszenarien bislang gediehen ist. Dabei steht die unmittelbare MRI entsprechend der Intention dieser Arbeit im Vordergrund, weshalb Mensch-Roboter-Schnittstellen zur telebasierten (Fern)Steuerung (siehe z.B. [Fis99]) und Web-basierte Interfaces (siehe z.B. [SBC00]) hier nur benannt, aber nicht näher betrachtet werden sollen.

2.4.1 Klassische MRI

Im einfachsten Fall können bisherige Ein-/Ausgabegeräte wie Tastatur, Maus und Bildschirm (Touchscreen) verwendet werden, um eine eher „klassische“ Art der Interaktion zu realisieren. Der Vorteil dieser Herangehensweise liegt darin, dass der Entwickler keine zusätzlichen Komponenten berücksichtigen muss und die Interaktion den Menschen, die den Umgang mit Computern gewohnt sind, nicht sonderlich schwer fallen dürfte. Für die Interaktion mit Personen, die über wenig oder keine Erfahrung im Umgang mit technischem Gerät verfügen, wird sich diese Form der Interaktion als nachteilig erweisen. Wer möchte sich schon blamieren, wenn er bei der Bedienung eines Informationsterminals beobachtet wird. Dennoch muss zum gegenwärtigen Zeitpunkt festgestellt werden, dass sich diese klassischen Eingabegeräte auf absehbare Zeit nicht vollständig ersetzen lassen werden. Es wäre zwar durchaus sinnvoll, ausschließlich die in der zwischenmenschlichen Interaktion entscheidenden visuellen und akustischen Kanäle in adäquater Weise zu nutzen, aber weder der aktuelle Entwicklungsstand zur Erkennung frei gesprochener Sprache noch die derzeit verfügbaren Methoden zur Erkennung von Gestik, Mimik und Körpersprache lassen dies derzeit zu (siehe dazu auch den folgenden Abschnitt).

2.4.2 Intuitive, natürliche MRI

Sehrwohl sinnvoll erscheint jedoch eine Erweiterung der klassischen Interaktionsformen um solche Komponenten, die die gesamte Interaktion möglichst intuitiv und natürlich machen. Die angestrebte Intuitivität und Natürlichkeit impliziert nahezu zwangsläufig die Einbeziehung des visuellen und des akustischen Kommunikationskanals, da diese auch für die zwischenmenschliche Kommunikation wesentlich sind.

2.4.2.1 Sprachbasierte Interaktion

Menschen kommunizieren in erster Linie über Sprache. Der entscheidende Vorteil beim Einsatz eines sprachbasierten Dialogs liegt entsprechend in der damit erreichbaren Natürlichkeit der Interaktion.

Analysiert man den aktuellen Entwicklungsstand solcher Systeme, stellt man fest, dass diese vorrangig für bestimmte Domänen, wie z.B. für die automatische Fahrplan- oder Programmauskunft via Telefon, eingesetzt werden. Ein Beispiel für ein solches System zeigt Abbildung 2.11.

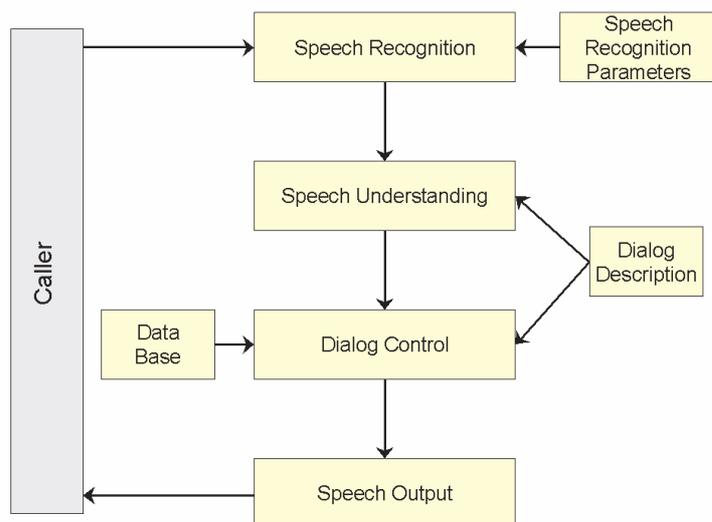


Abbildung 2.11: Architektur des Philips-Dialogsystems (nach [AN98]).

Wenn es beim Einsatz eines Serviceroboters gelingt, im Kontext der vorliegenden Anwendung bestimmte Domänen zu definieren, können auch sprachbasierte Dialogsysteme in die MRI sehr sinnvoll integriert werden. So wurden in den vergangenen Jahren verschiedene Systeme vorgestellt, die eine natürlichsprachliche Kommandierung von Robotern zum Ziel haben [LLH⁺94] [AFD⁺99] [GHN⁺00] [GS01] [CHS00] [BG99a] [RPT00]. Allen genannten Anwendungen ist gemein, dass die potentiellen Nutzer des Serviceroboters Kenntnis über das (typischerweise stark eingeschränkte) Vokabular haben müssen, welches der Roboter zu erkennen imstande ist, und welche Aktionen seitens des Roboters damit verbunden sind. Dies ist beispielsweise bei Roboter-Applikationen mit einem begrenzten und relativ konstanten Personenkreis oder in Personal-Robot-Anwendungen im häuslichen Bereich der Fall, wie z.B. in den in Kapitel 4 behandelten Nursebot-Szenarien.

Soll ein Serviceroboter jedoch mit beliebigen Personen interagieren können, treten hinsichtlich eines sprachbasierten Dialogs zunächst folgende Probleme auf: Es kann in diesem Fall nicht mehr davon ausgegangen werden, dass der Benutzer weiß, was der Roboter zu leisten vermag, welchen Zweck er erfüllen soll und wie er mit dem Roboter umzugehen hat. Der Benutzer hat typischerweise keinerlei Kenntnis darüber, a) welche Funktion der Roboter überhaupt erfüllen soll, b) ob der Roboter die menschliche Sprache versteht, c) ob er selbst vom Roboter wahrgenommen wird oder d) der Roboter nur einem starr vorgegebenen Ablauf folgt. Umgekehrt ist es für den Serviceroboter a priori ebenso unmöglich, Annahmen über typische Spracheingaben im Sinne einer Erwartungshaltung zu treffen. Dies bildet jedoch die Voraussetzung dafür, dass das zur Verfügung stehende Vokabular entweder a priori begrenzt gehalten werden oder zumindest der Suchbereich für den Erkenner bei sehr großem Vokabular eingeschränkt werden

kann, um eine effiziente Spracherkennung und kurze Reaktionszeiten zu gewährleisten. Erschwerend kommt hinzu, dass die Verwendung eines Headsets von vornherein ausscheidet, was zusätzliche Anforderungen an die Robustheit des Spracherkennungssystems impliziert. Letztlich hängt aber auch hier der sinnvolle Einsatz eines sprachbasierten Dialogs von der Definition der o.g. Domänen ab. So kann beispielsweise die Phase des ersten Kontakts als Begrüßungsdomäne angesehen werden, in der sich der Roboter selbst vorstellt und gezielt seine möglichen Leistungen offeriert. Damit erhält der menschliche Interaktionspartner quasi Kenntnis von den Möglichkeiten des Roboters und kann sich dann seinerseits darauf einstellen. Weiterhin kann seitens des Roboters während der Dialogführung versucht werden, den Menschen zu bestimmten Äußerungen zu führen, im Sinne von *Bitte nennen Sie eines der vorgeschlagenen Exponate!* im Falle eines Tourguide-Roboters oder auch *Soll ich Sie jetzt dorthin führen? Bitte antworten Sie mit Ja oder Nein!*. Ein Beispiel, in dem die eben skizzierte Herangehensweise erfolgreich für eine Serviceroboter-Applikation umgesetzt werden konnte, stellt der Roboter MOBSY dar, der am Lehrstuhl für Mustererkennung der Universität Erlangen als Empfangssystem für Besucher zum Einsatz kommt [ZDH⁺01].

Ein Nachteil der sprachbasierten Dialogführung liegt sicher in der Tatsache begründet, dass der Roboter deutlich unpräzisere Information verarbeiten muss, als das z.B. bei der Interaktion über einen Touchscreen oder eine Tastatur der Fall ist. Diesem Umstand muss entsprechend bei der Modellierung des Dialogs Rechnung getragen werden, wie dies z.B. in [RPT00] vorgeschlagen wird.

Letztlich muss im Kontext der vorliegenden Aufgabenstellung analysiert werden, an welcher Stelle eine bidirektionale sprachbasierte MRI sinnvoll eingesetzt werden kann. Eine vollständig sprachbasierte MRI wird jedoch in den seltensten Fällen möglich sein, weshalb die o.g. klassischen Interaktionsformen insbesondere bei Servicerobotern, die mit uneingewiesenen Nutzern interagieren sollen, sicherlich auch weiterhin eine zentrale Rolle spielen.

2.4.2.2 Visuell-basierte Interaktion

Verfahren zur aktiven und gezielten visuellen Wahrnehmung von Personen für eine möglichst natürliche und intuitive Mensch-Maschine-Kommunikation werden seit etwa 10 Jahren von einer Vielzahl von Forschungseinrichtungen intensiv bearbeitet. Eine Vorreiterrolle kommt hier dem MIT zu [RP98] [Wre00] [DB98], und die dort entwickelten Konzepte der „smart rooms“ (Räume, die ihre Insassen beobachten und mit diesen multimedial interagieren können) bzw. „smart desks“ bieten eine Reihe interessanter Ansatzpunkte, die ebenso für viele Anwendungsszenarien im Serviceroboterbereich relevant sind. Der Vorteil visuell-basierter Verfahren liegt unumstritten in der Tatsache, dass sich in erster Linie mit deren Hilfe hinreichend Informationen über den menschlichen Interaktionspartner gewinnen lassen, die mit anderen Sensormodalitäten nur schwer oder gar nicht extrahierbar sind, die aber die Voraussetzungen für eine angestrebte natürliche und intuitive MRI bilden. So sollte das Servicesystem in der Lage sein, seine Einsatzumgebung aktiv nach potentiellen Interaktionspartnern abzusuchen, anstatt passiv auf eine Eingabe zu warten. Weiterhin sollte das System wahrnehmen können, in welchem „Interaktionsstatus“ sich der menschliche Gegenüber gerade befindet (z.B. ob sich die Person dem System zuwendet oder nicht). Kann die erstgenannte Aufgabe u.U. auch auf abschließlicher Basis eines Laserscanners gelöst werden, trifft dies für den zweiten genannten Aspekt nicht mehr zu. Ebenso sollte der Kontakt zu einem Interaktionspartner möglichst konti-

nüerlich aufrecht erhalten werden können. Auch hier kann man sinnvoll entfernungsmessende Sensoren einsetzen, Informationen über den Interaktionsstatus oder auch darüber, ob es sich bei zwischenzeitlichem Kontaktverlust noch um die gleiche Person handelt, liefern diese Sensoren jedoch nicht.

Nachfolgend sind einige wesentliche Problemkreise genannt, für die visuelle Verfahren entscheidende Beiträge im Kontext intuitiver MRI leisten können.

Personenlokalisierung Zunächst steht das Problem der Lokalisation und des Verfolgens von Personen. Soll der „smart room“ auf seine Insassen oder der Serviceroboter auf seine potentiellen Benutzer reagieren können, so muss er wissen, wo sich diese befinden, und er muss in der Lage sein, diese Personen gegebenenfalls kontinuierlich zu verfolgen. Das wohl bekannteste System für Personenlokalisierung und Personentracking stellt der so genannte *Pfinder* (person finder, [WADP97] [Wre00]) dar, dessen grundlegende Funktionen sich in vielen Lokalisations- und Tracking-Systemen anderer Arbeitsgruppen wiederfinden. Der Pfinder verwendet ein statistisches Mehrklassenmodell, mit dem personentypische Farbe (Haut) und Formen für Personenbestandteile repräsentiert werden. Ausgehend von diesem Modell wird versucht, die Bildkoordinaten zu finden, an denen das Vorhandensein einer Person am besten verifiziert werden kann. Das Modell selbst besteht im Merkmalsraum aus einer Menge von Hyperellipsen, die räumlich untereinander verbunden sind, so dass durch Resynthese mittels dieser Hyperellipsen die gesamte erfasste Haltung der Person sichtbar gemacht werden kann. Nachteil des Systems ist, dass indirekt von einem im Vergleich zur Personenregion deutlich weniger dynamischen Hintergrund ausgegangen wird, was die Segmentierung der Person stark vereinfacht. Weiterhin wird eine stationäre, ruhende Kamera angenommen, was beispielsweise bei mobilen Servicerobotern nicht oder nur eingeschränkt vorausgesetzt werden kann. Andere Arbeiten lokalisieren Personen mittels einer Detektion von Gesichtern [TDA98] [SEN98] [RBK98], eine Vorgehensweise, die dann gerechtfertigt ist, wenn davon ausgegangen werden kann, dass die erforderlichen Gesichtsdetails immer erfasst werden können. Kann von einer ruhenden Kamera ausgegangen werden, so stellen auch bewegungsbasierte Verfahren eine geeignete Möglichkeit dar, Personen zu lokalisieren [RS98] [LS98] [Dav96] [DB97]. Während Hautfarbe, Gesichtsstruktur und Bewegung fast als klassische Merkmale zur Personenlokalisierung angesehen werden können, nutzen OREN und Mitarbeiter im Gegensatz dazu typische Konturverläufe, um Fußgänger in einer Videosequenz zu detektieren. Alle genannten Verfahren weisen Vor- und Nachteile auf. Hautfarbe lässt sich sehr einfach und damit schnell detektieren, ändert sich jedoch unter variablen Beleuchtungsbedingungen und kommt auch in unbelebten Objekten vor. Das Gesicht besticht durch seine hohe Spezifität, lässt sich jedoch nur bei entsprechender Ausleuchtung und Bildauflösung robust detektieren. Konturbasierte Verfahren bieten ebenfalls eine gute Spezifik, allerdings ist hier die Robustheit der Detektion sehr stark abhängig vom die Person umgebenden Szeneninhalte. Bewegungsbasierte Verfahren sind wenig berechnungskomplex, setzen jedoch eine ruhende Kamera voraus (zumindest dann, wenn eine aufwendige Eigenbewegungskompensation vermieden werden soll) und liefern keine hohe Spezifität für Personen.

Um im Rahmen einer Serviceroboter-Applikation die erforderliche Robustheit einer angestrebten Personenlokalisierung zu erreichen, wird deshalb auch versucht, mehrere Verfahren zu kombinieren und dabei möglichst die Schwächen eines Verfahrens durch die Stärken eines anderen Verfahrens zu kompensieren. So schlagen FEYRER und ZELL einen integrierten Ansatz zur Personenlokalisierung vor [FZ98] [FZ99a], der mehrere personentypische Merkmale (Haut-

farbe, Bewegung und Kontur) kombiniert und in [FZ00] durch die Hinzunahme eines Laserscanners zu einem multimodalen Ansatz erweitert wurde. Ein ähnlicher Ansatz wurde auch in den eigenen Forschungsarbeiten, die im Teil III ausführlich beschrieben sind, zunächst für die Personenlokalisierung [BBB⁺97] [BBB⁺98a] [BBCG99] [CBG00] und im weiteren Verlauf zur Personenverifikation [BG99b] [BWK⁺01a] (siehe auch Abschnitt 8.2), entwickelt.

Personentracking Die angestrebte Funktionalität des Serviceroboters entscheidet darüber, inwieweit Verfahren zum kontinuierlichen Kontakthalten zu einer Person erforderlich sind oder nicht (siehe auch Abschnitt 2.6). Dass ein autonomer Reinigungsroboter nicht über diese Fähigkeit verfügen muss, leuchtet sofort ein. Auch Systeme, die für Botendienste eingesetzt werden, sind nicht auf das kontinuierliche Kontakthalten zu einer bestimmten Person angewiesen, sondern müssen lediglich bei Erhalt und nach Beendigung eines Auftrages mit den entsprechenden Personen kommunizieren. Anders stellt sich die Situation dar, wenn für die Erfüllung der Serviceaufgabe diese Fähigkeit unbedingt notwendig oder zumindest stark wünschenswert ist, beispielsweise für mobile Informationskioske (siehe Teil III dieser Arbeit) oder auch für Tourguide-Systeme, die in Museen oder auf Messen eingesetzt werden sollen (siehe auch Kapitel 3).

Systeme zum Tracking von Objekten können in zwei elementare Bestandteile untergliedert werden: den eigentlichen Trackingalgorithmus und die zum Tracking verwendeten Merkmale. Ein Trackingalgorithmus kann letztlich nur so gut und robust funktionieren, wie es die verwendeten Merkmale, die auf das Vorhandensein des zu verfolgenden Objekts an einer bestimmten Raumposition hindeuten, zulassen.

Das Verfolgen von Objekten stellt im eigentlichen Sinn eine Zustandsschätzung dar, unabhängig davon, ob es sich dabei um die Pose eines Roboters innerhalb seiner Einsatzumgebung oder die Position eines Objekts in einer Bildsequenz handelt. Die Bestimmung der wahrscheinlichsten aktuellen Position stellt also die Suche nach der maximalen A-Posteriori-Wahrscheinlichkeit über alle möglichen Positionen dar. Das Kalman-Filter oder auch das erweiterte Kalman-Filter stellen einen möglichen methodischen Apparat zur Behandlung dieses Schätzproblems dar (einen guten Überblick liefert [May79]). Problematisch ist jedoch die Behandlung mehrerer Zustandshypothesen, die mit Hilfe separater Kalman-Filter modelliert werden müssen.

Als Alternative zum Kalman-Filter schlagen ISARD und BLAKE [IB98] deshalb einen Algorithmus vor, der unter dem Namen Condensation-Algorithmus (conditional density propagation, siehe auch Abschnitt 7.3) bekannt wurde und das Problem des Kalman-Filters, nur unimodale Verteilungsfunktionen behandeln zu können, umgeht. Weiterhin existieren mittlerweile eine Reihe von Verfahren, die eine sehr effiziente Umsetzung dieses Trackingalgorithmus' gestatten (z.B. [FDBT99]).

Trotz leistungsfähiger Trackingalgorithmen stellt das kontinuierliche, visuell-basierte Verfolgen von Personen unter realen, nicht beeinflussbaren Umgebungen nach wie vor ein schwierig zu lösendes Problem dar. Neben dem eigentlichen Trackingalgorithmus ist die Frage zu klären, welche Merkmale sich zum Personentracking eignen. Grundsätzlich kommen die bereits im Rahmen der Personenlokalisierung genannten Merkmale auch für das Personentracking in Betracht. Bezüglich der erreichbaren Robustheit gelten ebenso die für die Personenlokalisierung getroffenen Aussagen zur Kombination verschiedener Merkmale oder Sensormodalitäten (siehe auch [BWK⁺01b] oder [FZ00]). Alternativ zum visuell-basierten Ansatz wurden

weiterhin Verfahren zum Verfolgen von Personen mittels Laserscanner vorgeschlagen (siehe z.B. [SB01] [SBFC01]). Vorteile ergeben sich hier in erster Linie für das Personentracking, da die verfügbare 2D-Information für eine sichere Lokalisation als zu unspezifisch angesehen werden muss.

Auch hier kann wie schon im Kontext der Personenlokalisierung postuliert werden, dass für einen praktischen Realwelteinsatz erst durch die kombinierte Nutzung verschiedener Sensorysysteme die notwendige Robustheit erreicht werden kann.

Gestenbasierte MRI Ursprünglich maßgeblich getrieben durch die Arbeiten zur Erkennung und Synthetisierung von Gebärdensprache, bildet mittlerweile der gesamte Komplex der Gestenerkennung und -interpretation einen weiteren Schwerpunkt der Mensch-Maschine-Kommunikation (siehe auch [BGG⁺99] [WF98] [HE97]). Hidden-Markov-Modelle, deren Ursprung in der Sprachverarbeitung liegt, bilden mittlerweile für nahezu alle Systeme zur Gestenerkennung die methodische Basis (siehe z.B. [SP95] [RKE98]). Als Gesten werden meist Bewegungen von Kopf und Händen einer Person aufgefasst, wobei grob vereinfacht zwei wesentliche Zielrichtungen verfolgt werden: Zum einen nutzt man Gestik, um herkömmliche Eingabegeräte wie Tastatur oder Maus zu ersetzen. Typischerweise geht man dabei von einem vorab definierten Alphabet möglicher Gesten und einer festgelegten Bedeutung der Gesten für das erkennende System aus, was zwangsläufig die Frage nach der damit erreichbaren Intuitivität der Interaktion aufwirft, da diese Form der Eingabe auch wieder durch den menschlichen Interaktionspartner erlernt werden muss. Für eine gestenbasierte Mensch-Roboter-Interaktion gelten also derzeit sinngemäß die gleichen Limitierungen, die bereits hinsichtlich eines sprachbasierten Dialogs benannt wurden, da auf der Nutzerseite die Kenntnis der durch den Roboter erkennbaren Gesten und deren Bedeutung vorausgesetzt werden muss. Abgesehen von der erreichbaren Intuitivität kommt erschwerend hinzu, dass im Realwelt-Einsatz eine sichere und robuste Gestenerkennung ausgesprochen schwierig zu bewerkstelligen ist. Bekannte Systeme zur gestenbasierten Interaktion setzen deshalb meist stationäre und stark vereinfachte Umgebungsbedingungen (Szeneninhalt, Beleuchtung etc.) voraus. Perspektivisch relevant für den Einsatz zur MRI ist die Gestenerkennung jedoch in jedem Fall. So könnten z.B. aus der Beobachtung spontan ausgeführter Gesten Rückschlüsse auf den Interaktionsstatus des Benutzers abgeleitet werden. Dies trifft insbesondere dann zu, wenn man den Gestenbegriff weiter fasst und auch Körpersprache und Mimik als wesentliche zwischenmenschliche Kommunikationsform mit einbezieht (siehe z.B. [EP95] [IEE98] [IEE00]). Allerdings steckt die maschinelle Erkennung und Interpretation von Körpersprache aus wissenschaftlich-methodischer Sicht noch in den Kinderschuhen. Erste Ansätze gehen auch hier wiederum von einem fest vorgegebenen Alphabet von Aktionen aus, das verarbeitet werden kann [Dav96] [DB97].

Weiterhin sinnvoll erscheint perspektivisch die kombinierte Nutzung von Sprache und Gestik, da sich beide Kommunikationsformen oft gegenseitig ergänzen [Wre00] [Kah96]. Gestik wird in Verbindung mit Sprache besser interpretierbar oder vermittelt Zusatzinformationen, die allein aus dem Sprachkanal nicht extrahierbar sind. Allerdings beschränken sich die bekannten Anwendungen auf den Bereich der Virtuellen oder Erweiterten (Augmented) Realität (siehe z.B. [BGG⁺99] [WF98] [FBR98] [Wre00]), während unmittelbar aus dem Robotikbereich kaum Applikationen bekannt wurden, die wirklich praktisch einsetzbar sind. Einen ersten Ansatz bilden die Arbeiten im Rahmen des BMBF-Leitprojektes SMARTKOM, in denen einfache Zeigegesten für die Bedienung eines Informationsterminals genutzt werden (siehe auch

[BMB01]).

2.4.3 Roboterartikulation

Interaktion erfolgt stets bidirektional. Übertragen auf einen Serviceroboter bedeutet das, dass das System nicht nur über die Fähigkeit verfügen muss, seine Umgebung und potentielle Interaktionspartner wahrzunehmen und zu beobachten, sondern dass es ebenso über Möglichkeiten verfügen muss, seinen aktuellen Interaktionsstatus (interner Zustand) seinem Interaktionspartner mitzuteilen. Im Gegensatz zur Wahrnehmungsseite, wo die Problematik der uneingeschränkt funktionsfähigen Spracherkennung eher kritisch gesehen werden muss, bietet sich der Sprachkanal für die Artikulation des Serviceroboters geradezu an. Erfordert die Ausgabe von Informationen auf einen Bildschirm, dass dieser von der aktuellen Position des Interaktionspartners auch eingesehen werden kann, so besteht hinsichtlich einer Sprachausgabe diese Einschränkung nicht. Für die konkrete Realisierung kommen sowohl die Verwendung vorgefertigter Dateien als auch der Einsatz von Sprachsynthesystemen in Betracht.

Je nach Aufbau des Roboters kommen auch Verfahren zur Erzeugung einer maschinellen Körpersprache in Betracht. So kann ein Serviceroboter seinem menschlichen Interaktionspartner durch seine aktive Zuwendung deutlich machen, dass der Benutzer vom Roboter wahrgenommen wird oder kontinuierlicher Kontakt besteht. Durch typische menschenähnliche und damit intuitiv verständliche Bewegungen eines Active-Vision-Kopfes können Zustimmung (Nicken) oder Ablehnung (Kopfschütteln) einfach ausgedrückt oder vermittelt werden.

2.5 Lernfähigkeit

Je komplexer die Serviceleistung ist, die der Roboter vollbringen soll, und je schwieriger sich die Umgebung gestaltet, in der der Serviceroboter operiert, umso stärker rücken Verfahren, die eine Online-Adaption des angestrebten intelligenten Systems an veränderliche Aufgaben und Umgebungsbedingungen zulassen, in den Mittelpunkt des Interesses. Der prinzipielle Hintergrund erscheint völlig einleuchtend: Kann man bei einer leicht überschaubaren Aufgabe in einer wohldefinierten, vielleicht sogar robotergerecht gestaltbaren Umgebung noch davon ausgehen, dass das Gesamtverhalten des Systems vom Designer per Hand gut programmiert und abgestimmt werden kann, so gilt dies bei anspruchsvollen Szenarien in realen, nicht modifizierbaren Umgebungen keineswegs mehr. Deshalb erhofft man sich von Lernverfahren, die dem Roboter die Fähigkeit verleihen, sich selbständig anzupassen, entscheidende Vorteile, da so nur noch der Lernalgorithmus als Voraussetzung für eine erfolgreiche Adaption realisiert werden muss, man aber nicht mehr vor dem Problem steht, alle Eventualitäten und Ausprägungen der realen Welt bereits vorab explizit berücksichtigen zu müssen.

Dies betrifft alle o.g. Teilleistungen oder Verhaltensmodule und verdeutlicht gleichzeitig, dass die Frage der Lernfähigkeit bei typischen Industrieroboter-Applikationen kaum eine Rolle spielt.

Quasi alle bekannten Lernparadigmen (einen sehr guten Überblick über die verschiedensten Spielarten maschinellen Lernens liefert z.B. [Mit97]) lassen sich an verschiedenen Stellen einer Serviceroboter-Gesamtarchitektur einsetzen.

2.5.1 Sensordatenverarbeitung

Im Bereich der Sensordatenverarbeitung sind mit Abstand am häufigsten Lernverfahren anzutreffen. Insbesondere neuronale Lernparadigmen werden hier eingesetzt, um deren typische Eigenschaften wie Fehlertoleranz und Generalisierungsfähigkeit auszunutzen.

Anhand von zwei typischen Beispielen soll dies verdeutlicht werden: So eignen sich unüberwachte Lernverfahren (meist in Form von Vektorquantisierern) für die Erzeugung von kompakten sensorischen Repräsentationen oder für die Abbildung eines kontinuierlichen mehrdimensionalen Zustandsraumes auf eine endliche Menge diskreter Prototypzustände (siehe z.B. [Fri98] [GSS98] [GHSS99]). Überwachte Lernverfahren lassen sich nutzbringend für das Lernen von Sensormodellen einsetzen, wie das bei der in Abschnitt 7.1.2 Abbildung von Ultraschallmesswerten auf die Belegheitswahrscheinlichkeit einer Zelle in einem egozentrischen, rasterorientierten (lokalen) Umgebungsmodell der Fall ist.

2.5.2 Direkte Sensor-Aktor-Abbildungen

Das Erlernen von direkten (reaktiven) Sensor-Aktor-Abbildungen kann ebenfalls erfolgreich unter Zuhilfenahme von überwachten Lernverfahren realisiert werden. Als typisches Beispiel sei hier auf die von POMERLEAU entwickelte ALVINN-Architektur [Pom93] verwiesen, in der ein einfaches Multilayer-Perceptron ein Straßenfahrzeug anhand visueller Information steuert. Dieser Ansatz wurde auch in eigenen Forschungsarbeiten für die Navigation eines mobilen Roboters in einer Indoor-Umgebung erfolgreich eingesetzt [KBSG97b] [KBSG97c] [KBSG97a].

Reinforcement-Lernverfahren [Bar92] [KLM96] sind prädestiniert für den Einsatz zur Aktionsauswahl und zur Generierung von optimalen Verhaltens- oder Aktionsstrategien. Sie besitzen allerdings den Nachteil, dass sie, bedingt durch die notwendige große Zahl an Trainingsschritten, sehr zeitaufwendig sind. Während Reinforcement-Lernverfahren zur Ausbildung optimaler Interaktionsstrategien für Serviceroboter sehr gut einsetzbar sind (siehe Abschnitt 3.2), ergeben sich meist praktische Probleme beim Erlernen von Navigationsverhalten (z.B. für die Hindernisvermeidung), da das „Erleben“ des Fehlverhaltens eines großen und oft auch schweren Roboters in einer realen Umgebung oft nicht zugelassen werden kann. Weiterhin spielt auch hier der notwendige Zeitaufwand eine entscheidende Rolle. Deshalb beschränkt man sich hier, wie dies auch am eigenen Fachgebiet üblich ist, auf den Einsatz von Miniaturrobotern in Spielwelten, um methodische Fortschritte beim Erlernen von Navigationsstrategien zu erreichen [SSG98] [SDG⁺00] [HG01], die dann perspektivisch auf die eigentlichen Serviceroboter übertragen werden können.

2.5.3 Lernen von Umgebungsmodellen

Das Erlernen von Umgebungsmodellen bildet einen weiteren Schwerpunkt im Rahmen von Serviceroboter-Applikationen und stellt gleichzeitig eine wesentliche Voraussetzung für die Autonomie solcher Roboter dar. Nur wenn der Roboter sein Umgebungsmodell selbst aufbauen und entsprechend aktualisieren kann, ist der Umgang mit dynamischen Umgebungen, die ständigen Änderungen unterworfen sind (Heimbereich oder BauMarkt), möglich. Oft spielen hier auch geeignete Explorationsstrategien eine Rolle, deren Ziel darin besteht, dass der Roboter selbstständig seine Einsatzumgebung möglichst vollständig erkundet.

Die grundlegende Schwierigkeit beim Erlernen von Umgebungsmodellen besteht darin, dass zu Beginn des Lernprozesses noch kein Modell existiert, der Roboter aber für die kontinuierliche Bestimmung seiner aktuellen Pose (Position und Orientierung) eigentlich genau dieses benötigt, weshalb man hier auch von einem so genannten Henne-Ei-Problem spricht.

Gelingt es, die aktuelle Roboterpose (bezogen auf einen definierten Startpunkt) kontinuierlich und hinreichend exakt zu ermitteln, so stellt das Erlernen eines konsistenten Umgebungsmodells kein prinzipielles Problem mehr dar, wie das MORAVEC und ELFES bereits vor mehr als 10 Jahren demonstrierten [ME85] (siehe dazu auch Kapitel 7). Für den umgekehrten Fall, in dem ein konsistentes Umgebungsmodell bereits vorliegt und die Aufgabe darin besteht, die aktuelle Pose des Roboters anhand der Sensorinformation zu ermitteln, existieren ebenfalls verschiedene Algorithmen [BEF96].

In jüngster Zeit sind eine Reihe von Methoden vorgeschlagen worden (siehe z.B. [TBF00] [BH00] [HH01]), die beide Probleme, den Aufbau des Umgebungsmodells und die dazu notwendige Selbstlokalisierung, simultan betrachten und sich dabei durchweg probabilistischer Modellierungstechniken (Bayes-Filter, Maximum-Likelihood-Schätzung, EM-Algorithmus) bedienen. Fortschritte wurden hier insbesondere hinsichtlich der echtzeitfähigen Umsetzung dieser Verfahren, deren Vorteil in erster Linie in der enormen Robustheit besteht, erzielt [TBF00] [FDBT99], was beispielsweise einen vollständigen Verzicht auf Odometriedaten möglich und diese Verfahren sehr universell einsetzbar macht.

2.5.4 Lernen und MRI

Vor dem Hintergrund der Mensch-Roboter-Interaktion lassen sich zwei grundlegende Lernschemata unterscheiden: Zum einen versucht man, während der MRI Algorithmen (insbesondere aus dem Reinforcement-Lernen) einzusetzen, die dem Serviceroboter das Erlernen optimaler Strategien zur Aktionsauswahl ermöglichen, um so die jeweiligen Bedürfnisse bzw. Anforderungen des Benutzers sukzessive besser befriedigen zu können [AMH⁺96]. Zum anderen spielt der Erwerb von Wissen in direkter Interaktion zwischen Serviceroboter und Benutzer oder Supervisor in vielen Szenarien eine zentrale Rolle [KHH⁺01] [BBM⁺98] (siehe auch Abschnitt 4.1.4). Damit wird einer zunehmenden Komplexität des zu erlernenden Problems Rechnung getragen, indem der Roboter die Fertigkeit nicht allein (autonom), sondern in Interaktion mit dem Menschen erwirbt. Typische Beispiele sind das interaktive Erlernen von Umgebungsmodellen, bei dem der menschliche Supervisor dem Roboter typische Plätze seiner Umgebung zeigt und gegebenenfalls auch benennt, was für den Aufbau semantischer Umgebungsmodelle (siehe auch Abschnitt 2.2.4) von Bedeutung ist. Einen weiteren Schwerpunkt bilden Algorithmen zum so genannten Imitationslernen. Hier werden z.B. bestimmte Bewegungsabläufe vom Menschen vorgemacht und vom Roboter dann entsprechend nachvollzogen (siehe z.B. [Tri99] [BM01a] [NM01] [BBM⁺98] [MMH01]).

Einen weiteren Schwerpunkt wird zunehmend auch das Lernen im Kontext der Dialogmodellierung bzw. das Lernen unmittelbar während der MRI bilden [RPT00]. Hier spielt die Modellierung des Interaktionsprozesses bzw. die Modellierung der Aktionsgenerierung durch das Servicesystem mit Hilfe partiell beobachtbarer Markov-Entscheidungsprozesse (POMDP's) eine zentrale Rolle.

2.6 Szenariospezifische Serviceleistungen

Von der konkreten Applikation des Serviceroboters hängt letztendlich ab, was die eigentlichen Serviceleistungen des Systems sind und in welcher Art und Weise diese umgesetzt werden. Die Teilkomplexe Navigation (Abschnitt 2.2), Mensch-Roboter-Interaktion (Abschnitt 2.4) und Steuerarchitektur (Abschnitt 2.3) werden i.d.R. ebenso von diesen Serviceleistungen mitbestimmt. Navigation, Mensch-Roboter-Interaktion, konkrete Serviceleistung und Steuerarchitektur werden also immer eine (untrennbare) Einheit bilden. Das folgende Beispiel, das dem im Teil III dieser Arbeit ausführlich behandelten eigenen Forschungsprojekt entlehnt ist, soll diese Einheit veranschaulichen: Man stelle sich einen Roboter vor, dessen Serviceleistung u.a. darin besteht, Baumarktkunden zu einem benutzerdefinierten Marktbereich zu lotsen. Für die Erfüllung dieser Aufgabe sind auf der Navigationsseite zunächst eine Reihe von Voraussetzungen notwendig; der Roboter benötigt Wissen über seine Einsatzumgebung (globale Karte), er muss in der Lage sein, seine aktuelle Position innerhalb dieser Karte zu bestimmen, und er benötigt Mechanismen, die ihm die Planung eines Weges von der aktuellen Position zur benutzerdefinierten Zielposition gestatten. Um mit dem Baumarktkunden in Interaktion treten zu können, benötigt der Serviceroboter entsprechende Ein- und Ausgabegeräte sowie Methoden, um z.B. die Anforderung des Kunden, möglichst schnell zu einem Artikelstandort zu gelangen, in die entsprechenden Steueranweisungen (Auffinden des Artikels in der Umgebungsrepräsentation, daran anschließende Pfadplanung und Fahrt zu diesem Standort) umsetzen zu können. Weiterhin ist die möglichst intuitive und natürliche Interaktionsfähigkeit des Serviceroboters für die Akzeptanz solcher Systeme enorm wichtig. Der Serviceroboter sollte in der Lage sein, eine Person in seiner Einsatzumgebung aktiv wahrzunehmen, von sich aus den Kontakt zu seinem menschlichen Interaktionspartner suchen und sich nicht zuletzt intuitiv verständlich machen können. All dies sind Voraussetzungen dafür, dass der Serviceroboter nicht Gefahr läuft, das gleiche traurige Dasein zu führen, welches den oftmals anzutreffenden stationären Informationsterminals beschieden ist, die endlos ihre Informationen präsentieren, völlig unabhängig davon, ob sich überhaupt jemand dafür interessiert. Aufgabe der Verhaltenskoordination wiederum ist es, u.a. dafür zu sorgen, dass der Kontakt zu dem aktuell eingebuchten Kunden während der Fahrt zu diesem Standort aufrecht erhalten werden kann. Dies bedeutet beispielsweise, dass sich im System je nach aktueller Situation die beiden Modi Personentracking und Zielfahrt wechselseitig beeinflussen müssen, um zu gewährleisten, dass der Kunde dem System bei der Zielfahrt auch wirklich folgt.

Übergreifend über die verschiedenen denkbaren Applikationsszenarien sollen nachfolgend verschiedene Aspekte und Fragestellungen benannt werden, durch die die eigentliche Applikation die Teilleistungen Navigation, Verhaltenskoordination / Systemintegration und Mensch-Roboter-Interaktion maßgeblich determiniert:

- Benötigt das System neben einer mobilen Plattform Manipulatoren, um seine Serviceleistung realisieren zu können? Dies wäre z.B. dann der Fall, wenn ein Roboter zur Unterstützung älterer oder behinderter Menschen auch Handreichungen (z.B. Greifen und Übergeben einer Tasse oder des Telefons) erledigen soll.
- Wie muss die Repräsentation der Umwelt erfolgen? Reicht die Fähigkeit zur Vermeidung von Hindernissen während der Eigenbewegung des Systems aus, oder ist Wissen

über die Einsatzumgebung erforderlich (z.B. der Standort des Telefons in einer Karte der Wohnung)?

- Wenn Wissen über die geometrischen Gegebenheiten der Einsatzumgebung notwendig ist, steht die Frage, wie die geeignete Repräsentation dieses Wissens aussehen sollte. Für eine Karte der Einsatzumgebung ist z.B. entscheidend, wie genau die Ortsauflösung sein muss oder wie effizient eine Pfadplanung auf dieser Repräsentation realisiert werden kann.
- Die angestrebte Servicefunktionalität sowie die konkreten Gegebenheiten der Einsatzumgebung sind maßgeblich für die Sensorik, die zur Umgebungserfassung verwendet werden kann bzw. muss und für die Art und Weise der Umgebungsmodellierung. Handelt es sich um eine Umgebung mit sehr regelmäßiger Struktur (nur senkrecht zueinander verlaufende Wände), so reichen unter Umständen ebenenweise angeordnete Ultraschallsensoren aus, deren Messwerte unter Zuhilfenahme der bekannten Umgebungsstruktur (Anwendung von Heuristik) geeignet interpretiert werden können. Handelt es sich dagegen um eine sehr vielgestaltige Einsatzumgebung (was sicherlich den allgemeineren Fall darstellt), gestaltet sich die Verwendung von Heuristiken schwierig, so dass über den Einsatz von 3D-Laserscannern oder Kameras nachgedacht werden muss.
- Über welche Interaktionsfähigkeiten muss der Serviceroboter verfügen? Reicht eine einfache Bedienoberfläche zur Interaktion aus? Welche Annahmen können zur bedienerseitigen Kenntnis der zu erwartenden Servicefunktion und der Leistungsfähigkeit des Systems generell getroffen werden? Ist das System vorrangig zur Interaktion mit einem speziellen Benutzer (im Sinne eines Personal Robot) gedacht oder soll es mit beliebigen Nutzern interagieren können? Im letzteren Fall können entsprechend weniger Vorkenntnisse und Anpassungswille auf der Nutzerseite vorausgesetzt werden.
- Wie komplex ist die Gesamtaufgabe, die der Serviceroboter bewältigen soll? Lässt sich diese Gesamtaufgabe sinnvoll in Teilaufgaben zerlegen, deren Umsetzung separat angegangen werden kann?
- Welche Methoden sind zur Koordination der Teilaufgaben zu einem sinnvoll funktionierenden Gesamtsystem geeignet?
- Können bzw. müssen Lernverfahren auf dem Serviceroboter implementiert werden, die diesem eine flexible Adaption an wechselnde Umgebungsbedingungen, sich unterschiedlich verhaltende Nutzer, veränderliche Aufgaben etc. , ermöglichen?

Auf weitere konkrete Serviceleistungen wird in den Kapiteln im Teil II eingegangen, der sich mit ausgewählten Beispielszenarien und Applikationen beschäftigt.

2.7 Zusammenfassung

In diesem Kapitel wurde ein Überblick über die notwendigen Basisfunktionen interaktiver mobiler Serviceroboter gegeben. Ausgehend von der historischen Entwicklung der entsprechenden

wissenschaftlich-technischen Methoden wird deren aktueller Entwicklungsstand kurz charakterisiert. Anschließend erfolgt eine Diskussion dazu, wie diese Methoden abhängig von den Anforderungen und Randbedingungen der jeweiligen Applikation sinnvoll eingesetzt werden können. Dabei wird insbesondere herausgearbeitet, dass im Kontext eines Servicerobotik-Szenarios alle Basiskomponenten in ihrer Gesamtheit betrachtet werden müssen, da eine Vielzahl wechselseitiger Abhängigkeiten zwischen diesen besteht.

Hinsichtlich der robusten Navigation mobiler Roboter, auch in ausgedehnten und nicht beeinflussbaren Einsatzumgebungen, konnte mittlerweile ein Entwicklungsstand erreicht werden, der einen erfolgreichen Einsatz dieser Verfahren realisierbar erscheinen lässt. Das bedeutet jedoch nicht, dass ein konkretes Einsatzfeld nicht dennoch die Weiterentwicklung, zumindest aber die Anpassung dieser Verfahren notwendig macht (siehe dazu auch Abschnitt 7.1.3).

Die größte Herausforderung auf der Methodenseite besteht sicherlich im Rahmen der Mensch-Roboter-Interaktion, soll diese zukünftig mehr und mehr intuitiv und natürlich erfolgen. Der bislang erzielte Methodenfortschritt darf nicht darüber hinwegtäuschen, dass man sich hier erst am Anfang des Weges hin zur völligen Substitution der klassischen Eingabe- oder Interaktionsformen, wie Tastatur, Maus oder Touchscreen, bewegt. Für alle kurzfristig anvisierten Szenarien stellt deshalb die schlüssige Kombination klassischer und natürlicher (Geräusche, Sprache, Sehen, Gestik etc.) Interaktionsformen einen sinnvollen und gangbaren Weg dar. Genau dieser Weg wird auch im Teil III der Arbeit, der das eigene Forschungsprojekt zur Entwicklung eines interaktiven mobilen Shopping-Assistenten beschreibt, beschritten.

Teil II

Szenarien und Applikationen

Kapitel 3

Tourguide-Roboter

Nachdem mobile Roboter über erste Fähigkeiten zur autonomen Navigation verfügten und damit eine Entwicklung begann, die weg vom ausschließlichen Einsatz von Robotern in der Industrieautomation hin zu Servicerobotern führte, war die Funktion des Tourguides eines der ersten Anwendungsszenarien. Ein weiterer Grund, einen Serviceroboter als Tourguide einzusetzen, liegt sicherlich auch darin, dass die Fähigkeit zur autonomen Navigation und zur einfachen Mensch-Maschine-Kommunikation ausreicht, um ein solches System umzusetzen und beispielsweise keine manipulatorischen Komponenten benötigt werden. Kapitel 3 ist den bekanntesten Tourguide-Projekten gewidmet. Letztlich enthält das im Teil III der Arbeit behandelte System eines Shopping-Assistenten im BauMarkt ebenso viele Aspekte einer Tourguide-Anwendung.

Im Kapitel 4 stehen dann Assistenzsysteme im Vordergrund. Einen Schwerpunkt stellen hier die so genannten Nursebot-Szenarien dar, die die Entwicklung von Robotern zur Unterstützung älterer und hilfsbedürftiger Menschen zum Ziel haben. Weitere Assistenzsysteme betreffen Roboter, die im Produktionsbereich flexibel für Handhabungs- und Transportaufgaben eingesetzt werden können, sowie einige universitäre Projekte.

Sowohl Tourguide- als auch Assistenzrobotern ist gemein, dass diese neben der Fähigkeit zur robusten Navigation in einem typischerweise dynamischen Einsatzfeld auch über entsprechende Methoden zur intuitiven MRI verfügen müssen.

Betont werden soll, dass die detaillierte Beschreibung weniger im Vordergrund steht, sondern es wird vielmehr versucht, aus einer etwas abstrakteren Sicht grundlegende Zielstellungen und applikationsübergreifende Aspekte herauszuarbeiten, bevor in Kapitel 5 ein abschließendes Resümee, insbesondere auch vor dem Hintergrund der im Teil III der Arbeit dargestellten eigenen Forschungsarbeiten, gezogen wird.

3.1 Der Roboter POLLY als Laborführer

Der Roboter POLLY (Abbildung 3.1, siehe auch <http://www.ai.mit.edu/people/polly/polly.html>) wurde als einfaches Low-Cost-System von 1992 bis 1993 am MIT Artificial Intelligence Laboratory entwickelt [Hor93] [Hor94a] [Hor98] und war eines der ersten Robotersysteme, das als Tourguide fungierte. Seine Aufgabe bestand darin, innerhalb eines Flurbereiches zu patrouillieren, dabei Personen wahrzunehmen, diese gezielt anzusprechen und Touren zu offerieren.

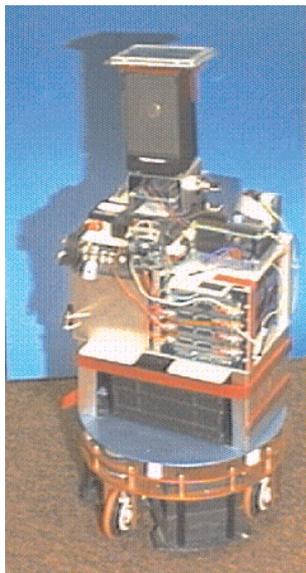


Abbildung 3.1: Robotersystem POLLY. Hardware-Konfiguration: Real World Interface B12 base, C30 32-bit DSP (Pentek 4283), B/W Frame-grabber (Data Translation DT1451), B/W Video surveillance camera (Chinon CX-101), Voice synthesizer (RC Systems)

POLLY ist deshalb so hervorhebenswert, weil hier eine Reihe von Ansätzen, die sich mittlerweile auf dem Gebiet der Servicerobotik als gebräuchlich bzw. selbstverständlich etabliert haben, erstmalig explizit eingesetzt und untersucht wurden. So nutzte POLLY ausschließlich visuelle Sensorik, und die gesamte benötigte Rechenleistung wurde on-board zur Verfügung gestellt. Das Paradigma des aktiven Sehens, welches sich gerade zu etablieren begann, wurde konsequent verfolgt und in Form verschiedener Module, die den einkommenden Datenstrom im Kontext einer genau definierten Verhaltensleistung analysieren, umgesetzt. Für die Koordination dieser Module wurde ein verhaltensbasierter (reaktiver) Ansatz gewählt, der kein zentrales Umweltmodell verwendet, sondern die Umwelt verteilt in den verschiedenen Verhaltensmodulen aufgabenspezifisch repräsentiert (siehe dazu auch [Hor95]).

Weiterhin wurde versucht, durch konsequente explizite Einbeziehung von Wissen über die Einsatzumgebung (Form der Hindernisse, Verlauf der Wände, textuelle Eigenschaften des Fußbodens etc.) die Auswertung der Sensordaten zu vereinfachen bzw. zu beschleunigen.

3.2 RHINO und MINERVA

RHINO [BBC⁺95] und MINERVA stellen die wohl bekanntesten Roboter dar, die im Rahmen von Tourguide-Projekten eingesetzt wurden. Während RHINO 1997 einige Tage im Deutschen Museum in Bonn operierte [BCF⁺98], wurde MINERVA 1998 für die Dauer von zwei Wochen im Smithsonian's National Museum of American History in Washington eingesetzt [TBB⁺99a, TBB⁺00]. An beiden Projekten war ein nahezu identisches Team der Universität Bonn und der Carnegie Mellon University in Pittsburgh beteiligt.

Die Abbildungen 3.2 und 3.3 zeigen beide Robotersysteme während ihres Einsatzes als Museumsführer.

Die Aufgabe beider interaktiven Roboter bestand ähnlich wie bei POLLY darin, Personen zu lokalisieren, diese (über Sprache) zur Interaktion mit dem Roboter zu ermuntern und Touren durch das Museum anzubieten, wobei die Auswahl der gewünschten Tour via Touchscreen erfolgte.



Abbildung 3.2: Robotersystem RHINO, ein B21-Roboter (Real World Interfaces, USA), während seines Tourguide-Einsatzes im Deutschen Museum in Bonn 1997.



Abbildung 3.3: MINERVA während ihres Tourguide-Einsatzes im Smithsonian's National Museum of American History 1998.

Für die Lokalisation konnte die Tatsache (Heuristik) ausgenutzt werden, dass die Einsatzumgebung eines Museums kaum Änderungen unterworfen ist. Vorausgesetzt, der Roboter verfügt über eine hinreichend gut aufgelöste Karte dieser Umgebung und kennt seine aktuelle Pose, liefern die Abweichungen zwischen erwartetem und tatsächlich vorliegendem sensorischen Eindruck (Laserscan) gute Hypothesen über das Vorhandensein von Personen.

Die Bedeutung dieser beiden Tourguide-Projekte liegt in den folgenden Aspekten begründet:

- Es konnte erstmalig erfolgreich demonstriert werden, dass der Einsatz solcher Systeme auch über einen längeren Zeitraum prinzipiell möglich ist.
- Entscheidend für den Erfolg war der konsequente Einsatz eines durchgehend probabilistischen Modellierungsrahmens, der explizit die zwangsläufig auftretenden Unsicherheiten hinsichtlich der Eigenbewegung des Roboters, der sensorischen Wahrnehmung und der Umgebung selbst berücksichtigt (siehe auch [Thr00a]).

- Probabilistische Belegtheitsgitterkarten (occupancy maps), von denen man zunächst erwartet hatte, dass diese nur für Umgebungen geringer Dimension effizient eingesetzt werden können, wurden erstmals für Umgebungen dieser Dimension skaliert.
- Beide Roboter waren für die Interaktion mit Menschen konzipiert, die über kein Vorwissen bezüglich Robotertechnologien verfügen, wobei die Gesamtfunktionalität maßgeblich von dieser Fähigkeit zur Interaktion abhing. Dies betraf sowohl die Interaktion mit den Museumsbesuchern vor Ort als auch die Interaktion mit Benutzern über das World Wide Web.

Während RHINO lediglich eine vorgegebene metrische Karte seiner Einsatzumgebung probabilistisch erweiterte und adaptierte, war MINERVA in der Lage, das Modell ihrer Einsatzumgebung vollständig ohne vorgegebene Karte zu erlernen (siehe auch [TBF98]), ein Umstand, der für jegliche ähnlich geartete Serviceroboter-Anwendung von zentraler Bedeutung sein dürfte. Zusätzlich zum Belegtheitsgitter nutzte MINERVA eine ebenfalls erlernte visuelle Deckenkarte, um die Lokalisation des Roboters insbesondere in großen Freiräumen zu verbessern.

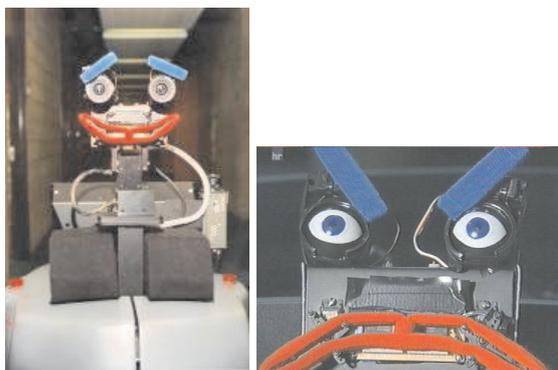


Abbildung 3.4: MINERVA's Gesicht. Durch entsprechende Stellung von Mund und Augenbrauen wird der „emotionale“ Zustand des Roboters für den Beobachter sehr leicht erfassbar.

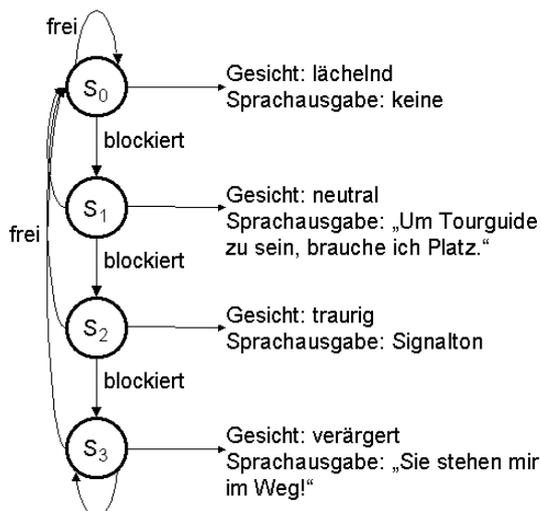


Abbildung 3.5: Zustandsdiagramm von MINERVA's Emotionen während einer Tour. Frei bzw. blockiert geben jeweils an, ob sich eine Person auf dem geplanten Pfad des Roboters befindet oder nicht (entnommen und adaptiert aus [TBB⁺00]).

Im Vergleich zu RHINO verfügte MINERVA über deutlich erweiterte Interaktionsfähigkeiten. Ein stilisiertes, steuerbares Gesicht sollte die Besucher zur Interaktion mit dem Roboter ermu-

tigen (siehe Abbildung 3.4) und den Roboter menschenähnlich erscheinen lassen, um eventuell vorhandene Berührungängste zu überwinden. Mit Hilfe dieses Gesichts und entsprechend angepasster Sprachausgaben ließ sich auch der „emotionale“ Zustand des Roboters selbst, der mit Hilfe eines endlichen Zustandsautomaten modelliert wurde (Abbildung 3.5) und von freundlich bis ärgerlich reichte (Abbildung 3.4), sehr leicht und intuitiv verständlich verdeutlichen.

Neben dem Erlernen des probabilistischen Umgebungsmodells kam ein Reinforcement-Lernverfahren zum Einsatz, um die während der Interaktion ausgewählten Aktionen besser an die jeweilige Situation anpassen zu können. So wurde beispielsweise untersucht, welche Aktionen sich besser eignen, die Aufmerksamkeit auf sich zu ziehen und die Personen zum Folgen während der Tour zu animieren. Im Ergebnis dieser Experimente konnte festgestellt werden, dass sich ein freundlicher und smarter Habitus offensichtlich besser eignet, mit Personen in Kontakt zu kommen.

Weiterhin war MINERVA in der Lage, abhängig von der aktuellen Situation ihre Touren zu adaptieren.

Die Arbeiten aus den Projekten RHINO und MINERVA werden u.a. in einem Tourbot-Projekt der Universität Freiburg in Kooperation mit griechischen Partnern fortgesetzt (<http://www.ics.forth.gr/tourbot/>). Hier steht die Internet-basierte Fernsteuerung des Roboters im Museum im Vordergrund. Der Roboter fungiert dabei als mobiler Avatar des nicht physisch anwesenden Besuchers. Diese Art der Roboteranwendung ist nicht nur für Museen interessant, sondern perspektivisch auch für den im Teil III dieser Arbeit beschriebenen Shopping-Assistenten.

Kapitel 4

Assistenz-Roboter

4.1 Nursebot-Szenarien

Unter dem „Nursebot“-Begriff sollen einige ausgewählte Szenarien zusammengefasst werden, die eine aktuelle Entwicklung im Serviceroboterbereich in Richtung persönlicher Assistenzsysteme für ältere und unterstützungsbedürftige Personen widerspiegeln. Diesem Anwendungsfeld für Serviceroboter wird längerfristig ein enormes Marktpotential zugemessen (siehe auch Abschnitt 1.5). Für den Einsatz interaktiver mobiler Systeme als Nursebots sprechen maßgeblich folgende Gründe und Überlegungen:

- Die prognostizierte demoskopische Entwicklung der nächsten Jahre belegt, dass der Anteil der älteren Personen, bezogen auf die Gesamtbevölkerung, deutlich zunehmen wird und diese zunehmend in Ein-Personen-Haushalten leben werden.
- Der Einsatz von Personal Robots verfolgt hier das Ziel, die Lebensqualität dieser Personen zu verbessern, indem die typischen Handicaps dieser Altersgruppe ausgeglichen werden.
- Für den Einsatz von Personal Robots in dem genannten Anwendungsbereich spricht weiterhin, dass einerseits die technologischen Voraussetzungen hinsichtlich Robustheit, Performanz und Verfügbarkeit gegeben sind und andererseits der Kostendruck im Gebiet Alten- und Krankenpflege enorm zugenommen hat und ein allgemein akzeptiertes gesellschaftliches Problem darstellt.
- Die Basisfunktionen eines solchen Systems könnten folgendermaßen aussehen:
 - kontinuierliche Beobachtung, Notfallerkennung und automatische Hilfealarmierung rund um die Uhr sind mit Hilfe solcher Systeme überhaupt erst möglich
 - sprach- und internetbasierte Auskunftssysteme (Wetter, TV-Programm) oder Einkaufssysteme können problemlos integriert werden und tragen wesentlich zur Lebensqualität bei
 - durch eine Internetanbindung kann die Telepräsenz des Hausarztes, von Verwandten oder des Pflegedienstes „on demand“ realisiert werden; damit verbunden ist eine bessere soziale Integration, die der persönlichen Vereinsamung vorbeugt

- intelligente Erinnerungsfunktion: Studien belegen, dass den Krankenversicherern enorme Kosten durch nicht oder falsch eingenommene Medikamente entstehen, eben weil bei der anvisierten Zielgruppe vorrangig Defizite bezüglich des Kurzzeitgedächtnisses bestehen
- solche Systeme können als intelligente Kommandozentrale nahezu alle technischen Abläufe innerhalb der Wohnung (Licht, Heizung etc.) entweder automatisch oder nach persönlichen Wünschen regeln
- einfach zu bedienende, videobasierte Kommunikationsplattform zwischen den Besitzern solcher Systeme mit dem Effekt der Herstellung sozialer Kontakte, des Abbaus der Vereinsamung etc.
- weiterhin bieten sie die Möglichkeit, bestimmte Vitalfunktionen kontinuierlich zu überwachen einschließlich einfacher Diagnose- und Therapiemöglichkeiten

Der Slogan des Nursebot-Projektes an der CMU in Pittsburgh (siehe Abschnitt 4.1.2), drückt die prinzipielle Zielrichtung beim Einsatz solcher Home-Care-Systeme folgendermaßen aus:

We have succeeded in helping people to live longer. Now we need to help them to live better.

Im Resultat sollen ältere oder hilfsbedürftige Menschen länger als bisher in ihrer gewohnten häuslichen Umgebung verbleiben können.

Anhand der Skizzen in Abbildung 4.1 sollen die wesentlichen Grundfunktionen eines solchen Serviceroboters veranschaulicht werden.

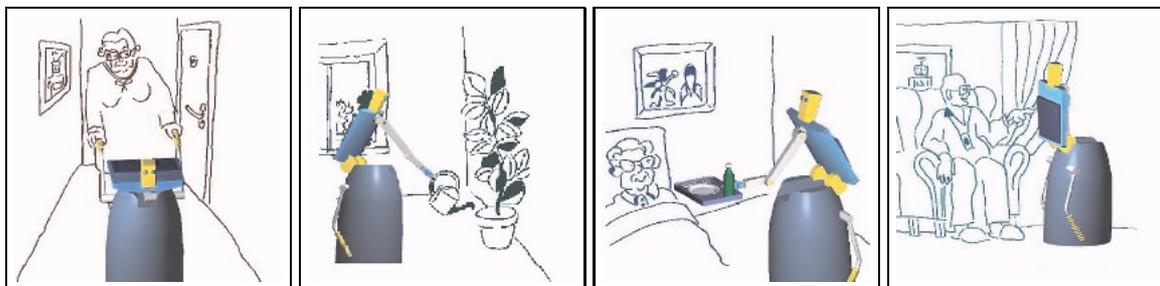


Abbildung 4.1: Grundfunktionen eines Serviceroboters zur Unterstützung älterer und pflegebedürftiger Personen. v.l.n.r.: Aufsteh- und Gehhilfe zur Verbesserung der Mobilität, Erledigung einfacher Tätigkeiten im Haushalt, Handreichung von gewünschten Objekten sowie Nutzung als Informationsterminal (nach [SM99, SSM98]).

Während die notwendigen Fähigkeiten zur robusten Navigation für Tourguide-Roboter und Nursebot-Systeme annähernd identisch sind, ergeben sich hinsichtlich der Mensch-Roboter-Interaktion einige gravierende Unterschiede, die bereits im Abschnitt 2.4 herausgearbeitet und diskutiert wurden. So kann man bei einem Nursebot davon ausgehen, dass der Benutzer zumindest bis zu einem gewissen Grade bereit ist, auf den Roboter und dessen Spezifika einzugehen, sich also an den Roboter zu adaptieren. Die Verwendung von Sprache als Kommunikationsmedium stellt sich hier in zweierlei Hinsicht als zentraler Aspekt dar: zum einen kann für die zu

betreuende Person die Bedienung eines Touchscreens infolge von körperlichen Handicaps problematisch sein; zum anderen kann beim Einsatz eines solchen Personal Robot weitestgehend sichergestellt werden, dass der menschliche Interaktionspartner die spezifischen Fähigkeiten des Systems zur Spracherkennung kennt, sich also beispielsweise an ein vorab definiertes Alphabet an möglichen Sprachkommandos hält. Es kann somit weitestgehend sichergestellt werden, dass sich Benutzer und Serviceroboter in einem gemeinsamen Referenzrahmen bewegen. Auch besteht die Möglichkeit, die Spracherkennung sprecherabhängig zu trainieren, ein Vorteil, der bei Servicerobotern, die mit beliebigen Personen interagieren sollen, nicht zum Tragen kommen kann.

Soll eine Spracherkennung eingesetzt werden, so verursacht diese allerdings im Vergleich zur Bedienung eines Touchscreens ein erhöhtes Maß an Unsicherheit auf der Eingabe- bzw. Wahrnehmungsseite, die durch geeignete Modellierung im Rahmen der Dialogführung berücksichtigt werden muss (siehe dazu auch [BFG⁺00, RPT00]).

Persönliche Service- und Assistenzsysteme operieren vorrangig im häuslichen Bereich, also in Umgebungen, in denen nicht nur Menschen anzutreffen sind (wie dies in Museen oder auf Bahnhöfen der Fall ist), sondern in denen Menschen wohnen, die also auf die speziellen Bedürfnisse und Fertigkeiten des Menschen ausgerichtet sind. Daraus ergibt sich ein weit engerer Kontakt zwischen Serviceroboter und Benutzer, als dies in den Tourguide-Szenarien gegeben war.

Die nachfolgenden Abschnitte erörtern kurz einige ausgewählte Projekte und Prototypen. Während bei Care-O-Bot (Abschnitt 4.1.1) und den Robotern FLO und PEARL (Abschnitt 4.1.2) die Interaktion mit unterstützungsbedürftigen Personen im Vordergrund steht, zielt der im DIROKOL-Projekt (Abschnitt 4.1.3) entwickelte Prototyp eher auf den klinischen Einsatz zur Entlastung des Klinikpersonals ab. Im Projekt MORPHA wird die Entwicklung von Assistenzsystemen sowohl für den häuslichen oder Pflegebereich (Abschnitt 4.1.4) als auch für industrielle Anwendungen angestrebt (Abschnitt 4.2.1).

4.1.1 Care-O-Bot

Am Fraunhofer Institut für Produktionstechnik und Automatisierung (FhG-IPA) Stuttgart wurde neben weiteren prototypischen Servicerobotern ein Konzept namens Care-O-Bot entwickelt [SM99, SSM98], das auf den Einsatz eines Personal Robot zur Unterstützung älterer und unterstützungsbedürftiger Personen in deren häuslicher Umgebung abzielt. Der erste Demonstrator eines solchen Systems wurde zur Hannover-Messe 1998 der Öffentlichkeit vorgestellt. Abbildung 4.2 zeigt das Oberteil des Care-O-Bot mit dem zur Interaktion verwendeten Touchscreen. Die beiden unter dem Touchscreen befindlichen Haltegriffe sollen eine Aufsteh- bzw. Gehhilfe realisieren.

Die folgende überblicksartige Systembeschreibung wurde der entsprechenden WWW-Seite (<http://www.ipa.fhg.de>) entnommen.

Mechanisches Design

Die hohe Komplexität eines mobilen Roboters wie die des Care-O-bot, machte am Fraunhofer Institut IPA eine besondere, anwendungsorientierte Planung und Konstruktion der Mechanik



Abbildung 4.2: Care-O-Bot.

erforderlich. Unter Berücksichtigung verschiedener Anforderungen wurde das in Abbildung 4.3 zusammengefasste technische Konzept entwickelt.

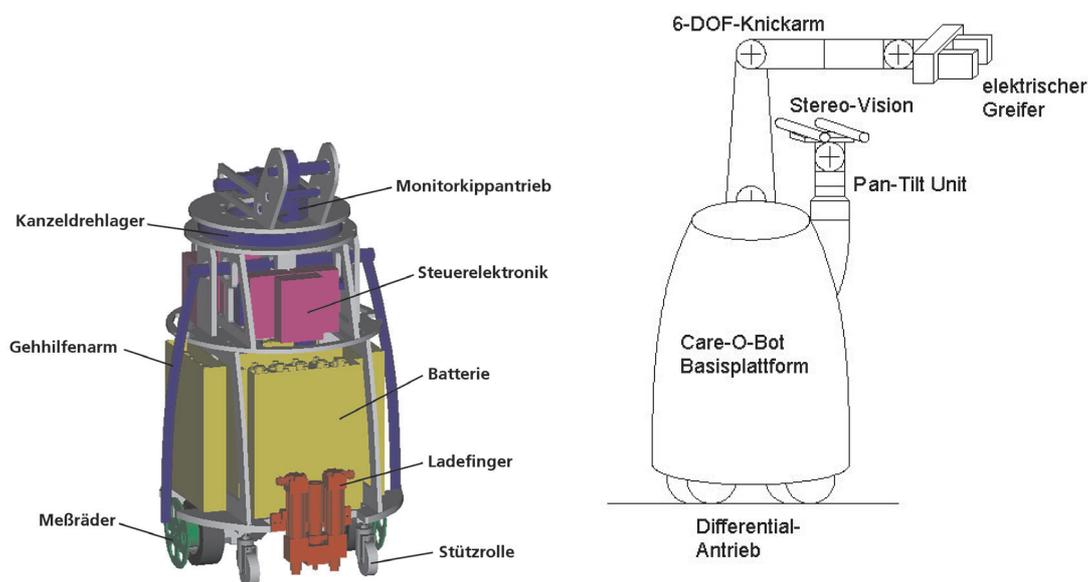


Abbildung 4.3: Hardware-Konzept des Care-O-Bot.

Neben ort- und kraftsensitiven Sicherheitsbumpen kommt als externer Sensor ein Laserscanner zum Einsatz.

Derzeit befindet sich ein neuer Prototyp des Care-O-bot im Aufbau (4.3 rechts). Neue Komponenten dieses Roboters sind ein Stereo-Kamerasystem, ein Manipulatorarm, sowie am Roboter angebrachte Gehhilfestützen.

Die mit Care-O-Bot begonnenen Arbeiten werden derzeit im Rahmen des BMBF-Leitprojektes MORPHA (siehe Abschnitt 4.1.4) weitergeführt. So ist mittlerweile ein Care-O-Bot II verfügbar, der beispielsweise in [GH01] beschrieben wird. Zur Mensch-Roboter-Interaktion werden dabei ein Touchscreen sowie Spracherkennung und Sprachausgabe genutzt. Visuelle Verfahren, sowohl zur Navigation als auch zur Interaktion, sind derzeit jedoch nicht implementiert oder zumindest konzeptionell vorgestellt worden.

4.1.2 FLO und PEARL

Seit etwa zwei Jahren kooperieren die Carnegie Mellon University Pittsburgh und die University of Pittsburgh in einem interdisziplinären Projekt, das ebenfalls die Entwicklung eines Personal Robots zur Unterstützung älterer und hilfsbedürftiger Menschen zum Ziel hat. Weitere Partner sind die Arbeitsgruppen der Universität Bonn (Prof. Cremers) und der Universität Freiburg (Prof. Burgard), die bereits in den in Abschnitt 3.2 beschriebenen Tourguide-Projekten kooperierten. Während die University of Pittsburgh Expertise auf dem Gebiet der Alten- und Krankenpflege einbringt, steuern die anderen genannten Partner ihre umfangreichen Erfahrungen auf dem Gebiet der mobilen Robotik bei.

Abbildung 4.4 zeigt die beiden Robotersysteme FLO (in Anlehnung an Florence Nightingale) und PEARL, die am Learning Robot Lab der Carnegie Mellon University Pittsburgh entwickelt wurden (siehe auch [BFG⁺00, RPT00]).



Abbildung 4.4: Nursebots am Learning Robot Lab der Carnegie Mellon University Pittsburgh. Links der erste Prototyp FLO, daneben der aktuelle Prototyp PEARL.

Aktuelle Forschungen im Rahmen dieses Projektes konzentrieren sich auf die Frage der Dialogmodellierung [RPT00] sowie auf robuste Verfahren zum Lokalisieren und Verfolgen von Personen [SB01, SBFC01], wobei der bereits im Rahmen der Tourguide-Projekte eingesetzte probabilistische Modellierungsrahmen auch hier wieder konsequent eingesetzt wird.

Als Eingabemodalitäten hinsichtlich der Mensch-Roboter-Interaktion werden ein Touchscreen sowie Sprache verwendet. Lokalisation und Tracking von Personen erfolgen derzeit ausschließlich auf der Basis eines Laserscanners, wobei auch erste Arbeiten zur Lokalisation und zum Tracking von Gesichtern für die Mensch-Maschine-Kommunikation vorgestellt wurden [BFG⁺00].

4.1.3 DIROKOL

Im von der Bayerischen Forschungsförderung geförderten Projekt DIROKOL (DienstleistungsROboter in KOstengünstiger Leichtbauweise) wurde das Ziel verfolgt, einen autonom agierenden Dienstleistungsroboter zu entwickeln, der im Besonderen für den Einsatz in einer Krankenhausumgebung konzipiert ist. Gerade in diesem Umfeld fallen eine Reihe von Routinetätigkeiten an, die den Schwestern und Pflegekräften durch den Einsatz eines Roboters abgenommen werden können, was diesen mehr Zeit für die individuelle Pflege und Betreuung der Patienten einräumt. Parallelen ergeben sich hier zum Robotersystem HELPMATE (siehe Abbildung 1.4), der ebenfalls für den klinischen Einsatz konzipiert wurde und bereits seit mehreren Jahren an verschiedenen Krankenhäusern der USA Botendienste übernimmt.

Beteiligte Partner waren neben der DLR Wessling (Prof. Hirzinger) als Koordinator die DASA (Bremen), InnoStrat (Selb), das Konsortium MayTec, KUHNKE und RoboSystem (Vierkirchen), der Lehrstuhl für Mustererkennung (LME) der Universität Erlangen-Nürnberg, Med-Inno-Tech (Regensburg), Sedlbauer AG (Grafenau), Technology Consultants (Kressbronn und München), der Lehrstuhl für Steuerungs- und Regelungstechnik der TU München sowie das Institut für Internationales Innovationsmanagement der Universität Bern.

Der entwickelte Prototyp ROMAN II, der in Abbildung 4.5 zu sehen ist, befindet sich bereits in der Neurologischen Klinik Bad Aibling im Testeinsatz. Eine Auswahl der im Rahmen dieses Vorhabens bearbeiteten Forschungsprojekte findet sich im Tagungsband zum 15. Fachgespräch Autonome Mobile Systeme (AMS) 1999 [SHF99].

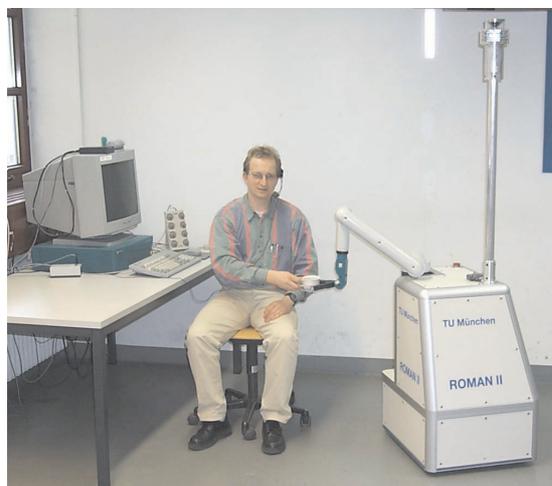


Abbildung 4.5: Prototyproboter ROMAN II, der im Rahmen des DIROKOL-Projektes entwickelt wurde.

Zur Navigation bzw. Selbstlokalisierung wird eine omnidirektionale Kamera eingesetzt, die in einer Höhe von 180cm montiert ist, mit deren Hilfe Landmarken (Türen, Fenster etc.) extrahiert werden, an denen sich der Roboter orientiert. Die eigentliche Hindernisvermeidung basiert auf den Messwerten von Ultraschall-Sensoren, die im Basisteil des Roboters integriert sind.

Neben dem mobilen Basisteil verfügt der Roboter über einen MANUS-Manipulatorarm, der ihm z.B. das Öffnen von Türen und die Manipulation von Objekten ermöglicht.

Die Instruierung des Roboters erfolgt über Spracheingabe. Auch hier kann wieder davon ausgegangen werden, dass die instruierenden Personen (Klinikpersonal und Patienten) das Sprachrepertoire kennen, welches der Roboter zu verarbeiten im Stande ist.

Visuell-basierte Verfahren zur Mensch-Maschine-Interaktion sind auch in diesem Szenario nicht zu finden.

4.1.4 Projekt MORPHA

Das Projekt MORPHA wurde aus der Leitprojektausschreibung des Bundesministeriums für Bildung und Forschung (BMB+F) unter dem Titel „Mensch-Technik-Interaktion in der Wissensgesellschaft“ initiiert und stellt das gegenwärtig wohl größte deutsche Konsortium im Bereich mobiler Service- und Assistenzsysteme dar. Aus Abbildung 4.6 sind die am Projekt MORPHA (Intelligente anthropomorphe Assistenzsysteme) beteiligten Partner ersichtlich. Die Beschreibung des Vorhabens orientiert sich auch hier an der Web-Präsenz des Projektes, wo auch die aktuellen Arbeiten zu finden sind, die im Rahmen des Projektes publiziert wurden und werden (<http://www.morpha.de>).



Abbildung 4.6: Partner im Projekt MORPHA.

Die Leitidee des Verbundvorhabens MORPHA (siehe auch [Lay00] [HNS01]) ist es, intelligente mechatronische Systeme, insbesondere Robotersysteme, mit leistungsstarken Kommunikations-, Interaktions- und Verhaltensmechanismen auszustatten. Diese sollen die Systeme befähigen, mit dem menschlichen Benutzer unter dessen Anleitung und Kontrolle zu kooperieren und ihm zu assistieren. Sie orientieren sich an der menschlichen Gestalt ebenso wie an den menschlichen Sinnen. Der Begriff „anthropomorph“ ist also weiter gefasst, als im üblichen Sprachgebrauch. Er soll als „mensenähnlich“ im weitesten Sinn verstanden werden und auch die menschlichen Sinne umfassen.

Neben der Sprache stellt die visuelle Kommunikation, beispielsweise über Gesten, ein wichtiges Thema des Vorhabens dar, wobei hier die in Abschnitt 2.4.2.2 diskutierten Aspekte des robusten und sinnvollen Einsatzes zu beachten sind. Diese Form der Kommunikation soll die Kontrolle und Kommandierung sowie die „Belehrung“ und „Programmierung“ der Assistenzsysteme erleichtern. Ein weiterer menschlicher Sinn, der in die Interaktion zwischen Mensch und

Maschine mit einbezogen werden soll, ist das „Fühlen“, also das Wahrnehmen und Reagieren auf Kräfte.

Zu den im Projekt anvisierten Produkten zählen neue Programmier- und Bedienkonsolen, Personenerkennungs- und intelligente Raumüberwachungssysteme bis hin zu intelligenten Steuerungen für unterschiedlichste mechatronische Systeme. Langfristig soll MORPHA den Weg ebnen für neuartige Assistenzsysteme in der Produktion ebenso wie im Bereich Haushalt und Pflege. Beispiele sind Handhabungs-, Montage-, Mobilitäts- und Reinigungshilfen oder mechatronische Assistenten im Bereich Rehabilitation.

Das Szenario Roboterassistent für Haushalt und Pflege zielt auf den Einsatz robotischer Assistenzsysteme in häuslichen Alltagsumgebungen (siehe Abbildung 4.7), wobei auch hier die in Abschnitt 4.1 diskutierten Aspekte maßgeblich sind.

Der Roboterassistent im Haushalt soll zum Teil gemeinsam mit dem Menschen einfache Hausarbeiten verrichten. Dies sind neben Hol- und Bringendiensten, Aufgaben wie das Tischdecken oder einfache Reinigungsaufgaben, was zum aktuellen Zeitpunkt als ausgesprochen ambitioniert angesehen werden muss.



Abbildung 4.7: Designstudie des MORPHA-Projektes zu einem Assistenten im Haushalt und Pflegebereich.

Die Interaktion mit dem Assistenzsystem soll zum einen zum Zweck der Kommandierung und Belehrung des Roboters erfolgen, zum anderen soll diese auch Möglichkeiten zur Leistungssteigerung des Gesamtsystems bieten. Ein Mindestmaß an Interaktionsfähigkeit soll den entwickelten Systemen ermöglichen, sich Hilfestellungen des Bedieners nutzbar zu machen, um so direkt ihre verfügbare Kompetenz zu erweitern. Voraussetzung dafür ist allerdings, dass die Kommunikation für den Bediener einigermaßen natürlich ist und von ihm somit nicht als Last empfunden wird.

Als natürliche Kommunikationskanäle werden vorrangig Sprache und Gestik angesehen. Es wird dementsprechend davon ausgegangen, dass die Interaktionspartner des Roboters sich zumindest über Sprache und Gesten äußern können, wobei hier wiederum die in Abschnitt 2.4.2.2 genannten Einschränkungen hinsichtlich Robustheit und sinnfälliger Anwendbarkeit gelten. Die Interaktion soll weiterhin über ein spezielles, multimodales Bediengerät laufen, wobei eine

Interaktion auch aus Nachbarräumen möglich wäre. Neben Sprache und Gesten wird in manchen Situationen auch eine direkte physische Interaktion stattfinden. Beispielsweise soll der Roboter zu Zwecken der Belehrung vom Bediener geführt werden können.

Als Fazit zum Assistenten im Haushalt und Pflegebereich im Rahmen des MORPHA-Projekts kann zusammengefasst werden, dass dieser insgesamt sehr visionär und ambitioniert angelegt ist und eine Reihe von Themen beinhaltet, die noch nicht unmittelbar von praktischer Relevanz sind, allerdings langfristig bedeutsam werden (siehe auch [BMB01]). Hervorgehoben werden sollte, dass hier explizit der Bedeutung visueller Informationen für den Aspekt der Mensch-Maschine-Kommunikation mit verschiedenen Forschungsthemen Rechnung getragen wird.

4.2 Weitere Assistenzsysteme

4.2.1 Produktionsassistent im MORPHA-Projekt

Durch mobile Assistenzsysteme in der Produktion (siehe Abbildung 4.8) soll eine neue Qualität in der Verbesserung von Produktionsprozessen hinsichtlich Produktivitätssteigerung und Humanisierung von Arbeitsplätzen erzielt werden. Produktionsassistenten sollen Aufgaben in Interaktion mit dem Menschen erledigen, d.h. den Menschen nicht ersetzen, sondern ihn unterstützen. Der Mensch übernimmt Kommandierungs-, Überwachungs- und Belehrungsfunktionen. Der Mensch wird in Fällen, in denen die Maschine nicht weiterweiß, zur Führung und weiteren Belehrung eingreifen und soll Partner im arbeitsteiligen Fertigungsprozess sein.



Abbildung 4.8: Designstudie des MORPHA-Projektes zu einem Produktionsassistenten.

In einem künftigen Produktionsszenario soll der angestrebte Produktionsassistent den Menschen bei Handlanger-, Transport- und Inspektionsaufgaben in einer typischen komplexen Fertigungsumgebung mit Werkzeugmaschinen, Förderbändern, Lagern usw. unterstützen. Eine typische Aufgabe ist das Greifen von Teilen aus Behältern, der Transport zu einer Werkzeugmaschine, zu einem Montagearbeitsplatz, das Assistieren bei der Montage oder der Transport

zu einem Inspektions- bzw. Messplatz.

Eine beispielhafte Aufgabenkette, die vom technischen System und dem Menschen arbeits- teilig bewältigt wird, kann folgendermaßen aussehen: Entnahme von komplexen Werkstücken aus Behältern; Transport des Werkstückes zu einer Bearbeitungsstation; Bereitstellen der Be- arbeitungswerkzeuge Realisierung mehrerer Bearbeitungsvorgänge; Montage, d.h. Fügen und Befestigen mehrerer Werkstücke; Abtransport des Werkstückes.

Um eine derartige Aufgabe in der sich oft ändernden Fabrikumgebung bei variablen Posi- tionen von verschiedenen Transportkisten, bei Werkstücken mit unterschiedlicher Gestalt und Gewicht in beliebigen Lagen in der Transportkiste und für unterschiedliche Werkzeugmaschi- nen bzw. Zielobjekte bewerkstelligen zu können, ist ein Höchstmaß an Flexibilität erforderlich. Diese Flexibilität soll hier vorrangig durch die Belehrung der Maschine in Interaktion mit dem Menschen erzielt werden. Dies soll erfolgen, indem der Mensch

- dem Produktionsassistenten die Fertigungsumgebung durch Herumführen und Benen- nung ausgezeichneter Plätze zeigt (siehe z.B. [KHH⁺01]), und die Maschine damit ein Modell der Umgebung als Basis für die freie, zielgerichtete Navigation für Transportauf- gaben lernt,
- dem Produktionsassistenten Objekte wie Transportkisten, Werkstücke und Werkzeugma- schinen zeigt und benennt, und die Maschine damit eine Repräsentation dieser Objekte lernt,
- dem Produktionsassistenten Bewegungen im sensorischen Kontext, z.B. zum Andocken an Transportkisten und zum Greifen von Teilen, vormacht, und die Maschine damit sen- somotorische Fähigkeiten (Skills) erwirbt,
- komplexe Operationsfolgen vormacht, und die Maschine damit Handlungspläne lernt, indem er der Maschine das Lernen der Kooperation beibringt, damit die Maschine die Intentionen des Menschen erkennen bzw. schätzen kann und diese in ihre Planungen und Handlungen mit einbezieht.

Diese interaktive, intuitive Belehrung der Maschine durch Zeigen und Vormachen für Funk- tionen wie Mobilität und Manipulation bildet die Basis für intelligentes Verhalten des Produk- tionsassistenten. Diese Belehrung muss sowohl vor als auch während des Betriebs (inkremen- telles Lernen) möglich sein.

Entscheidend am oben beschriebenen Szenario ist ein möglichst hoher Arbeitsanteil des tech- nischen Assistenten und dessen Fähigkeit, flexibel unterschiedliche Abfolgen unter variablen Randbedingungen zu bewältigen. Der Produktionsassistent muss dabei mindestens zeitweise autonom arbeiten, wodurch er sich von Handlinggeräten unterscheidet.

Sinngemäß gelten auch hier die bereits in Abschnitt 4.1.4 getroffenen zusammenfassenden Aussagen hinsichtlich Ambitioniertheit und der Erreichung unmittelbar praktisch einsetzbarer Verfahren.

4.2.2 HERMES

Mit dem Robotersystem HERMES (Abbildung 4.9) am Institut für Messtechnik der Universi- tät der Bundeswehr München wird das Ziel verfolgt, einen anthropomorphen Serviceroboter

zu entwickeln [Bis00a]. Anthropomorph bedeutet in diesem Projekt, dass der Roboter sowohl über natürliche Interaktionsfähigkeiten (Sehen, Sprache) verfügt als auch hinsichtlich seines Aktionsrepertoires möglichst menschenähnlich ausgelegt ist und man hier auch von einem humanoiden System sprechen kann.

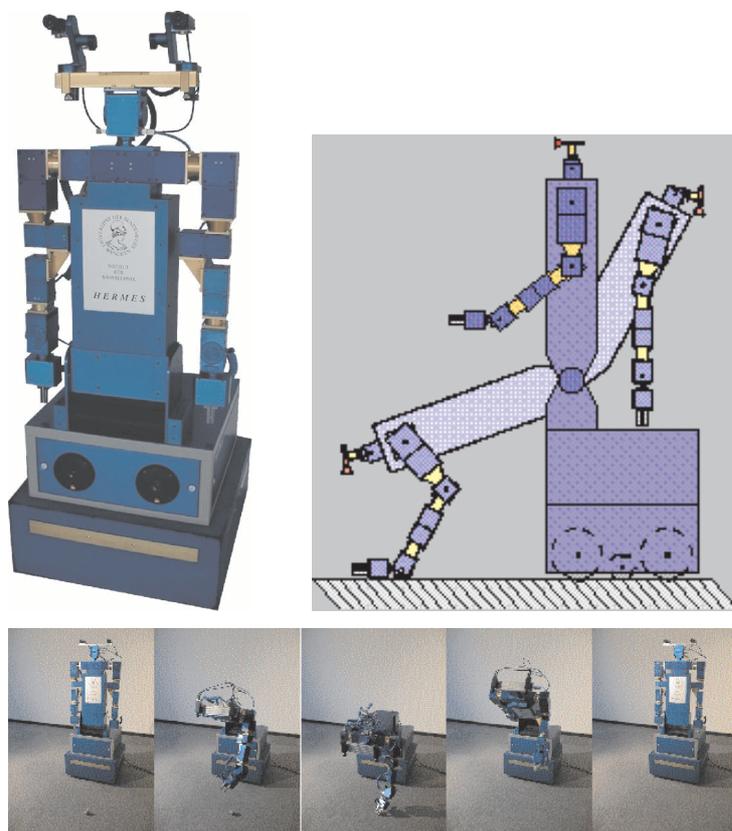


Abbildung 4.9: Serviceroboter HERMES.

Der Roboter verfügt über zwei Manipulatorarme und ein Stereokamerasystem, das als universeller Sensor für die Wahrnehmung der Umgebung des Roboters fungiert. Eine holonome Basisplattform gestattet das Drehen auf der Stelle. Weiterhin kann der Roboter seinen „Oberkörper“ neigen. Durch sein anthropomorphes Design ist der Roboter, der fast ausschließlich aus Basismodulen der Firma AMTEC aufgebaut wurde, in der Lage, seinen Arbeitsraum im Vergleich zu vielen anderen Robotersystemen deutlich zu erweitern.

Neben einer im Grundansatz verhaltensbasierten und in Abbildung 4.10 dargestellten Steuerarchitektur, die auf den grundlegenden Konzepten *Situation*, *Verhalten* und *Fertigkeiten (Skills)* basiert [BG99a], wurde und wird besonderer Wert auf die Mensch-Roboter-Interaktion gelegt, die integraler Bestandteil dieser Steuerarchitektur ist.

Während visuell-basierte Methoden für das Erkennen (und Greifen) von Objekten genutzt werden, erfolgt die Kommunikation mit dem Benutzer ausschließlich sprachbasiert über ein drahtloses Mikrofon bzw. ein Headset.

Hervorhebenswert erscheint der Ansatz, *Situationen* als Basiskonzept einer Steuerarchitektur zu definieren. Die Situationsbeschreibung beinhaltet den aktuellen Status aller verfügbaren Skills des Roboters und bildet diese auf die entsprechend der aktuellen Situation zu aktivierenden Verhalten ab. Wie dies genau geschieht, kann den verfügbaren Publikationen jedoch nicht im Detail entnommen werden. Jedoch erscheint es auf diese Weise möglich, einen sprachbasier-

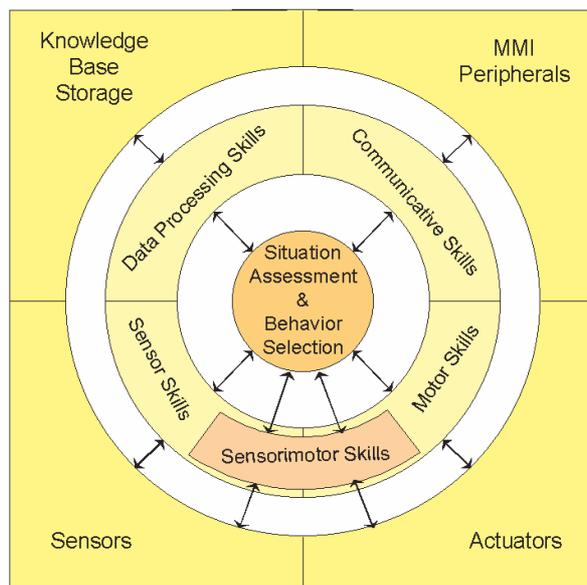


Abbildung 4.10: Systemarchitektur des Roboters HERMES, basierend auf den Konzepten Situation, Verhalten und Fertigkeiten (Skills) (nach [BG99a]).

ten Dialog relativ robust umzusetzen, da alle durch den Roboter wahrgenommenen Eingaben jeweils im Kontext der aktuellen Situation interpretiert werden können.

Für eine detaillierte Darstellung sei auf [Bis00b] verwiesen.

4.2.3 ARNOLD

Auch am Institut für Neuroinformatik der Ruhr-Universität Bochum wird mit dem Robotersystem ARNOLD (Abbildung 4.11) das Ziel verfolgt, ein sowohl hinsichtlich der menschlichen Wahrnehmungs- als auch Aktionsfähigkeiten anthropomorphes System zu entwickeln.

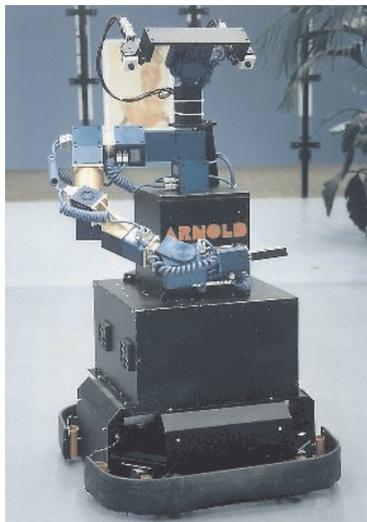


Abbildung 4.11: Robotersystem ARNOLD.

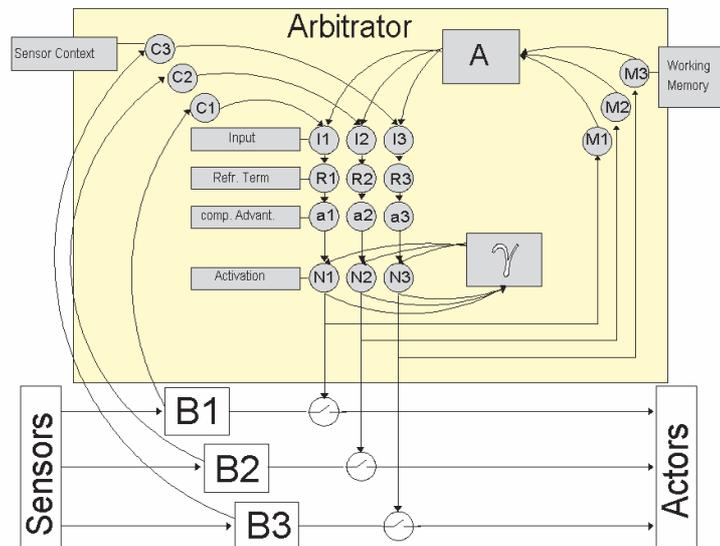


Abbildung 4.12: Architektur des Dynamic Approach (nach [BS98]).

ARNOLD verfügt im Gegensatz zu HERMES nur über einen Manipulatorarm (ebenfalls der Firma AMTEC) sowie ebenfalls über ein Stereokamerasystem, das auch in diesem Fall als universeller Sensor Verwendung findet.

Hervorhebenswert an diesem Projekt ist der durchgängig verwendete sogenannte *Dynamic Approach* zur Verhaltensgenerierung, der alle Teilverhalten des Roboters koordiniert [Ste97] [SB98] [BBD⁺99] (Abbildung 4.12), und der in ähnlicher Form sowohl in anderen Forschungsgruppen (siehe z.B. [BcJ⁺00] [LC99]) als auch im Rahmen der im Teil III der Arbeit beschriebenen eigenen Forschungsarbeiten eingesetzt wird (siehe Abschnitt 9.1).

Die Grundidee dieses verhaltensbasierten Ansatzes besteht darin, die aus der Aufgabenstellung definierten Elementarverhalten jeweils in Form eines dynamischen Systems zu implementieren. Die Elementarverhalten sind dabei nicht direkt untereinander, sondern lediglich über die Sensordaten (die Umwelt) miteinander verkoppelt. Über die Zeitkonstanten des Systems lassen sich Hierarchien verschiedener Elementarverhalten sowohl definieren als auch flexibel während der Laufzeit des Gesamtsystems anpassen.

4.2.4 COLIN

Am Lehrstuhl für Rechnerarchitektur der Universität Tübingen wurde in den vergangenen Jahren ein Gesamtsystem (Abbildung 4.13) entwickelt, welches dem Roboter COLIN (B21-Roboter mit Stereokamerasystem und Laserrangefinder) das Wahrnehmen und Verfolgen von Personen in unmodifizierten belebten Innenraumumgebungen gestatten soll (siehe [Fey00]).

Das entwickelte System besteht nach [Fey00] aus mehreren Komponenten. Die beiden Hauptkomponenten sind die Detektion und die Blicksteuerung auf der einen sowie die Navigation und Fahrsteuerung auf der anderen Seite. Beide Komponenten nutzen Basisdienste, die den Zugriff auf die Sensorik und Aktorik des Roboters gestatten.

Die Personendetektion basiert auf einer Kombination von Farb-, Bewegungs-, Kontur- und

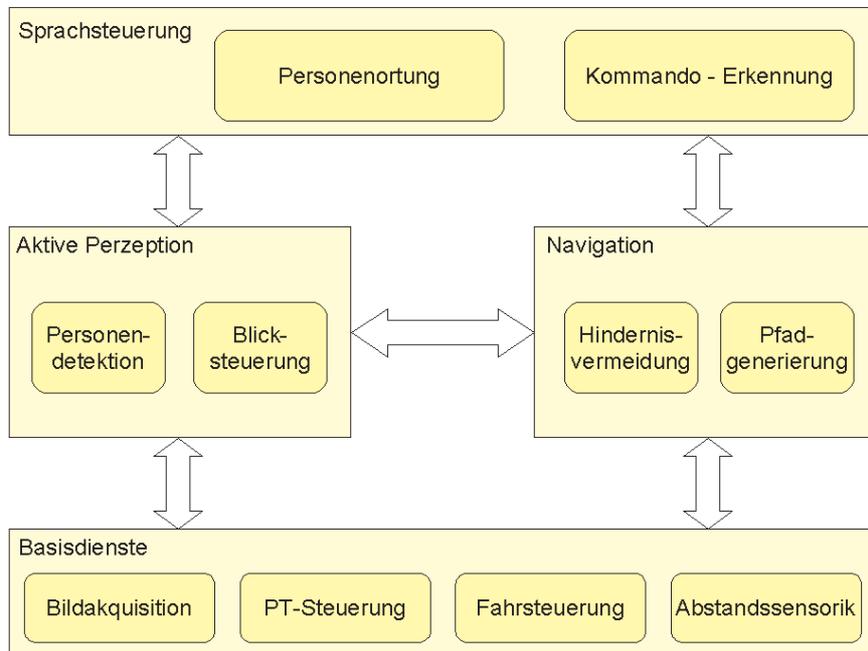


Abbildung 4.13: Aufbau des Gesamtsystems zur Personendetektion und zum Verfolgen von Personen (nach [Fey00]).



Abbildung 4.14: Ausschnitte aus einer Sequenz zum Personentracking (oben) sowie der Pfad einer Person (unten links), der der Roboter (unten rechts) unter Hindernisvermeidung folgt (entnommen der Webseite des Lehrstuhls für Rechnerarchitektur der Universität Tübingen).

Tiefeninformation. Letztere dient sowohl zur Ermittlung von Kopfbereichen als auch zur Lokalisation der Person im dreidimensionalen Szenenraum.

Aufgabe der Navigation ist es, einer wahrgenommenen Person zu folgen. Die Orientierung erfolgt dabei sowohl anhand des von der Person zurückgelegten Pfades als auch mit Hilfe einer abstandssensorbasierten Analyse der lokalen Umgebung. Stationäre und dynamische Hindernisse können somit umfahren werden.

Die beiden Hauptkomponenten sind nicht nur auf der logisch-strukturellen Ebene als zwei eigenständige Einheiten zu betrachten. Auch auf der Implementationsebene sind sie als zwei separate Prozesse realisiert, die über ein schnelles Interprozess-Kommunikationsprotokoll Informationen austauschen. Diese Entkopplung hat den Vorteil, dass die Kameranachführung und somit die Blickfixierung auch bei schnelleren Bewegungen der Person schritthaltend erfolgt, während die Navigationskomponente davon unabhängig die Auswertung der Abstandssensordaten und die Pfadplanung realisieren kann.

Der Gesamt Ablauf des Systems wird über eine Sprachsteuerung kontrolliert. Diese enthält eine Aufmerksamkeitskomponente zur akustischen Lokalisierung von Personen sowie eine einfache Spracherkennung zur Kommandierung des Roboters.

Abbildung 4.14 zeigt einige exemplarische Ergebnisse zur Personendetektion während einer Trackingsequenz sowie den Kurs der Person und den Kurs des Roboters, der dieser Person folgt.

Hervorhebenswert ist der konsequente Einsatz multimodaler Verfahren, die nach Einschätzung von FEYRER in [Fey00] erst die notwendige Robustheit unter nicht beeinflussbaren Umgebungsbedingungen sichern. Sowohl die Detektion als auch die Lokalisierung von Personen erfolgen sowohl visuell-basiert als auch mit Hilfe eines Laserscanners. Beide Komponenten stützen sich dabei nicht auf ein spezielles Merkmal, sondern verwenden wiederum eine Reihe verschiedener Verfahren.

Diese Herangehensweise findet sich ebenso in den im Teil III der Arbeit vorgestellten eigenen Forschungsarbeiten, die auf die Entwicklung eines mobilen interaktiven Serviceroboters als Informationsterminal im BauMarkt abzielen.

Kapitel 5

Resümee

Abschließend soll in diesem Kapitel der Frage nachgegangen werden, welche prinzipiellen Aussagen sich aus den vorgestellten Szenarien und Prototypen ableiten lassen und wo deren Bedeutung im Kontext interaktiver mobiler Serviceroboter zu sehen ist.

Mit den Tourguide-Projekten und hier insbesondere durch den Erfolg von RHINO und MINERVA gelang erstmals der Nachweis, dass eine robuste Navigation in weiträumigen Umgebungen auch über einen längeren Zeitraum möglich ist. Es konnte der Nachweis erbracht werden, dass mit der heute verfügbaren Sensorik und dem konsequenten Einsatz probabilistischer Modellierungstechniken alle Grundprobleme der Roboternavigation, also die Beantwortung der Fragen „Wo bin ich?“, „Wo befindet sich meine Zielposition?“ und „Wie komme ich von meiner aktuellen Position zu dieser Zielposition?“, prinzipiell lösbar sind. Auch für die Komplexe des Aufbaus konsistenter Umgebungsmodelle bei simultaner Zustandsschätzung (Selbstlokalisierung) und der globalen Lokalisation bei zwischenzeitlichem Verlust der eigenen Zustandsinformation (*Kidnapped Robot Problem*) stehen entsprechende methodische Rahmen zur Verfügung. Aspekte der intuitiven und natürlichen MRI spielten im Rahmen der Tourguide-Projekte noch eine eher nebenläufige Rolle. Weitere Gründe für den Einsatz eines Serviceroboters als Tourguide liegen sicherlich auch darin begründet, dass die Fähigkeiten zur autonomen Navigation und zur einfachen Mensch-Maschine-Kommunikation ausreichen, um ein solches System umzusetzen, und dass beispielsweise keine manipulatorischen Komponenten benötigt werden.

Bei Assistenzrobotern stellt sich die Situation etwas anders dar. Hier treten zunehmend die Aspekte der intuitiven und natürlichen MRI ebenso wie der Einsatz verschiedenster Manipulatoren (im Rahmen dieser Arbeit nicht betrachtet) in den Vordergrund. Folgerichtig wird insbesondere dem Problemkreis MRI zunehmend durch verschiedenste Forschungsprojekte, wie beispielsweise im Verbundprojekt MORPHA, mehr und mehr Aufmerksamkeit gewidmet. Dennoch muss zum gegenwärtigen Zeitpunkt resümiert werden, dass den Aspekten des Auffindens von potentiellen Interaktionspartnern und dem kontinuierlichen Kontakthalten zu diesen in allen genannten Projekten mit Ausnahme der Arbeiten an der Universität Tübingen (Abschnitt 4.2.4) noch zu wenig Bedeutung beigemessen wird. Diese zunächst einfach anmutenden Fähigkeiten, die in einer nicht beeinflussbaren und hinreichend komplexen Einsatzumgebung, wie sie ein Haushalt im Kontext der Nursebot-Szenarien darstellt, sicherlich schwierig zu realisieren sind, stellen jedoch eine Grundvoraussetzung für die Anwendung weiterer Verfahren zur intuitiven MRI, wie Gesten- und Spracherkennung, dar.

In diesem Zusammenhang wird beim Einsatz von interaktiven Assistenzrobotern zukünftig die intensive Nutzung des visuellen Interaktionskanals eine zentrale Rolle spielen. Um all die sehr visionären Ziele zu erreichen, wie sie zum Beispiel in MORPHA postuliert werden, benötigt der Roboter viel mehr Informationen über den Nutzer, die sich allein aus dem meist ausschließlich genutzten Laserscan, wie das z.B. bei den Robotern des IPA im Museum für Kommunikation in Berlin der Fall ist, nicht gewinnen lassen (siehe dazu auch die Diskussion in Abschnitt 2.4.2.2). Hier ist aus aktueller Sicht der größte Bedarf an geeigneten Methoden zu verzeichnen, wie dies bereits im Abschnitt 2.7 postuliert wurde.

Teil III

PERSES - ein Intelligenter Interaktiver Mobiler Shopping-Assistent

Kapitel 6

Das BauMarkt-Szenario

Nach einem kurzen Abriss der Vorgeschichte, die letztendlich zu dem konkreten Anwendungsszenario hinführt, wird im Abschnitt 6.4 zunächst die Experimentalplattform PERSES beschrieben, die die derzeitige Basis für den angestrebten interaktiven intelligenten Shopping-Assistenten bildet. Abschnitt 6.2 ist der Beschreibung der für den Einsatz des Roboters relevanten Umgebungseigenschaften gewidmet, in Abschnitt 6.3 werden anschließend die konkreten Serviceleistungen, die der Roboter vollbringen soll, definiert. Ausgehend von der angestrebten Servicefunktionalität und den Eigenschaften der Einsatzumgebung lassen sich dann die wesentlichen notwendigen Fähigkeiten des Serviceroboters hinsichtlich Navigation (Abschnitt 6.5), Mensch-Roboter-Interaktion (Abschnitt 6.6) und Steuerarchitektur bzw. Verhaltenskoordination (Abschnitt 6.7) ableiten.

Die nachfolgenden Kapitel widmen sich dann detailliert der Umsetzung dieser Fähigkeiten.

6.1 Vorgeschichte

Getragen von dem Gedanken, dass intelligentes Verhalten nur dann entwickelt werden kann, wenn ein körperlich existierendes System mit einer realen Umwelt interagiert (siehe [Chr96]), wurden am Fachgebiet Neuroinformatik zunächst verschiedene Robotersysteme für Experimentierzwecke und methodisch-konzeptionelle Untersuchungen angeschafft (Abbildung 6.1). Dabei handelt es sich um mehrere KHEPERA-Miniaturroboter, das MILVA-System (Multisensorielles intelligentes Vehikel in neuronaler Architektur), das über eine Dreiradkinematik verfügt, sowie um einen B21-Roboter PERSES (PERSONAL SERVICE SYSTEM, Real World Interfaces, USA), der für das BauMarkt-Szenario von unmittelbarer Relevanz ist.

Alle genannten Systeme dienen als Forschungsplattformen, um die Methodenentwicklung in den Bereichen der sensomotorischen Koordination und der aktiven Wahrnehmung im Action-Perception-Cycle voran zu treiben. Die Roboter stellen also intelligente, aktiv handelnde Agenten dar, wobei ein besonderer Schwerpunkt auf der Verarbeitung visueller Information liegt (Bildverarbeitung auf Rädern).

Während die Miniaturroboter in erster Linie für Forschungsaktivitäten im Bereich des Reinforcement-Lernens, für Prinzipuntersuchungen zur Roboternavigation sowie für Lehrzwecke eingesetzt werden, stand für die beiden „großen“ Systeme die Frage, wie sich die Arbeiten zur Roboternavigation, zur Verhaltensgenerierung und zur Mensch-Roboter-Interaktion in ein

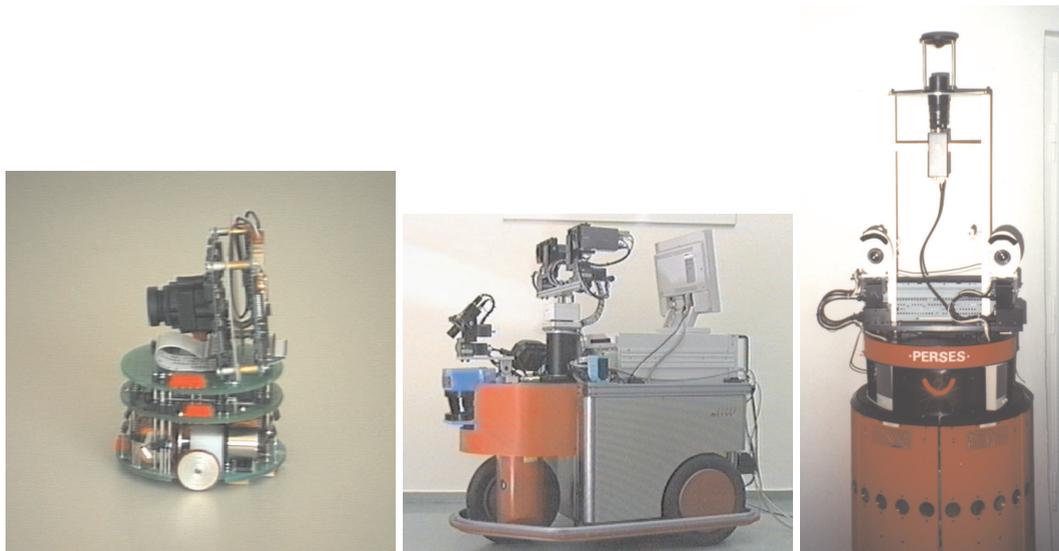


Abbildung 6.1: Roboter des Fachgebiets Neuroinformatik der TU Ilmenau: (v.l.n.r.) Miniaturroboter Khepera, Roboter MILVA und Roboter PERSES

schlüssiges Gesamtszenario einbetten lassen, das als einende Klammer alle genannten Forschungsgebiete in sich vereint. Ausgehend vom Aufbau und der technischen Ausstattung des Roboters MILVA entstand zunächst die Idee, den Einsatz eines solchen Roboters als intelligentes und interaktives Transportsystem, z.B. auf Bahnhöfen oder Flughäfen, als Anwendungsszenario zu wählen (siehe Abbildung 6.2).



Abbildung 6.2: Veranschaulichung des Einsatzes eines intelligenten Transportroboters auf einem Bahnhof: Fotomontage des Bahnhofs Ilmenau und der am Fachgebiet eingesetzten MILVA-Plattform

Nach eingehender Prüfung lagen jedoch die Nachteile dieses Szenarios als konzeptioneller Rahmen für Arbeiten zur MRI und robusten Navigation auf der Hand:

- Der Aufbau des Roboters bedingt die Beschränkung seiner Mobilität auf einen ebenen Untergrund. Treppen, Förderbänder und Stufen sind jedoch typisch für Bahnhöfe und Flughäfen, so dass davon ausgegangen werden musste, dass die Transportfunktion durch ein solches Systems nur eingeschränkt wahrgenommen werden kann.

- Die zumindest zeitweise extreme Überfüllung des Einsatzgebietes, z.B. unmittelbar nach Ankunft eines Zuges oder Flugzeuges, ließ die Navigationsproblematik extrem schwierig erscheinen, was sich zwangsläufig negativ auf die Funktionalität des Transportroboters auswirken würde.
- Hinsichtlich der Mensch-Roboter-Interaktion, die forschungsseitig eine tragende Säule am Fachgebiet darstellt, ergaben sich ebenfalls einige kritische Aspekte. Die Wahrnehmung und das kontinuierliche Kontakthalten zu einer bestimmten Person ist hier, wie auch im nachfolgend beschriebenen BauMarkt-Szenario, von essentieller Bedeutung. Allerdings erscheint die Realisierung dieser Funktion mittels der dem Roboter zur Verfügung stehenden Sensorik (Kameras, Entfernungsmesser) ohne weitere technische Hilfsmittel (z.B. Transponder) vor dem Hintergrund sehr vieler Personen in der unmittelbaren Umgebung des Roboters und hochgradig variabler Umgebungsbedingungen als derzeit nicht realisierbar.
- Ein ganz wesentlicher Kritikpunkt an diesem Anwendungsszenario ist nicht zuletzt auch die Tatsache, dass solche Systeme hier unmittelbar als „Jobkiller“ wirksam werden könnten.

Zunächst weiterhin der Idee eines vorwiegend für Transportaufgaben gedachten Systems folgend, wurde dessen Einsatz in Bau- bzw. Supermärkten in Erwägung gezogen. Insbesondere im BauMarkt fanden sich einige sehr interessante Aspekte, die den Einsatz eines solchen Systems sinnvoll erscheinen ließen:

- Nahezu jeder, der des öfteren einen BauMarkt aufsucht, hat die Erfahrung gemacht, dass er eigentlich genau weiß, welches Produkt er sucht, aber oft nicht weiß, wo es im Markt zu finden ist. Ärgerlich ist dabei meist, dass kein Fachberater in der Nähe ist, der weiterhelfen könnte. Genau an dieser Stelle könnte ein mobiles Servicesystem ansetzen und den Kunden auf dessen Wunsch zu einem bestimmten Marktbereich oder Artikel lotsen.
- Im Gegensatz zum Supermarkt, der typischerweise von Stammkunden frequentiert wird, die nach einer kurzen Zeit genau wissen, was wo zu finden ist, kommt im BauMarkt einer fachkundigen Beratung eine wesentlich größere Rolle zu. Hier stellte sich die Frage, in wie weit ein mobiler Serviceroboter die Beratung des Kunden als Dienstleistung des Marktes entscheidend verbessern könnte.
- Im Vergleich zum Bahnhof-Szenario oder auch zum Supermarkt rückte die Transportfunktion etwas in den Hintergrund, da sich im BauMarkt sehr viele große und vor allem sperrige Artikel finden, die sich mit einem solchen System nur schwer transportieren lassen. Z.B. würden weit über die Roboterbegrenzung hinausragende Gegenstände die Navigation des Systems entscheidend beeinträchtigen oder gänzlich unmöglich machen.

Nachdem die prinzipielle Entscheidung zugunsten des BauMarkt-Szenarios gefallen war, galt es nun, einen BauMarktbetreiber zu finden, der unsere Forschungsaktivitäten wohlwollend unterstützt und diese als potentieller Endanwender solcher Serviceroboter fachlich und organisatorisch begleitet. Letztlich konnte mit der toom BauMarkt GmbH ein Partner gewonnen werden,

der sowohl Marktareal für Vor-Ort-Experimente zur Verfügung stellt als auch als Diskussionspartner für die Klärung inhaltlicher Fragen und die gemeinsame Definition von geeigneten Serviceleistungen fungiert. Die in dieser Arbeit verwendete, etwas ungewöhnliche Schreibweise des Begriffs BauMarkt ist der Firmenbezeichnung der toom BauMarkt GmbH entlehnt und soll als kleine Reminiszenz an die gute und stets konstruktive Zusammenarbeit verstanden werden.

Nach dem kurzen Abriss zum BauMarkt-Szenarios folgt in den weiteren Abschnitten dieses Kapitels eine Charakterisierung des BauMarkt-Szenarios. Dabei wird die im Kapitel 2 des ersten Teils der Arbeit verwendete Abfolge, ergänzt um eine Charakterisierung der relevanten Umgebungseigenschaften, wieder aufgegriffen.

6.2 Typische Eigenschaften der BauMarktumgebung

Bei der Darstellung grundlegender Eigenschaften des gewählten Anwendungsszenarios geht es in erster Linie um die Aspekte, die für den Einsatz eines Serviceroboters als interaktiver intelligenter mobiler Shopping-Assistent vorrangig von Relevanz sind:

- Die Grundfläche der Einsatzumgebung weist eine typische Dimension von etwa 60×100 Metern auf. Dies hat unmittelbare Auswirkungen auf die Art und Weise der Umgebungsmodellierung.
- Die Einsatzumgebung weist hinsichtlich der folgenden Aspekte eine hohe Dynamik auf:
 - Durch die permanente Umgestaltung des Marktes kann nicht von einer statischen, einmal zu erstellenden Umgebungsrepräsentation (z.B. metrische, gridbasierte Karte) ausgegangen werden.
 - Die Beleuchtungsbedingungen sind ständigen Schwankungen unterworfen, was beim Entwurf visuell-basierter Verfahren zu berücksichtigen ist. Gebiete mit Tageslicht wechseln mit Gebieten, in denen ausschließlich direkte oder indirekte Kunstlichtbeleuchtung vorzufinden ist.
 - Wie nicht anders zu erwarten, stellt ein BauMarkt ein Areal dar, in dem sich ständig Personen aufhalten, deren Zahl abhängig von Jahreszeit, Wochentag und Tageszeit extremen Schwankungen unterliegt.
- Ein weiteres Charakteristikum der Einsatzumgebung stellt deren extrem vielfältige Struktur dar, die den Einsatz von Heuristiken bei der Modellierung deutlich erschwert.

Die o.g. Umgebungseigenschaften sollen durch die in Abbildung 6.3 illustriert und untermauert werden.

6.3 Szenariospezifische Serviceleistungen

Nachdem ein konkretes Anwendungsszenario herausgearbeitet werden konnte, lassen sich nun eine Reihe von Funktionen definieren, die der intelligente interaktive Shopping-Assistent selbst umsetzen soll oder die mit seiner Hilfe realisiert werden können.



Abbildung 6.3: Einige repräsentative Abbildungen aus der realen Einsatzumgebung: (v.l.n.r.) Blick durch den Mittelgang, der mehr als 100 Meter lang ist; Aktionsware, deren Anordnung sich wöchentlich bzw. täglich verändern kann; Situation im Eingangs- bzw. Kassenbereich; typische Hinderniskonfiguration, die bei ausschließlicher Verwendung von ebenenweise messenden Entfernungssensoren problematisch wäre.

Aus Sicht des Endanwenders wird mit dem Einsatz solcher Systeme das langfristige Ziel verfolgt, die Servicequalität, also die Dienstleistung am Kunden, entscheidend zu verbessern. Für die Festlegung der konkreten Aufgaben des Servicesystems spielen dabei sowohl Aspekte der technischen oder informationstechnischen Machbarkeit eine Rolle wie auch ganz entscheidend strategische Überlegungen zur prinzipiellen Orientierung des BauMarktes selbst. Aus diesem Grund wurde von Beginn an ein sehr enger Kontakt zum Endanwender hergestellt, damit hier von vornherein auf beiden Seiten Konsens hinsichtlich der anzustrebenden Servicefunktionalität besteht.

Als erste Servicefunktion des Shopping-Assistenten wird das Lotsen des BauMarktkunden zu einem gewünschten Marktbereich oder Artikelstandort definiert. Analysiert man den Arbeitsablauf eines Fachberaters im BauMarkt, so stellt man fest, dass sich bis zu 90% der Kundenanfragen auf den Standort eines gewünschten Artikels beziehen. Das bedeutet einerseits, dass diese Auskunft, in Verbindung mit dem Geleiten zu der entsprechenden Lokalität im Markt, sehr gut von einem technischen System übernommen werden könnte, andererseits würde der Fachberater damit gleichzeitig von eher stupiden Aufgaben entlastet und könnte sich so seiner eigentlichen Aufgabe, der kompetenten Beratung des Kunden, viel besser widmen. Das Lotsen des Kunden zu einem gewünschten Marktbereich oder Artikelstandort in Verbindung mit den dazu notwendigen Navigations- und Interaktionsfähigkeiten stellt gegenwärtig *den* zentralen Schwerpunkt der Arbeiten dar (siehe auch Kapitel 9). Aufbauend darauf lassen sich weitere Serviceleistungen ableiten, die im BauMarkt-Szenario sinnvoll einsetzbar sind, jedoch momentan nur auf Konzeptebene existieren.

Die eigentliche Beratungsfunktion, z.B. die Anwendung eines speziellen Werkzeuges oder der technologisch sinnvolle Ablauf einer Renovierungsmaßnahme, lässt sich durch den Shopping-Assistenten auch in absehbarer Zeit nicht zufriedenstellend realisieren. Dies liegt u.a. darin begründet,

- dass eine immens große Datenbank mit den entsprechenden Informationen vorgehalten werden müsste,
- dass eine hinreichend robuste, sprecher- und dialektunabhängige Sprach- oder zumindest

Texterkennung und -interpretation verfügbar sein müsste, da eine grafische Dialogführung aufgrund der zu erwartenden Vielfalt hier sicherlich ungeeignet ist,

- dass das System über dem menschlichen Fachberater vergleichbares Kontextwissen verfügen müsste, was sicherlich das entscheidende Defizit darstellen dürfte.

Angesichts der o.g. Aspekte wird deutlich, dass es mit dem Einsatz des Shopping-Assistenten im BauMarkt nicht darum geht, den Fachberater zu ersetzen, sondern vielmehr darum, dessen Einsatz zu effektivieren. Um dennoch eine Beratungsfunktion des Shopping-Assistenten gewährleisten zu können, wird perspektivisch die Einblendung eines menschlichen Fachberaters auf dem am Roboter verfügbaren Touchscreen-Display angestrebt. Dieser telepräsenste Fachberater, den der Kunde bei Bedarf anwählen kann, wäre somit ein robotischer Avatar, der den Fachberater immer an genau der Stelle im Markt verfügbar macht, an der er gerade gebraucht wird.

Der Definition von über die bisher genannten Serviceleistungen hinausgehenden Funktionen sind prinzipiell keine Grenzen gesetzt. So könnte der Shopping-Assistent Werbung abspielen, auf aktuelle Sonderangebote hinweisen, Montageanleitungen einblenden und vieles andere mehr. Limitierend wirkt hier weniger die technische Machbarkeit als vielmehr der begrenzte Umfang an Ressourcen, die dem Servicesystem zur Verfügung stehen. Weiterhin setzt die Umsetzung der eben genannten Servicefunktionen, ebenso wie die Telepräsenz des Fachberaters, eine entsprechende technische Infrastruktur im BauMarkt selbst voraus, die momentan noch nicht gegeben ist.

Nach einer kurzen Darstellung des Roboters selbst werden dann in den nachfolgenden Abschnitten die notwendigen Fähigkeiten des Shopping-Assistenten bezüglich Navigation, Mensch-Roboter-Interaktion und Verhaltenskoordination abgeleitet, die sich aus der angestrebten Servicefunktionalität und den Eigenschaften der Einsatzumgebung ergeben.

6.4 Die Experimentalplattform PERSES

Seit Mai 1999 steht dem Fachgebiet Neuroinformatik der TU Ilmenau ein Roboter des Typs B21 der Firma IS Robotics (RWI, USA) zur Verfügung, der an verschiedenen Forschungseinrichtungen als Experimentalsystem Verwendung findet. Der Roboter besitzt eine kreisrunde Form und verfügt über die in Abbildung 6.4 dargestellte technische Ausstattung. Charakteristisch für diesen Roboter ist das mögliche Drehen auf der Stelle, das nicht über einen differentiellen Antrieb realisiert wird, sondern durch gleichzeitige Lenkbewegungen aller vier Räder, wobei sich das Oberteil synchron mitdreht, so dass die beiden Frontkameras immer in Fahrtrichtung weisen. Neben den für Robotikanwendungen typischen Ultraschallsensoren, Stereokameras, Mikrofonen etc. besitzt PERSES eine zusätzliche omnidirektionale Kamera, die den Blick auf die gesamte Einsatzumgebung gestattet. Ein auf der Rückseite des Roboters befindlicher Touchscreen dient sowohl der Kommunikation mit der roboterinternen Hardware als auch zur eigentlichen Mensch-Roboter-Interaktion im Sinne des Anwendungsszenarios.

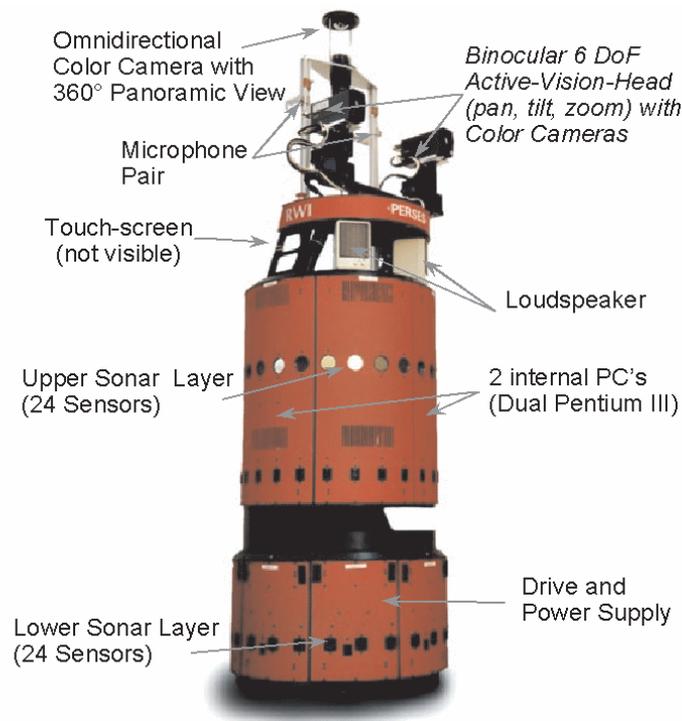


Abbildung 6.4: Roboter PERSES.

6.5 Navigation

Aus der angestrebten Servicefunktionalität des Shopping-Assistenten und den Eigenschaften der Einsatzumgebung ergeben sich nahezu zwangsläufig die Aspekte, die beim Entwurf und der anschließenden Realisierung eines robusten Navigationssystems zu berücksichtigen sind. Nachdem bereits im Abschnitt 2.2 ausgeführt wurde, dass Hindernisvermeidung und lokale Navigation für jedes mobile Servicesystem unerlässlich sind, besteht im vorliegenden Einsatzfall auch die Notwendigkeit der globalen Navigation mit den Teilkomplexen Umgebungsmodellierung, Selbstlokalisierung und Pfadplanung. Soll der Roboter den BauMarktkunden ausgehend vom aktuellen Standort zu einer bestimmten Zielposition führen, so muss der Roboter wahrnehmen können, wo er sich gerade befindet, wo sich die Zielposition befindet und wie er den Weg zum Ziel zu wählen hat. Dies setzt gleichzeitig das Vorhandensein eines Umweltmodells voraus, welches der Roboter im einfachsten Fall vorgegeben bekommt (z.B. anhand der Baupläne des Marktes) oder welches er selbständig erlernt und gegebenenfalls auch kontinuierlich aktualisiert.

Hinsichtlich der Hindernisvermeidung bestehen höchste Anforderungen an die Sicherheit beim Einsatz eines solchen Systems, andererseits erschweren die Gegebenheiten der Einsatzumgebung diese erheblich. Wie die Beispielszenen in Abbildung 6.3 verdeutlichen, gestalten sich die möglichen Hindernissituationen derart vielfältig, dass nahezu keine Heuristiken anwendbar sind. Die mit dem Roboter gegebene Sensorkonfiguration erschwert das Problem zusätzlich. Gegenstände, die sich ober- bzw. unterhalb einer der beiden Sonarebenen befinden, können

mit Hilfe dieser prinzipiell nicht wahrgenommen werden. Auch ein eventuell verfügbarer 2D-Laserscanner würde das Problem nur mildern, aber nicht generell entschärfen.

Aus diesen Gründen muss die Hindernisvermeidung auf visuelle Information zurückgreifen. Argumente für die von uns favorisierte visuelle Führung des Roboters sind neben der biologischen Vorbildtreue im Kontext der Weiterentwicklung neuroinformatischer Methoden, die hier jedoch nicht unbedingt im Vordergrund steht, die mittlerweile leistungsfähigen Kamera-Sensoren, die innerhalb eines großen Sichtbereiches Bilder mit hoher Ortsauflösung und großem Detailreichtum bereitstellen. Die erzeugten Bilder besitzen eine sehr hohe Informationsdichte und Strukturvielfalt und ermöglichen daher - je nach Aufgabe - die spätere Extraktion komplexer ortszeitlicher Informationen aus der dreidimensionalen Umwelt. Dem gegenüber stehen die harten Echtzeitanforderungen, die für eine robuste Navigation notwendig sind. Der Roboter muss den Bereich, den er befahren soll, vorher vollständig auf eventuell vorhandene Hindernisse untersucht haben. Geht man zusätzlich davon aus, dass ein Shopping-Assistent zumindest zügige Schrittgeschwindigkeit (ca. 1 m/s) erreichen sollte, so wird deutlich, welche Anforderungen an eine effiziente Bildverarbeitung hier bestehen.

Um die Hindernisvermeidung wirklich robust gestalten zu können, erscheint die parallele Nutzung aller dafür zur Verfügung stehenden sensorischen Informationen als das Mittel der Wahl. Optimal im Sinne der Informationsgewinnung wären hier redundanzfreie Sensorkanäle, zwangsläufig auftretende Redundanzen zwischen den verschiedenen sensorischen Kanälen stellen jedoch keineswegs einen Nachteil dar, sondern tragen vielmehr zur Erhöhung der Robustheit bei.

Die Notwendigkeit globaler Navigationsfähigkeiten wurde zu Beginn des Abschnitts bereits herausgearbeitet. Zunächst muss die Frage geklärt werden, wie ein geeignetes Umweltmodell für den Roboter aussehen könnte und wie dieses generiert werden kann. Dabei sind die mit dem Roboter angestrebten Servicefunktionen entsprechend mit einzubeziehen. Für die Wahl des Umweltmodells sind folgende Aspekte maßgeblich:

- Die Dimension des Einsatzfeldes spielt eine wesentliche Rolle, da dieses vollständig erfasst werden muss.
- Die notwendige örtliche Auflösung wird einerseits durch das angestrebte Anfahren eines bestimmten Artikelstandortes bestimmt und sollte deshalb minimal in der Größenordnung von etwa einem Meter liegen. Weiterhin zu berücksichtigen sind hier die konkreten Abmessungen des Roboters und die daraus resultierende minimale Breite von Durchfahrten.
- Die Umgebungsrepräsentation muss die zu erwartende Dynamik, hervorgerufen durch Personen und die ständig neue Anordnung verschiedenster Aufbauten, berücksichtigen können.
- Das Umweltmodell muss eine effektive Pfadplanung ermöglichen.

Für die Art und Weise der Generierung und Verwaltung des Umgebungsmodells stehen folgende Aspekte im Vordergrund:

- Zunächst muss wiederum die Dimension des Einsatzfeldes betrachtet werden. Diese impliziert, dass beispielsweise ein vollständig selbständiges Explorieren ungeeignet erscheint, weil dieses einerseits zu zeitaufwendig wäre und andererseits nicht sichergestellt

werden kann, dass der Roboter wirklich das gesamte für ihn zugängliche Marktareal kartiert hat.

- Der Roboter benötigt Information über den Standort der einzelnen Artikel oder Marktbe-
reiche, die einmal generierte Umgebungsrepräsentation muss also entsprechend attribu-
tiert werden.
- Veränderungen von Artikelstandorten oder Marktbereichen dürfen nicht zu Problemen
führen.

Selbstlokalisierung bedeutet, dass der Roboter seine aktuelle Pose, die stets Positions- und Orientierungsinformation beinhaltet, innerhalb seiner Einsatzumgebung bzw. innerhalb des Umgebungsmodells schätzt. Um dies zu realisieren, bestehen zwei prinzipielle Möglichkeiten:

- Zum einen versucht man, anhand von Informationen aus der Umgebung des Roboters, die eigene aktuelle Pose zu ermitteln. So orientieren sich landmarkenbasierte Verfahren an markanten Punkten der Umgebung. Solche markanten Punkte können auffällige und robust detektierbare Objekte, visuelle Ansichten oder auch typische Scanprofile sein, die beispielsweise mittels Referenzscan-Matching [WFJv00] ausgewertet werden.
- Bei Kenntnis der initialen Position des Roboters kann man auch versuchen, ausschließlich auf Basis der Odometriedaten, die anhand interner Sensorik angeben, wie sich der Roboter translatorisch und rotatorisch bewegt hat, die aktuelle Pose des Roboters zu ermitteln, ohne Information aus der Umgebung mit einzubeziehen. Dieses Verfahren, welches auch unter dem Namen Pfadintegration [FM00] bekannt ist, setzt allerdings eine sehr exakte Odometrie voraus, die bei realen Systemen in realen Umgebungen typischerweise nicht gegeben ist bzw. erfordert Verfahren, die eine Odometriekorrektur anhand extern erfasster Information ermöglichen.

Sinnvoll erscheint in jedem Fall die kombinierte Nutzung beider oben genannter Verfahren.

Für die Pfadplanung sind mittlerweile verschiedene Algorithmen entwickelt wurden, wobei der DIJKSTRA-Algorithmus (siehe z.B. [BKOS97]) eines der bekanntesten Verfahren darstellt. Gemein ist allen Pfadplanungsverfahren, dass sie einen möglichst optimalen Pfad innerhalb einer Graphenstruktur suchen, wobei die Optimalität von zu definierenden Kriterien, wie z.B. der Pfadlänge, abhängt (siehe auch Abschnitt 7.4).

Im Vergleich mit den in den Kapiteln 3 und 4 diskutierten Szenarien nimmt das BauMarkt-Szenario eine Art Zwischenstellung ein. Die zu erwartenden Veränderungen der Einsatzumgebung sind deutlich größer als bei einem Tourguide-Roboter in einem Museum, wo die Umgebung weitgehend unverändert bleibt. Beim Einsatz eines Roboters im Home-Care-Bereich dürfte diese Art der Umgebungsdynamik allerdings weitaus gravierender ausfallen, als das im BauMarkt gegeben ist. Hinsichtlich der Umgebungsdynamik, die durch Personen verursacht wird, weist das BauMarkt-Szenario im Vergleich zu einem Museum eine ähnliche Komplexität auf, wenn man die typischerweise gegebenen Platzverhältnisse und die Anzahl der zu erwartenden Personen betrachtet.

6.6 Mensch-Roboter-Interaktion

Der Shopping-Assistent soll perspektivisch als mobiler Informationskiosk im BauMarkt fungieren und durch seinen Einsatz die Qualität der Dienstleistung am Kunden entscheidend verbessern. Voraussetzung dafür ist, dass der Serviceroboter von den BauMarktkunden auch angenommen und benutzt wird. Betrachtet man die derzeit oft eingesetzten stationären Informationsterminals, die meist spezielle Produktinformationen präsentieren, so fällt auf, dass diese vom Großteil der BauMarktkunden gar nicht wahrgenommen bzw. sogar ignoriert werden.

Damit dem Serviceroboter nicht das gleiche traurige Schicksal beschieden ist, muss er über die Fähigkeit verfügen, Personen in seiner Einsatzumgebung wahrzunehmen, diese gegebenenfalls zu lokalisieren, sich der Person bzw. den Personen aktiv zuzuwenden und sie auf seine Serviceangebote aufmerksam zu machen. Der Serviceroboter soll sich also möglichst so verhalten, wie es der Benutzer auch von einem menschlichen Interaktionspartner erwarten würde. Hat sich ein Kunde dann auf dem System eingebucht, so sollte der Roboter zu diesem Kunden kontinuierlichen Kontakt halten und in der Lage sein, ihn auch nach einem zwischenzeitlichen Kontaktverlust wieder als *seinen* aktuellen Kunden zu identifizieren. In der relativ gering technisierten BauMarktumgebung und vor dem Hintergrund, mit beliebigen, nicht unterwiesenen Benutzern interagieren zu müssen, sind gerade diese Fähigkeiten notwendig, um ein intuitives und möglichst natürliches Gesamtverhalten des Servicesystems zu gewährleisten. Die Akzeptanz eines solchen Systems wird maßgeblich von der robusten Umsetzung der o.g. Teilleistungen abhängen, da die eigentliche Servicefunktionalität entscheidend durch den Aspekt der intuitiven Mensch-Maschine-Schnittstelle mitgetragen wird.

Für die Eingabe muss das Medium gewählt werden, welches für den konkreten Einsatzfall die wenigsten Probleme bereitet. Da eine Spracherkennung sowohl aus den in Abschnitt 2.4 dargelegten Gründen als auch wegen des im BauMarkt typischerweise herrschenden Geräuschpegels (Durchsagen, Unterhaltungsmusik etc.) derzeit nicht sinnvoll eingesetzt werden kann, muss hier auf den Einsatz klassischer Eingabegeräte, wie z.B. Touchscreen oder Tastatur, zurückgegriffen werden.

Als Ausgabemedien kommen sowohl grafische Ausgaben auf dem Display als auch Sprachausgaben in Frage. Während Ausgaben auf dem Display vorrangig für die unmittelbare Dialogführung, z.B. bei der Auswahl eines bestimmten Artikels, von Bedeutung sind, spielen Sprachausgaben auch für die Artikulation des Serviceroboters selbst eine wichtige Rolle. Für den menschlichen Interaktionspartner ist es enorm wichtig, dass er weiß, was der Roboter als nächstes zu tun gedenkt oder in welchem aktuellen Status sich der Roboter gerade befindet (z.B. ein Hinweis darauf, ob noch Kontakt zum Interaktionspartner besteht oder dieser gerade abgebrochen wurde). Ein Vorteil der Sprachausgabe ist weiterhin, dass diese von beliebigen Positionen aus wahrgenommen werden kann, während eine Ausgabe auf dem Display voraussetzt, dass der Interaktionspartner das Display von seiner aktuellen Position aus einsehen und ausgegebene Informationen auch erkennen kann. Dazu muss der Shopping-Assistent wiederum wissen, ob sich der Kunde ihm gerade zuwendet oder nicht.

Abbildung 6.5 fasst die für die Mensch-Roboter-Interaktion wesentlichen Aspekte in grafischer Form zusammen.

Vergleicht man die Anforderungen, die im BauMarkt-Szenario hinsichtlich der Mensch-Roboter-Interaktion bestehen, mit den in den vorangegangenen Kapiteln beschriebenen Tourguide- bzw. Home-Care-Szenarien, so erscheinen die folgenden Aspekte als wesentlich:



Abbildung 6.5: Grafische Illustration der notwendigen Interaktionsfähigkeiten des Shopping-Assistenten und der damit verbundenen Verhaltensleistungen (v.o.l.n.u.r.): Wahrnehmung eines potentiellen Kunden und Zuwendung zu diesem; Lotsen des Kunden zu einem bestimmten Standort; Begleiten des Kunden während der Einkaufstour; kontinuierliches Kontakthalten während des gesamten Interaktionszyklus.

- Die Interaktion soll mit beliebigen Personen und dennoch zumindest temporär mit einer bestimmten Person erfolgen. Damit sind hier eher Parallelen zum Tourguide gegeben.
- Die Interaktion mit nicht unterwiesenen Personen erschwert den Einsatz von Interaktionsmodalitäten, die bei einer Personal-Robot-Anwendung im Home-Care-Bereich prinzipiell in Frage kommen (wie z.B. Spracherkennung oder Gesteninterpretation), im BauMarkt-Szenario, da notwendige Vorkenntnisse auf der Seite der Benutzer des Systems nicht vorausgesetzt werden können.
- Von einem Roboter in einem Museum kann vielleicht noch intuitiv eine Tourguide-Funktion erwartet werden. Aber was soll ein Roboter im BauMarkt? Diese Überlegung führt dazu, dass der Shopping-Assistent möglichst selbsterklärend agieren sollte. Dazu gehört in erster Linie auch, dass er die Kunden des Marktes, die er zu kontaktieren versucht, über seine Funktion im BauMarkt informiert.
- Letztlich bietet das BauMarkt-Szenario eine nahezu einmalige Chance, alle o.g. Aspekte der MRI in realen Feldtests zu untersuchen. Dies unterscheidet den Shopping-Assistenten von den Projekten im Servicerobotikbereich, die vorrangig in Laborumgebungen durchgeführt werden.

6.7 Steuerarchitektur und Verhaltenskoordination

Abbildung 6.6 gibt einen schematischen Überblick über einen Interaktionszyklus, wie er während eines BauMarktbesuchs typischerweise ablaufen könnte.

Anhand von Abbildung 6.6 wird deutlich, dass Servicefunktionalität, Navigation und Mensch-Roboter-Interaktion eine untrennbare Einheit bilden müssen. Um einen sinnfälligen und mög-

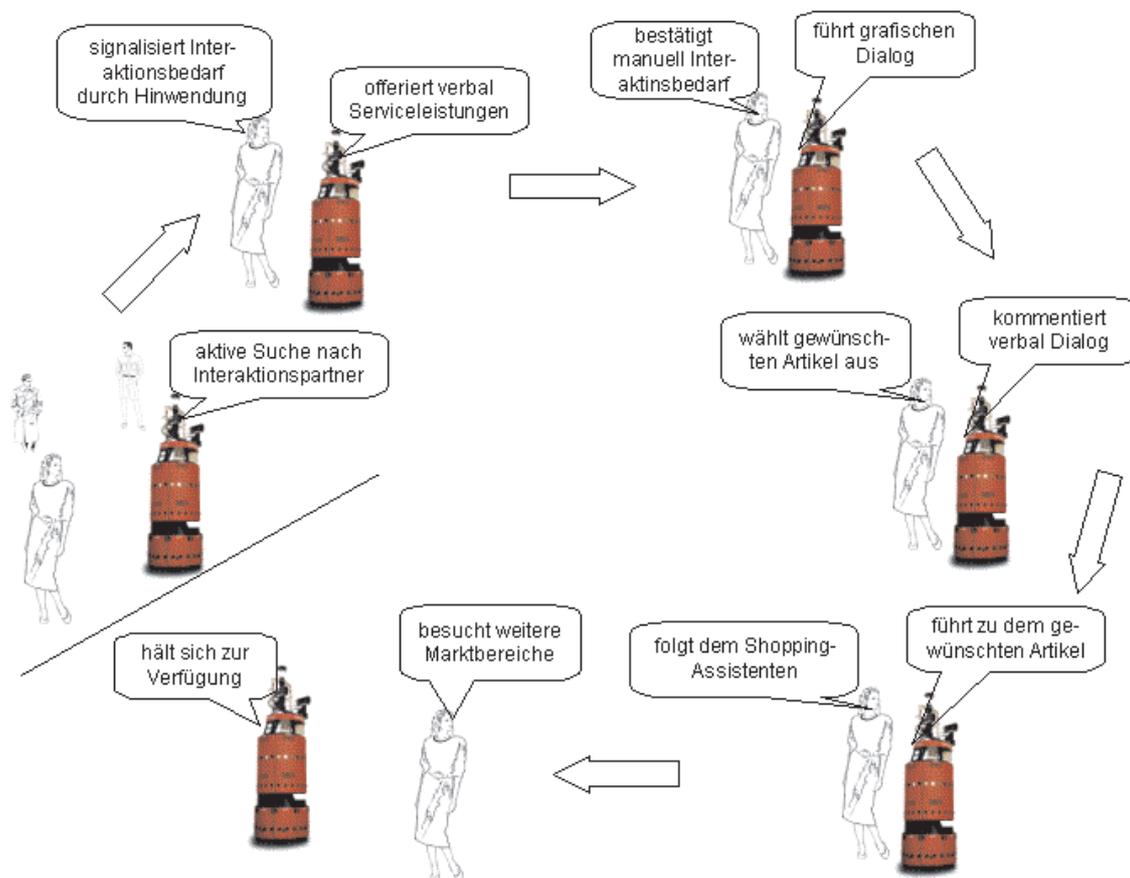


Abbildung 6.6: Typischer Interaktionszyklus beim Einkauf im BauMarkt.

lichst reibungslosen Ablauf eines solchen Interaktionszyklusses gewährleisten zu können, werden Methoden benötigt, die das Wechselspiel der verschiedenen Teilmodule situationsabhängig steuern. An einem einfachen Beispiel soll dies illustriert werden: Für die Lokalisation von Personen kommen sowohl visuell- als auch akustisch-basierte Verfahren zum Einsatz. Wurde eine Person als potentieller Interaktionspartner detektiert, sollte sich das Servicesystem nicht von weiteren akustischen Ereignissen ablenken lassen.

Der aus der Abbildung 6.6 ableitbaren Komplexität des Gesamtverhaltens des Shopping-Assistenten wurde zunächst dadurch Rechnung getragen, dass eine Aufteilung des Gesamtverhaltens in Teilverhalten erfolgte, die zunächst mehr oder weniger unabhängig voneinander umgesetzt werden können. Anschließend wird versucht, diese Teilverhalten so zu koordinieren, dass ein sinnvolles, an die jeweilige Situation angepasstes Gesamtverhalten erzielt werden kann. Entsprechend der Diskussion zu generellen Steuerstrategien in Abschnitt 2.3 wird die Steuerarchitektur dann möglichst optimal an den Erfordernissen des konkreten Einsatzszenarios ausgerichtet (siehe Kapitel 9) und sowohl Komponenten eines typischen verhaltensbasierten Systems [Ark98, KBM98] als auch Komponenten planender Systeme in einer hybriden Architektur integrieren.

6.8 Zusammenfassung

In diesem Kapitel wurde das BauMarkt-Szenario für den Einsatz eines intelligenten interaktiven Shopping-Assistenten herausgearbeitet. Anhand der typischen Eigenschaften dieses Szenarios und der angestrebten Servicefunktionalität konnten dann die wesentlichen Anforderungen für die Teilkomplexe Navigation, Mensch-Maschine-Interaktion und Verhaltenskoordination abgeleitet werden.

Die folgenden Kapitel beschreiben die konkrete Umsetzung der verschiedenen Teilleistungen des interaktiven mobilen Shopping-Assistenten PERSES. Kapitel 7 widmet sich zunächst dem Gesamtkomplex Navigation, der eine elementare Voraussetzung für die Gesamtfunktionalität des Shopping-Assistenten darstellt. Dies umfasst sowohl die Frage der robusten Hinderungsvermeidung als auch die globale Navigation mit den Aspekten Kartenaufbau, Selbstlokalisierung und Pfadplanung. In Kapitel 8 werden die Teilmodule beschrieben, die eine intuitive und möglichst natürliche Mensch-Maschine-Interaktion sicherstellen sollen. Diese Teilmodule beinhalten aufgrund der Multimodalität der Interaktion visuelle, scan-basierte, akustische und grafische Methoden. Der Problembereich Steuerarchitektur und Verhaltenskoordination wird in Kapitel 9 abgehandelt, das sich auch der Umsetzung erster szenariospezifischer Serviceleistungen widmet, wobei die in den vorangegangenen Kapiteln beschriebenen Teilleistungen die Voraussetzung für deren Umsetzung bilden.

Kapitel 7

Navigation in der BauMarkt-Umgebung

Es wurde bereits mehrfach betont, dass alle zu realisierenden Teilkomponenten eines Service-roboters vor dem Hintergrund der Service- bzw. Gesamtaufgabe eine untrennbare Einheit bilden. In diesem Kontext sind auch die notwendigen Fähigkeiten des Roboters zur Navigation zu sehen. Oberflächlich betrachtet erscheinen die Problemkreise Navigation und MRI zunächst relativ unabhängig voneinander zu sein. Bei genauerer Analyse stellt man jedoch sehr schnell fest, dass dem bei weitem nicht so ist. Ein einfaches Beispiel soll dies illustrieren: Soll der Roboter während der Fahrt zu einer benutzerdefinierten Zielposition kontinuierlich überprüfen, ob ihm der Kunde überhaupt noch folgt, so setzt das voraus, dass das Navigationsverhalten vom aktuellen Trackingstatus abhängen muss. Weiterhin sollte der Roboter bei drohendem Kontaktverlust in der Lage sein, seine Fahrt zur ursprünglich anvisierten Zielposition zu unterbrechen, um gegebenenfalls dem aktuellen Kunden hinterherfahren.

Für die konkrete Umsetzung der verschiedenen notwendigen Navigationsleistungen wurde zunächst in erster Linie der Versuch unternommen, bekannte Ansätze und Verfahren für den konkret vorliegenden Einsatzfall zu übernehmen und entsprechend den konkreten Gegebenheiten der Einsatzumgebung anzupassen. Dies betrifft die in Abschnitt 7.1.1 skizzierte Art der Umgebungsmodellierung in Form von Belegtheitsgitterkarten, den Algorithmus zum Aufbau einer probabilistischen globalen Umgebungskarte (Abschnitt 7.2), sowie die Pfadplanung und Bewegungssteuerung (Abschnitt 7.4).

Die Charakteristik der Einsatzumgebung erlaubte einerseits, den eigentlichen Kartenaufbau mit Hilfe eines visuell-basierten Verfahrens zur Odometriekorrektur (Abschnitt 7.2.1) deutlich zu vereinfachen, andererseits resultierte aus dieser Charakteristik auch die Notwendigkeit für die Entwicklung neuer Verfahren zur visuell-basierten Hindernisvermeidung (Abschnitt 7.1.3) und zur visuell-basierten Selbstlokalisierung (Abschnitt 7.3).

Die nachfolgenden Abschnitte sind der ausführlichen Beschreibung der einzelnen Teilsysteme zur Roboternavigation gewidmet. Um mit der üblichen Terminologie konform zu gehen, wird fortan der Begriff der Roboterpose verwendet, der sowohl die Position als auch die Orientierung des Roboters umfasst. Soweit Methoden verwendet wurden, die dem allgemeinen aktuellen Kenntnisstand zuzuordnen sind, wird auf die entsprechenden Publikationen bzw. auf einschlägige Standardliteratur verwiesen. Darüber hinaus gehende Erweiterungen oder ausschließlich in der Arbeitsgruppe entwickelte Verfahren und Algorithmen werden entsprechend

detaillierter behandelt, um dem Leser stets eine möglichst gute Nachvollziehbarkeit zu garantieren.

7.1 Hindernisvermeidung und lokale Umgebungskarte

7.1.1 Basisrepräsentation zur Umgebungsmodellierung

Als Basisrepräsentation für die Umgebungsmodellierung wurde das in [Elf89] vorgestellte Konzept der Belegtheitsgitterkarte (occupancy grid) gewählt. Ausschlaggebend dafür war die einfache und intuitive Handhabung dieser Datenstruktur einerseits und deren erfolgreicher Einsatz auch in Umgebungen großer Dimension andererseits, wie dies beispielsweise in [Thr99] publiziert wurde. Die Vorteile dieser Art der Umgebungsrepräsentation, die eine Projektion der dreidimensionalen Hindernissituation auf eine zweidimensionale Karte der Umgebung darstellt, liegen nach FOX [Fox98] in folgenden Punkten:

- Sie gestattet eine akkurate Bestimmung der Position des Roboters (abhängig von der Gitterauflösung).
- Es werden keinerlei Annahmen bezüglich der Geometrie der Umgebung gemacht, was in der BauMarkt-Umgebung ohnehin sehr schwierig wäre.
- Durch die Integration über die Sensordaten (Abstandswerte) können beliebige Umgebungsgeometrien modelliert werden (siehe auch Abschnitt 7.2).

Abbildung 7.1 veranschaulicht das Grundprinzip der Belegtheitsgitterkarte. Es erfolgt eine (hier egozentrische) metrische, zweidimensionale Repräsentation der Umgebung in Form eines Gitters, wobei für jede Gitterzelle vermerkt wird, ob diese Zelle durch ein Hindernis belegt (okkupiert) wird (Belegtheitswert 1) oder frei ist (Belegtheitswert 0). Infolge der auftretenden Unsicherheit bezüglich der Wahrnehmung und der Eigenbewegung des Roboters verwendet man typischerweise Belegtheitswahrscheinlichkeiten, die im Wertebereich von Null bis Eins liegen können. Die Auflösung des Gitters (Größe der Zellen) wird entsprechend der Charakteristik der Einsatzumgebung und abhängig von der für die Lösung der Gesamtaufgabe notwendigen örtlichen Auflösung festgelegt.

Analysiert man die einschlägige Literatur, so stellt man fest, dass Belegtheitsgitterkarten nahezu ausschließlich auf ebenenweise gewonnenen Entfernungsprofilen, die mittels Ultraschall- oder Lasersensoren gewonnen werden, beruhen. Prinzipiell können jedoch auch Kameras als Sensorbasis verwendet werden, wenn es gelingt, aus den aufgenommenen Bildern die Entfernung zum nächsten Hindernis in einer bestimmten Richtung zu ermitteln (siehe dazu auch Abschnitt 7.1.3).

7.1.2 Ultraschall-basiertes Verfahren zur Hinderniswahrnehmung

Zunächst benötigt der Roboter einen Eindruck von den Gegebenheiten in seiner unmittelbaren Umgebung, welcher zunächst als lokale (egozentrische) Belegtheitsgitterkarte auf der Basis der Ultraschallsensoren realisiert wird, wie dies z.B. auch in [BFHS96] vorgeschlagen wurde.

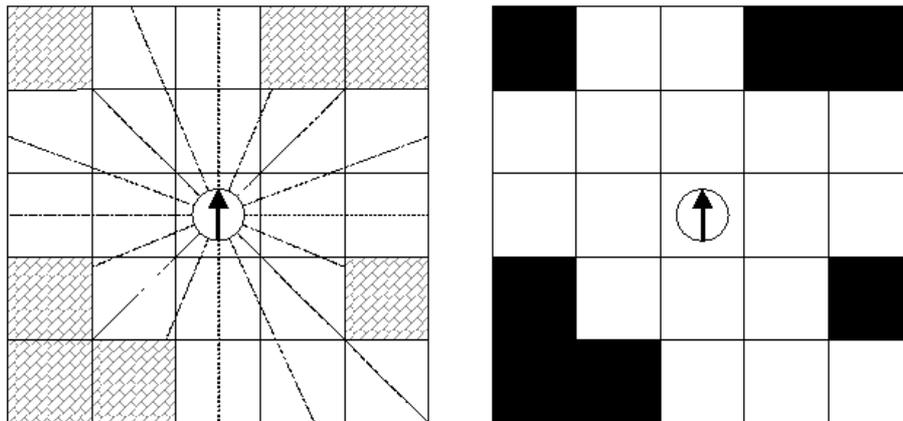


Abbildung 7.1: Veranschaulichung einer Belegtheitsgitterkarte. Links dargestellt ist die reale Hindernissituation, wie sie vom Roboter, dessen Position und Orientierung durch Kreis und Pfeil gekennzeichnet sind, mit seinen Ultraschallsensoren wahrgenommen wird, wobei hier idealerweise die Begrenzungen der Hindernisse entlang der Gitterbegrenzungen verlaufen. Rechts ist die resultierende (ideale) lokale Umgebungskarte zu sehen, in der Weiß Freiraum (Belegtheitswert 0) und Schwarz Hindernis (Belegtheitswert 1) repräsentieren.

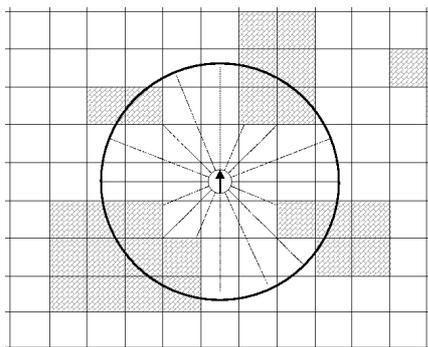


Abbildung 7.2: Veranschaulichung der lokalen Belegtheitsgitterkarte. Der Kreis mit einem festgelegten Radius von 5 m markiert den sensorischen Horizont des Roboters.

Um die Belegtheitswerte der Zellen möglichst robust zu ermitteln, wird ausgehend von der aktuellen Roboterpose die Belegtheitswahrscheinlichkeit jeder Zelle innerhalb des sensorischen Horizonts mit Hilfe eines Neuronalen Netzes ermittelt (siehe Abbildungen 7.3 und 7.4, analog wie dies in [Thr99] vorgeschlagen wird), was dem Erlernen eines inversen Sensormodells entspricht.

Den Eingang für das Netzwerk bilden neben der Position der betrachteten Zelle (relativ zum Roboter, kodiert in Polarkoordinaten) die jeweils vier in Richtung dieser Zelle ermittelten Abstandswerte. Am Netzwerkausgang soll dann die Belegtheitswahrscheinlichkeit für die betrachtete Zelle erscheinen. Um das Netzwerk trainieren zu können, werden entsprechende Trainingsdaten benötigt, die mit Hilfe eines Robotersimulators gewonnen werden, da nur so der tatsächliche Belegtheitswert der betrachteten Zelle als Teach-Signal auf einfache Weise ermittelt werden kann. Entscheidend für die Qualität der gewonnenen Trainingsdaten ist, dass die realen Sensoreigenschaften möglichst gut im Simulator berücksichtigt wurden. Um möglichst alle real auftretenden Situationen abdecken zu können (Generalisierungseigenschaften

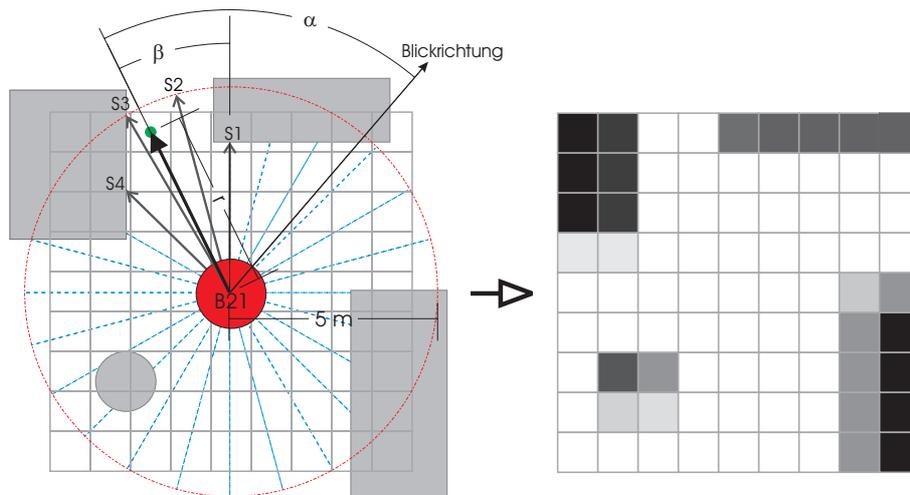


Abbildung 7.3: Entstehung der lokalen Belegtheitskarte. Die vom Roboter erfassten Sensorwerte werden auf Belegtheitswerte der Zellen der lokalen Umgebung des Roboters abgebildet. Unterschiedliche Grauwerte kodieren unterschiedliche Belegtheitswahrscheinlichkeiten. Schwarz entspricht 1 (belegt), Weiss entspricht 0 (frei bzw. nicht belegt). In die Schätzung des Belegtheitswertes gehen die vier Messwerte ein, die benachbart zur Richtung der betrachteten Zelle liegen. Die Position der betrachteten Zelle relativ zum Roboter ist durch den Abstand r und den Winkel β zwischen der Richtung zur Zelle und dem ersten betrachteten Sensor gegeben. Der Winkel α gibt die Richtung der Zelle bezogen auf die aktuelle Orientierung des Roboters an (nach [Lie00]).

des Netzwerkes), sollte darauf geachtet werden, dass die im Simulator verwendeten Hinderniskonfigurationen möglichst vielfältig sind. Ein weiterer Vorteil bei der Verwendung eines Simulators ergibt sich durch die Tatsache, dass die zum Training verwendeten Daten nicht zwischengespeichert werden müssen, sondern unmittelbar während der Bewegung des Roboters durch die simulierte Umwelt gewonnen werden können. Abbildung 7.4 zeigt die Topologie des MLP-Netzwerkes zur Schätzung der Belegtheitswahrscheinlichkeiten.

7.1.3 Visuell-basiertes Verfahren zur Hinderniswahrnehmung

Die alleinige Verwendung der Ultraschallsensoren für die Navigation erweist sich jedoch aus den bereits in Abschnitt 6.5 genannten Gründen als nicht ausreichend, da die im BauMarkt auftretenden, sehr vielgestaltigen Hinderniskonfigurationen nicht robust genug wahrgenommen bzw. erfasst werden können. Dies wirkt sich sowohl auf den lokalen / globalen Kartenaufbau als auch insbesondere auf die eigentliche Hindernisvermeidung aus. Deshalb wurde und wird versucht, visuelle Verfahren in die Umgebungsrepräsentation mit einzubeziehen. Einen sehr einfachen und effizienten Ansatz stellte beispielsweise HORSWILL in [Hor94b] bereits 1994 vor, der lediglich Grauwertgradienten in Fahrtrichtung des Roboters auswertet und hohe Gradienten als Indiz für den Übergang vom Fußboden zu einem Hindernis annimmt. Allerdings beruht dieses Verfahren, das z.B. auch in [Bis00b] Anwendung findet, auf stark vereinfachenden Annahmen bezüglich der Geometrie der auftretenden Hindernisse und der Gestaltung des

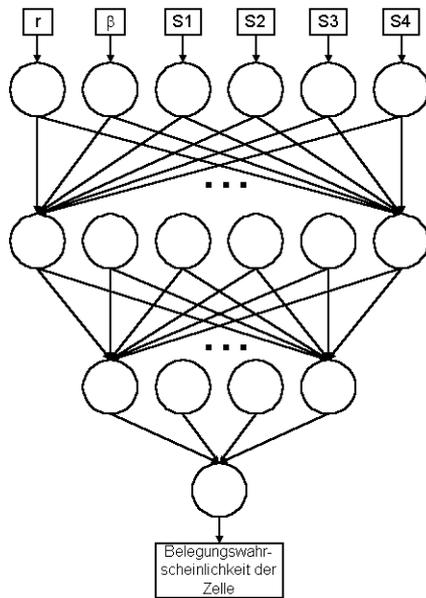


Abbildung 7.4: Topologie des MLP-Netzwerkes zur Schätzung der Belegungswahrscheinlichkeiten. Den Netzwerkeingang bilden die Polarkoordinaten (r und β) der betrachteten Zelle bezüglich des Roboters sowie die vier in Richtung dieser Zelle gewonnenen Sonar-Messwerte ($S1$ bis $S4$).

Fußbodens, was seine universelle Verwendung stark einschränkt.

Der in dieser Arbeit verwendete Grundansatz, der im Rahmen einer Diplomarbeit [Web98] entwickelt und in [KWBG98] erstmalig publiziert wurde, ist in Abbildung 7.5 dargestellt. Er beruht auf dem Prinzip der invers-perspektivischen Kartierung [Boh94] [MLB91] mit dem Ziel, anhand einer Sequenz einkommender Bilder eine Segmentierung in befahrbaren Freiraum (Untergrund) und Hindernisse zu realisieren (siehe auch [BM00] [BM01b]). Um die Umgebung des Roboters in Fahrtrichtung zu erfassen, wird die rechte der beiden Frontalkameras um ca. 65° nach unten geneigt.

Den Ausgangspunkt bildet zunächst die invers-perspektivische Transformation der einkommenden Bilder (Block (a) in Abbildung 7.5). Mit Hilfe dieser Transformation werden die perspektivischen Verzerrungen in der Bildebene bezüglich einer zweidimensionalen Bezugsebene (der Bewegungsebene des Roboters) eliminiert. Dadurch entsteht eine virtuelle Draufsicht auf diese Bezugsebene. Die Transformation ist nur dann korrekt, wenn alle Punkte des Bildes in dieser Bezugsebene liegen. Abbildung 7.6 zeigt ein Beispiel für die Transformation, in dem diese Voraussetzung erfüllt ist.

Bewegt sich der Roboter auf der Bezugsebene, so wird das invers-perspektivisch transformierte Bild um den Betrag dieser Bewegung verschoben (bewegungskompensiert) und mit dem transformierten Bild vor Ausführung der Bewegung verglichen. Liegen alle Bildpunkte in der Bezugsebene, so ergeben sich im Idealfall keine Differenzen. Ist diese Annahme verletzt, so ergeben sich für die entsprechenden Bildbereiche entsprechend Differenzen, die auf dort vorhandene Hindernisse hindeuten.

Die Ermittlung solcher Differenzen erfolgt mit zwei unterschiedlichen Verfahren. Der optische Fluss (Block (b) in Abbildung 7.5) liefert sehr genaue Information über die tatsächliche Verschiebung einzelner Bildbereiche, während die Differenz- bzw. Strukturanalyse (Block (c) in Abbildung 7.5) diese Verschiebungen mehr qualitativ erfasst. Die Ergebnisse beider Module werden zur Gewinnung eines farbbasierten Merkmalsmodells (Block (d) in Abbildung 7.5) ge-

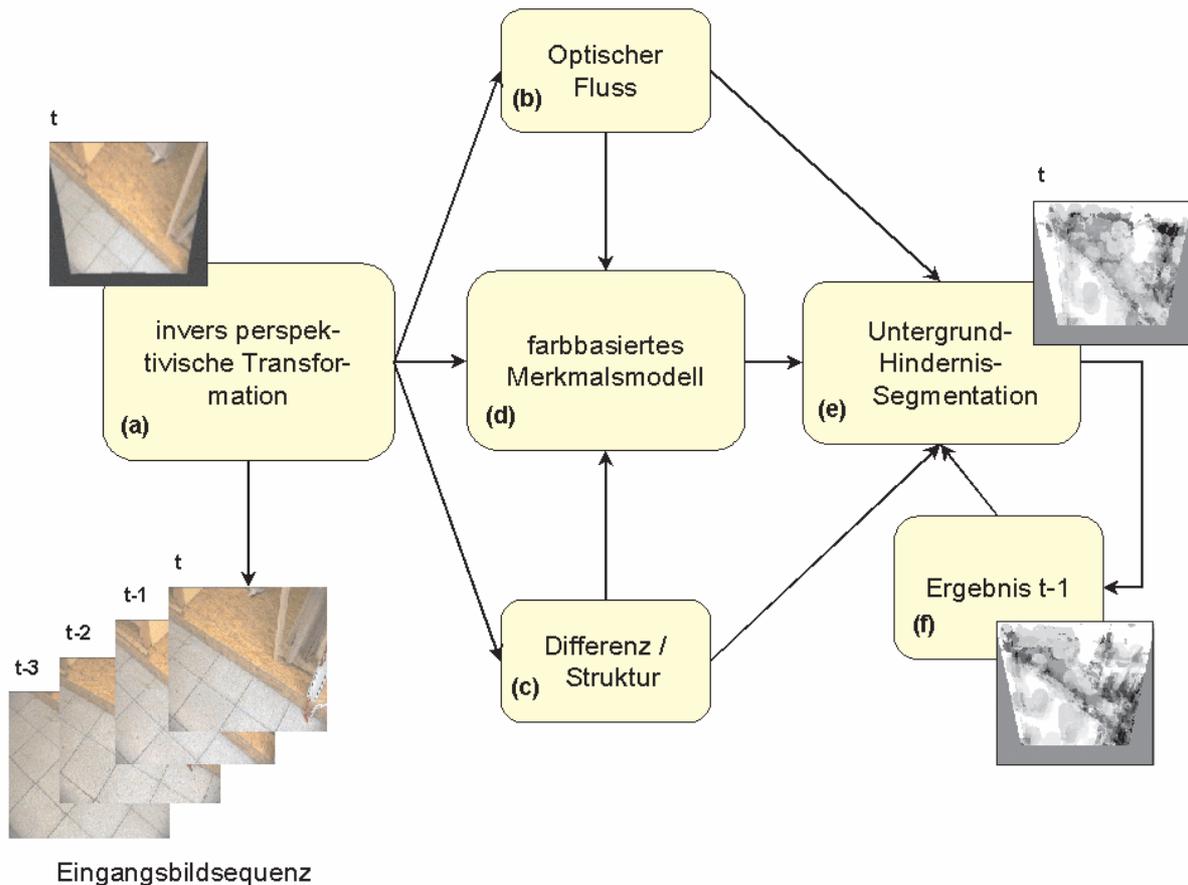


Abbildung 7.5: Architektur zur visuell-basierten Untergrund-Hindernis-Segmentation.

nutzt, um auch in den Bildregionen, die wenig oder gar keine Struktur aufweisen, eine Unterscheidung in Hindernis und befahrbaren Freiraum vornehmen zu können. Anschließend erfolgt eine Überlagerung aller Teilergebnisse zu einem resultierenden Untergrund-Hindernis-Segmentation (Block (e) in Abbildung 7.5), in das das jeweils vorangegangene Segmentationsergebnis (Block (f) in Abbildung 7.5) eingekoppelt wird. Die daraus resultierende Tiefpasswirkung trägt maßgeblich zur Stabilisierung des Segmentationsergebnisses bei.

Die folgenden Abschnitte gehen auf die einzelnen Verarbeitungsstufen dieser Architektur, die im Vergleich zu [KWBG98] und [Web98] überarbeitet und weiterentwickelt wurde [MB01], detaillierter ein.

Berechnung der invers-perspektivischen Transformation Für die analytisch geschlossene Berechnung der Transformation (siehe z.B. [Boh94] [MBLB91] [Web98]) wird typischerweise von der in Abbildung 7.7 dargestellten Anordnung ausgegangen.

Die geschlossene analytische Berechnung setzt jedoch die Kenntnis aller einfließenden geometrischen Größen voraus, was sich praktisch als problematisch erweist. So sind Parameter wie die Größe eines Pixels auf dem CCD-Chip oder die Orientierung des CCD-Chips bezüglich der optischen Achse der Kamera nur schwer zugänglich. Weiterhin erweist sich die Tatsache, dass

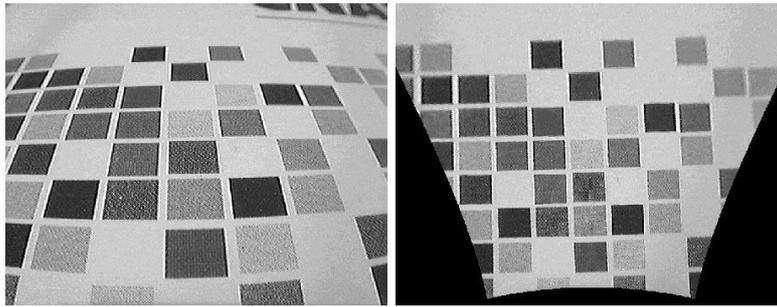


Abbildung 7.6: Beispiel für die invers-perspektivische Transformation. Links dargestellt ist das Originalbild, welches durch den Miniaturroboter KHEPERA mit einer um 45° nach unten geneigten Kamera aufgenommen wurde. Das rechte Bild zeigt das Ergebnis der Transformation, wobei alle Punkte des Originalbildes in der Bezugsebene liegen (nach [KWBG98]).

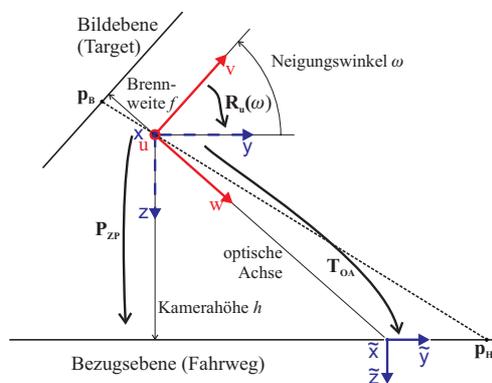


Abbildung 7.7: Prinzipanordnung für die Berechnung der invers-perspektivischen Transformation.

teilweise sehr kleine Änderungen von Parametern sehr große Auswirkungen auf das Transformationsergebnis haben, als ausgesprochen nachteilig. Aus den genannten Gründen wurde ein Näherungsverfahren implementiert, welches im Gegensatz zum Originalverfahren nur noch drei adjustierbare Parameter aufweist. Diese sind:

- k : Parameter zur Korrektur der tonnenförmigen radialen Bildverzerrung gemäß

$$\mathbf{x}' = \frac{\mathbf{x}}{1 + k|\mathbf{x}|^2} \quad (7.1)$$

- ϕ : Winkel der Kameraneigung bezüglich der horizontalen Bezugsebene
- s (*stretch*-Parameter): Maß für die Länge der optischen Achse von der Kamera bis zur Bezugsebene

Der Parameter k hängt nur von Kamera und verwendetem Kameraobjektiv ab und muss also nur einmal bestimmt werden. Zur (halb)automatischen Bestimmung der Parameter ϕ und s wird ein Testbild aufgenommen, welches eine regelmäßige orthogonale Gitterstruktur aufweisen muss. Anschließend wird die Transformation für verschiedene Werte von ϕ und s berechnet. Für jedes Transformationsergebnis wird nun mit Hilfe eines Orientierungstensors nach [Jae97] (siehe auch Abschnitt 7.2.1 und Abbildung 7.16) ein Histogramm für die Winkel von 0 bis 90°

berechnet. Wenn durch die aktuelle Parameterwahl die korrekte invers-perspektivische Transformation sehr gut angenähert wird, weist das Histogramm aufgrund der dann orthogonalen Gitterstruktur nur einen deutlich dominierenden Peak auf. Damit müssen also nur die Parameterwerte gesucht werden, für die das Histogramm den am besten ausgeprägten Peak beinhaltet (siehe auch Abschnitt 7.2.1). Abbildung 7.8 zeigt ein Beispiel für die invers-perspektivische Transformation des realen BauMarkt-Fußbodens, wobei das dargestellte Bild gleichzeitig auch als Testbild zur Bestimmung der Transformationsparameter genutzt werden kann.

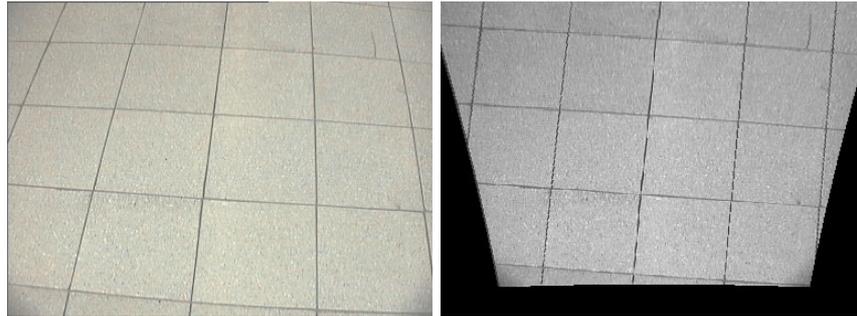


Abbildung 7.8: Beispiel für die invers-perspektivische Transformation des BauMarkt-Fußbodens. Links dargestellt ist das Originalbild, welches durch den PERSES-Roboter mit einer um 65° nach unten geneigten Kamera aufgenommen wurde. Das rechte Bild zeigt das Ergebnis der Transformation, wobei auch hier wie in Abbildung 7.6 alle Punkte des Originalbildes in der Bezugsebene liegen (nach [MB01]).

Berechnung und Analyse des optischen Flusses Die Einbeziehung des optischen Flusses dient dem Zweck, die Detektion von Verschiebungen, die nicht durch den Untergrund, also von potentiellen Hindernissen hervorgerufen werden, robuster zu gestalten.

Der optische Fluss stellt ein korrelatives Verschiebungsdetektionsverfahren dar. Für jeden Bildpunkt (x, y) im Bild C_t wird in einem Suchfenster im Bild C_{t-1} versucht, die Stelle $(x', y') = (x + v_x, y + v_y)$ zu finden, an der die größte Übereinstimmung herrscht. Dazu wird die Summe der absoluten Differenzen (SAD) für eine Nachbarschaft (= Korrelationsfenster $(x \pm \frac{i}{2}, y \pm \frac{j}{2})$) um den Punkt (x, y) untersucht:

$$SAD_{\vec{v}}(x, y) = \sum_j \sum_i \| C_t(x + i, y + j) - C_{t-1}(x' + i, y' + j) \| \quad (7.2)$$

Diese SAD wird für alle Verschiebungen $\vec{v} = (v_x, v_y)$ innerhalb des Suchfensters berechnet. Die gesuchte (Bild-)Verschiebung \vec{v}_{best} kann dann durch das Minimum über allen $SAD(x, y)$ bestimmt werden:

$$\vec{v}_{best}(x, y) = \operatorname{argmin} (SAD_{\vec{v}}(x, y)) \quad (7.3)$$

Die Qualität der berechneten Verschiebungsvektoren ist sehr stark abhängig vom Bildinhalt. Beispielsweise ist die Berechnung in großen einfarbigen oder unstrukturierten Flächen stark fehlerbehaftet, da eine eindeutige Korrelation innerhalb des Suchfensters nicht möglich ist. Um dieses Manko teilweise zu beheben, kann folgendes getan werden:

- Tiefpassfilterung der Verschiebungen \vec{v}_{best} , um stark rauschbehaftete und betragsmäßig kleine Verschiebungsvektoren zu eliminieren
- Ermittlung einer *Vertrauenswürdigkeit*. Dazu wird neben der besten Verschiebung \vec{v}_{best} noch die zweitbeste Verschiebung \vec{v}_{second} ermittelt. Ein Verschiebungsvektor \vec{v}_{best} wird dann als sicher definiert, wenn

1. $SAD_{\vec{v}_{best}}(x, y) \leq \theta_1$ und
2. $SAD_{\vec{v}_{second}}(x, y) - SAD_{\vec{v}_{best}}(x, y) > \theta_2$

gilt.

Die Berechnung des optischen Flusses ist sehr rechenintensiv. Daher ist es ratsam, mittels einer Vorfilterung der Eingangsdaten die Bildregionen zu ermitteln, in denen eine Flussberechnung überhaupt sinnvoll erscheint. Dazu soll für jeden Bildpunkt (x, y) ein *Interessenwert* $I(x, y)$ bestimmt werden, der angibt, wie gut sich die betreffende Bildregion zur Bestimmung des optischen Flusses eignet.

Dazu wird der in [Mor88] vorgeschlagene *Interest-Operator* verwendet. Für diesen Operator ist die Beschaffenheit der Grauwertstruktur unerheblich. Entscheidend für die Operatorantwort ist die Varianz der in der Struktur enthaltenen Grauwerte. Damit können mit diesem Filter auch Bildregionen detektiert werden, die keine Kanten aufweisen, aber deutlich strukturiert sind, was einen deutlichen Vorteil gegenüber der Verwendung eines reinen Kantensfilters bedeutet.

Jedem Bildpunkt (x, y) wird ein Interessenwert zugewiesen, indem zunächst für die horizontalen, vertikalen und diagonalen Bildrichtungen die summierten Richtungsunterschiede in der Umgebung dieses Bildpunktes berechnet werden:

$$I_1 = \sum_{(x,y) \in S} [C(x, y) - C(x, y + 1)]^2 \quad (7.4)$$

$$I_2 = \sum_{(x,y) \in S} [C(x, y) - C(x + 1, y)]^2 \quad (7.5)$$

$$I_3 = \sum_{(x,y) \in S} [C(x, y) - C(x + 1, y + 1)]^2 \quad (7.6)$$

$$I_4 = \sum_{(x,y) \in S} [C(x, y) - C(x + 1, y - 1)]^2 \quad (7.7)$$

Dabei steht S für die Menge der Bildpunkte des betrachteten Bildfensters. Nach der Berechnung der summierten Bilddifferenzen wird dem Bildpunkt (x, y) folgender Interessenwert zugewiesen:

$$I_{Op}(x, y) = \min(I_1, I_2, I_3, I_4) \quad (7.8)$$

Um eine Vergleichbarkeit der Ergebnisse bei verschiedenen Filtergrößen zu sichern, wird weiterhin eine Normierung des Interessenwertes vorgenommen. Wenn S ein quadratisches Filterfenster mit der Kantenlängen n_S ist, dann gilt:

$$I_{IOp}^{norm}(x, y) = \frac{1}{n_S^2} \cdot I_{IOp}(x, y) \quad (7.9)$$

Abbildung 7.9 zeigt drei Beispiele bei Anwendung des Interest-Operators mit verschiedenen Filtergrößen. Im Gegensatz zum einfachen Sobel-Operator als reinem Kantenfilter werden hier auch Bildregionen detektiert, in denen deutliche Strukturen, aber keine Kanten vorhanden sind.

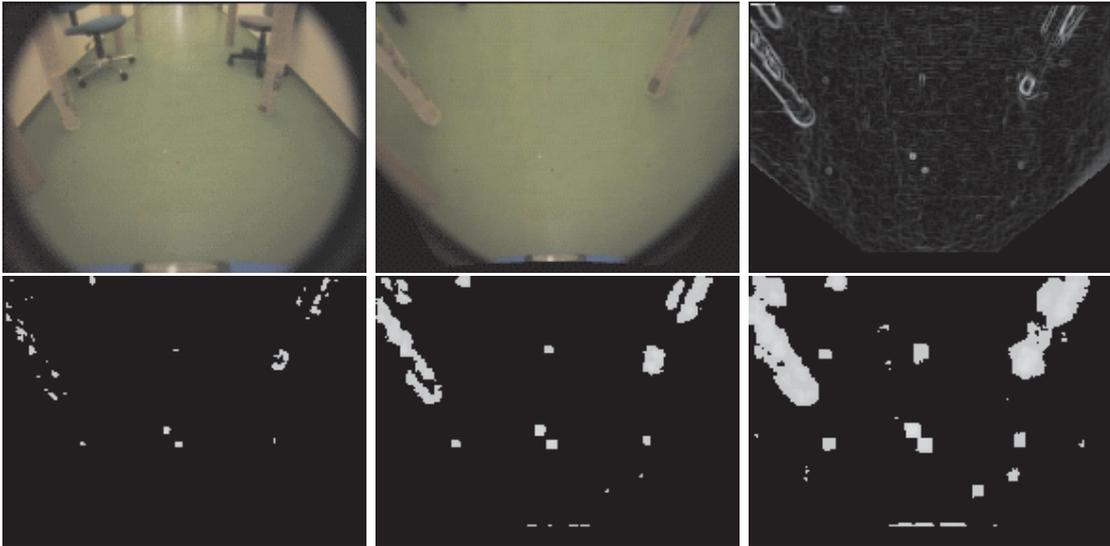


Abbildung 7.9: Beispiele für verschiedene Filter zur Detektion von interessanten Bildregionen. Von links-oben nach rechts-unten: Eingangsbild der Kamera; invers-perspektivisch transformiertes Bild; Ergebnis bei Anwendung des Sobel-Operators; Resultat bei Anwendung des Interest-Operators mit den Filtergrößen 3, 5 und 7 (nach [MB01]).

Bildbereiche mit einem zu geringen Interessenwert $I(x, y)$ werden a priori der Klasse *unsicher* zugeordnet und von der Berechnung des optischen Flusses ausgeschlossen.

Die Berechnung des optischen Flusses wird wie oben erläutert durchgeführt. Die Berechnung erfolgt jetzt aber lediglich für die Bildpunkte (x, y) , deren Interessenwert $I(x, y)$ über einer Schwelle θ_{OF} liegt. Nach der Berechnung der Verschiebungsvektoren \vec{v}_{best} und \vec{v}_{second} wird eine Klassifikation des Bildpunktes (x, y) in die drei Klassen *Hindernis*, *Freiraum* und *unsicher* vorgenommen:

1. Fall:

$$\begin{aligned} SAD_{\vec{v}_{best}}(x, y) &\leq \theta_1 \wedge \\ SAD_{\vec{v}_{second}}(x, y) - SAD_{\vec{v}_{best}}(x, y) &> \theta_2 \wedge \\ \|\vec{v}_{best}(x, y)\| &< \theta_v \end{aligned} \quad (7.10)$$

$\Rightarrow (x, y)$ ist Freiraum

Die beste Verschiebung führt zu einer guten Korrelation (kleiner Werte von $SAD_{\vec{v}_{best}}(x, y)$) der Bildfenster. Die zweitbeste Verschiebung besitzt deutlich höhere SAD. Die Verschiebung \vec{v}_{best} ist also eindeutig. Die Länge der Verschiebung \vec{v}_{best} ist klein. Damit hat sich diese Bildregion kaum bzw. nur sehr geringfügig bewegt. Damit kann diese Region der Klasse *Freiraum* zugeordnet werden.

2. Fall:

$$\begin{aligned} SAD_{\vec{v}_{best}}(x, y) &\leq \theta_1 \wedge \\ SAD_{\vec{v}_{second}}(x, y) - SAD_{\vec{v}_{best}}(x, y) &> \theta_2 \wedge \\ \|\vec{v}_{best}(x, y)\| &\geq \theta_v \end{aligned} \quad (7.11)$$

$\Rightarrow (x, y)$ ist Hindernis

Wie bei Fall 1. Die Länge der Verschiebung \vec{v}_{best} ist aber groß. Damit hat sich diese Bildregion erheblich bewegt. Der Geschwindigkeitsbetrag ist somit relativ groß und diese Region kann der Klasse *Hindernis* zugeordnet werden.

3. Fall:

$$\begin{aligned} SAD_{\vec{v}_{best}}(x, y) &\leq \theta_1 \wedge \\ SAD_{\vec{v}_{second}}(x, y) - SAD_{\vec{v}_{best}}(x, y) &\leq \theta_2 \end{aligned} \quad (7.12)$$

$\Rightarrow (x, y)$ ist unsicher

Die erst- und die zweitbeste Verschiebung besitzen beide sehr geringe SAD's. Damit kann keine eindeutige Verschiebung bestimmt werden. Der Bildpunkt wird der Klasse *unsicher* zugeordnet.

4. Fall:

$$SAD_{\vec{v}_{best}}(x, y) > \theta_1 \quad (7.13)$$

$\Rightarrow (x, y)$ ist unsicher

Wenn keiner der Fälle 1 - 3 aufgetreten ist, kann keine genaue Aussage zum Bildpunkt (x, y) getroffen werden. Der Bildpunkt wird somit der Klasse *unsicher* zugeordnet.

Bei einer idealen Kompensation der Roboterbewegung kann der Schwellwert $\theta_v = 0$ gesetzt werden. Da eine solche Bewegungskompensation aber nur in den seltensten Fällen zu 100% realisiert werden kann, ist es hilfreich, $\theta_v > 0$ zu wählen. Damit wird dann sichergestellt, dass Verschiebungsvektoren, die durch eine nichtideale Bewegungskompensation entstanden sind, nicht mit berücksichtigt werden.

Alle Punkte, die aufgrund eines zu geringen Interessenwertes $I(x, y)$ nicht untersucht wurden, werden wie oben bereits erläutert ebenfalls der Klasse *unsicher* zugeordnet.

Anhand der Klassifizierung der einzelnen Bildpunkte wird ein Ergebnisbild $R_{OF}(t)$ des Optischen-Fluss-Cue erzeugt, indem jedem Bildpunkt ein Grauwert zugewiesen wird.

Abbildung 7.10 zeigt ein Resultat bei der Analyse des optischen Flusses bei der Roboterplattform MILVA. In Abbildung 7.10 unten-rechts kann man erkennen, dass die Analyse des optischen Flusses im Prinzip keinen Freiraum detektiert hat. Die Ursache dafür ist, dass der Bereich des Fußbodens in diesem Bild keinerlei Struktur aufweist. Daher werden hier nur sehr kleine Interessenwerte $I(x, y)$ berechnet und damit werden diese Regionen von der Berechnung des optischen Flusses ausgeschlossen.

Differenzbild- bzw. Strukturanalyse Sind im Kamerabild keine Hindernisse vorhanden, so bewegen sich alle Punkte des Bildes entsprechend der Eigenbewegung des Roboters. Kompensiert man diese Eigenbewegung anhand der Odometriedaten, die aufgrund der kleinen ausgeführten Bewegung nicht korrigiert werden müssen, so erhält man wie oben bereits erläutert den gleichen Bildinhalt wie vor der Ausführung der Bewegung.

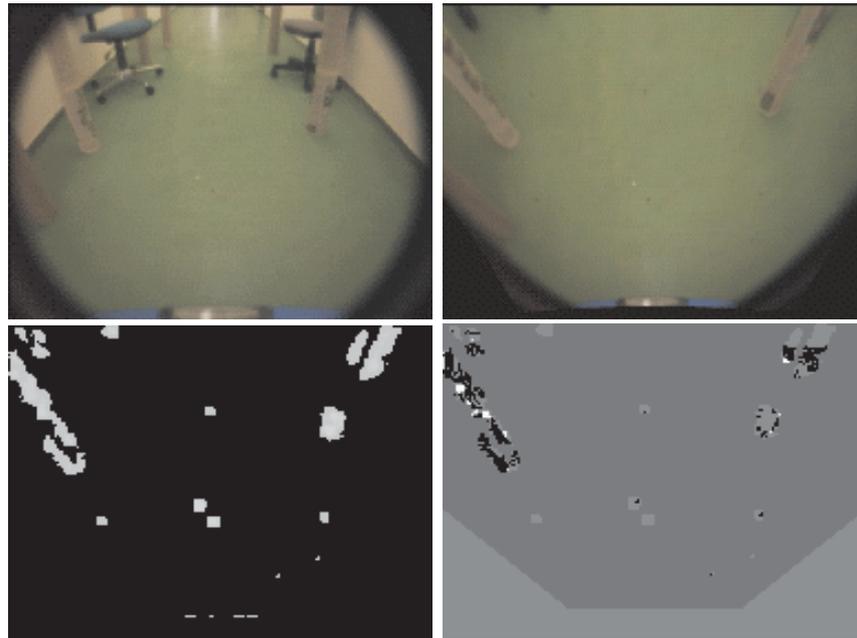


Abbildung 7.10: Resultat der Analyse des optischen Flusses. *Oben-Links: Originalkamera-Bild. Oben-Rechts: Invers-perspektivisches Kamerabild. Unten-Links: Ausgabebild des Interest-Operators mit Filtergröße 5 (zusätzlich mit einer Gamma-Korrektur behandelt). Unten-Rechts: Ergebnisbild der Analyse des optischen Flusses. Im Ergebnisbild kodieren die Grauwerte Schwarz Hindernis, Weiß Freiraum und Grau unsicheres Terrain (nach [MB01]).*

Unter der Voraussetzung, dass die Odometriedaten des Roboters hinreichend genau bekannt sind, kann ein Differenzbild D_t aus dem aktuellen Eingangsbild C_t und dem bezüglich der Eigenbewegung des Roboters kompensierten Bildes C_{t-1} berechnet werden. Dazu wird für jeden Punkt der Betrag der Differenz gebildet:

$$D_t(x, y) = \| C_t(x, y) - C_{t-1}(x, y) \| \quad (7.14)$$

Für die Bildpunkte der horizontalen Bezugsebene ist $D_t(x, y)$ idealerweise gleich Null. In der Praxis trifft dies jedoch aufgrund des Kamerarauschens und der nicht 100%-igen Bewegungskompensation nicht zu. Die Auswertung des Differenzbildes ist weiterhin nur für Bildregionen sinnvoll, die strukturiert sind. Die Ursache dafür liegt darin, dass kleine Werte von $D_t(x, y)$ sowohl durch gleichförmige Grauwerte innerhalb einer Bildregion als auch durch eine exakte Bewegungskompensation entstanden sein können.

Zur Ermittlung des Strukturgehaltes einer Bildregion wird hier wiederum der Interest-Operator eingesetzt, wobei dieser direkt auf das Differenzbild angewandt wird. Das Ausgabebild D_t^{IOP} dieser Operation kann dann direkt zur Einordnung der einzelnen Pixel in die Klassen *Freiraum*, *unsicher* und *Hindernis* genutzt werden (siehe Gleichung 7.15 bis 7.17).

Dies ist möglich, da einerseits der befahrbare Untergrund nur sehr kleine Werte im Differenzbild erzeugt, und somit der Interessenwert in diesen Regionen sehr klein ausfallen wird. Gleichzeitig werden Regionen mit Hindernissen durch deutlich höhere Werte im Differenzbild gekennzeichnet. Damit wird der Interest-Operator in diesen Regionen einen hohen Interessen-

wert errechnen.

Um durch das Kamerarauschen entstehende Bildstörungen herauszufiltern, wird vor der Anwendung des Interest-Operators auf das Differenzbild dieses mit einem Glättungsfilter vorbehandelt. Dazu kommt hier ein einfacher Binomialfilter (siehe [Jae97]) zum Einsatz.

Fallunterscheidung zur Klassifizierung der einzelnen Bildpunkte:

1. Fall:

$$D_t^{IOP}(x, y) > \theta_{max} \quad (7.15)$$

$\Rightarrow (x, y)$ ist Hindernis

2. Fall:

$$D_t^{IOP}(x, y) \in [\theta_{min}, \theta_{max}] \quad (7.16)$$

$\Rightarrow (x, y)$ ist unsicher

3. Fall:

$$D_t^{IOP}(x, y) < \theta_{min} \quad (7.17)$$

$\Rightarrow (x, y)$ ist Freiraum

Anhand der Klassifizierung der einzelnen Bildpunkte wird wieder ein Ergebnisbild $R_{ST}(t)$ der Struktur-Cue erzeugt, indem wiederum jedem Bildpunkt ein Grauwert zugewiesen wird (Weiß für Freiraum, Schwarz für Hindernis, Grau für unsicheres Terrain). Abbildung 7.11 zeigt ein Beispiel für die Strukturanalyse.

Adaptives Farbmodell zur Bildsegmentation Mit den in den letzten beiden Abschnitten vorgestellten Verfahren können Bildpunkte ermittelt werden, die eindeutig den Klassen *Hindernis* oder *FREIRAUM* zugeordnet werden können. Die Anzahl dieser Pixel kann jedoch von Bild zu Bild stark schwanken. Auch innerhalb eines Bildes können unter Umständen große Flächen vorhanden sein, die noch nicht klassifiziert werden konnten.

Man kann davon ausgehen, dass sich der Inhalt von aufeinanderfolgenden Bildern nicht vollständig ändert. Regionen, die als Hindernis bzw. Freiraum erkannt wurden, werden mit hoher Wahrscheinlichkeit auch noch vorhanden sein. Die Idee besteht nun darin, die Farbinformationen der eindeutig einer Klasse zugeordneten Pixel bei der Segmentierung in Hindernis und Freiraum zu benutzen.

Um Helligkeitsunterschiede im Bild zu ignorieren, werden die Farbwerte der Bildpunkte in den dichromatischen Farbraum transformiert:

$$r = \frac{R}{R + G + B} \quad g = \frac{G}{R + G + B} \quad (7.18)$$

Die mit den Verfahren zur Analyse des optischen Flusses und der Strukturanalyse ermittelten Bildpunkte können somit als Trainingsdaten für einen Vektorquantisierer genutzt werden. Damit kann dann ein Merkmalsmodell gelernt werden, mit dem anschließend eine weitere Segmentierung des Eingangsbildes erfolgen kann. Um das Training der Farbmodelle für Untergrund und Hindernis zu vereinfachen, werden zwei separate Vektorquantisierer verwendet, von denen einer die Farbwerte des sicher detektierten Untergrundes, der andere die Farbwerte der sicher detektierten Hindernisbereiche repräsentiert.

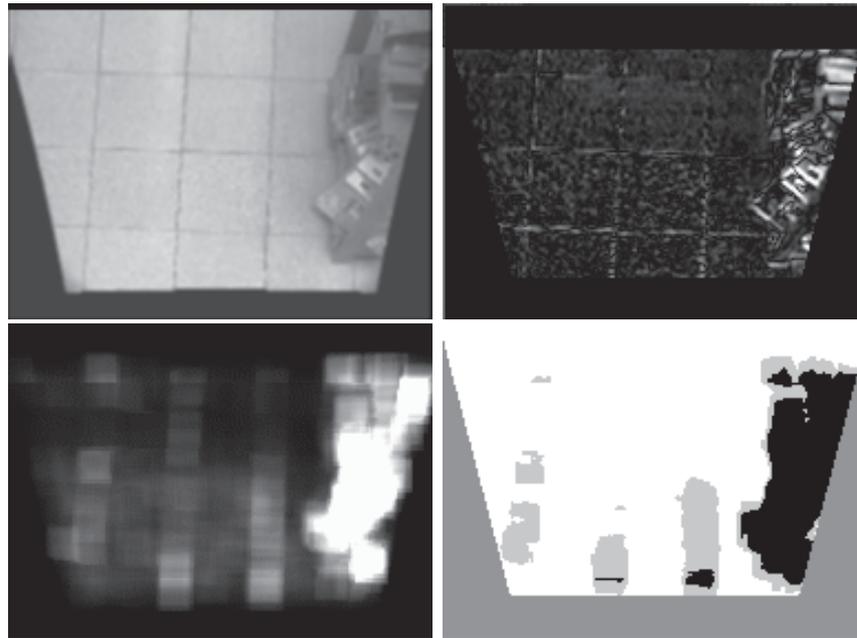


Abbildung 7.11: Resultat der Strukturanalyse. Oben links: Invers-perspektivisches Kamerabild C_t . Oben rechts: Differenzbild D_t zum vorherigen Zeitschritt. Unten links: Differenzbild mit Interest-Operator behandelt D_t^{IOP} . Unten rechts: Ergebnisbild $R_{ST}(t)$ der Struktur- und Differenzanalyse. Im Ergebnisbild kodieren die Grauwerte Schwarz Hindernis, Weiß Freiraum und Grau unsicheres Terrain (nach [MB01]).

Folgende Aspekte sind für die Bereitstellung von Trainingsdaten sowie für die Handhabung der beiden Vektorquantisierer wesentlich:

- Durch die Einbeziehung der Ortsinformationen der sicher detektierten Pixel wird verhindert, dass über das gesamte Bild Pauschalaussagen für Farbwerte getroffen werden. Ein einzelnes Trainingsdatum besteht somit aus einem Tupel der Orts- und Farbinformation $[x, y, C'_t(x, y)]$.
- Um eine variable Anzahl von Stützstellen zu realisieren und den Berechnungsaufwand klein zu halten, wurde ein einfacher Vektorquantisierer verwendet. Der Vektorquantisierer besteht aus einer Menge von Stützstellen V_i im „Orts-Farb-Raum“ mit $V_i = [x_i, y_i, R_i, G_i, n_i]$. Der Wert n_i steht dabei für die Anzahl der Pixel der Trainingsdaten des aktuellen Eingangsbildes, die durch diese Stützstelle abgedeckt werden.

Vor dem Beginn des Trainings werden alle $n_i = 0$ gesetzt. Danach erfolgt eine Vektorquantisierung nach folgendem Algorithmus:

1. Hole Trainingsdatum $[x, y, R, G]$
2. Prüfe alle vorhandenen Stützstellen $V_i = [x_i, y_i, R_i, G_i, n_i]$:
3. Bestimme Ortsabstand $d_i = \| (x, y)^T - (x_i, y_i)^T \|$.
4. Falls $d_i > \theta_{distance}$ gehe zu 2.

5. Bestimme Farbabstand $c_i = \| (R, G)^T - (R_i, G_i)^T \|$.
6. Falls $c_i > \theta_{color}$:
7. Erzeuge neue Stützstelle $V_j = [x, y, R, G, 1]$
8. Sonst:
9. Inkrementiere n_i .
10. gehe zu 2, falls noch nicht alle Stützstellen untersucht wurden.

Danach erfolgt eine Segmentierung des Eingangsbildes, indem für jeden Pixel (x, y) die Stützstellen in V^{ground} und $V^{obstacle}$ gesucht werden, die sowohl örtlich als auch farblich gut zum aktuellen Eingangsbild passen.

Im Vektorquantisierer V^{ground} wird die Stützstelle V_i^{ground} gesucht, die den minimalen Farbabstand c_i^{ground} zum Farbwert des Pixels (x, y) und einen örtlichen Abstand d_i^{ground} hat, der kleiner als $\theta_{distance}$ ist.

Analog werden für den Vektorquantisierer $V^{obstacle}$ die Werte $c_j^{obstacle}$ und $d_j^{obstacle}$ berechnet. Anschließend erfolgt eine Klasseneinordnung des Pixels (x, y) gemäß folgender Fallunterscheidung:

1. Fall:

$$\begin{aligned} d_i^{ground} &> \theta_{maxdistance} \wedge \\ d_j^{obstacle} &> \theta_{maxdistance} \end{aligned} \quad (7.19)$$

$\Rightarrow (x, y)$ ist unsicher

2. Fall:

$$\begin{aligned} c_i^{ground} &> \theta_{maxcolor} \wedge \\ c_j^{obstacle} &> \theta_{maxcolor} \end{aligned} \quad (7.20)$$

$\Rightarrow (x, y)$ ist unsicher

3. Fall:

$$\begin{aligned} c_i^{ground} &< \theta_{mincolor} \wedge \\ c_j^{obstacle} &< \theta_{mincolor} \end{aligned} \quad (7.21)$$

$\Rightarrow (x, y)$ ist unsicher

4. Fall:

$$c_i^{ground} \leq c_j^{obstacle} \quad (7.22)$$

$\Rightarrow (x, y)$ ist Freiraum

5. Fall:

$$c_i^{ground} > c_j^{obstacle} \quad (7.23)$$

$\Rightarrow (x, y)$ ist Hindernis

Anhand der Klassifizierung der einzelnen Bildpunkte wird wieder ein Ergebnisbild erzeugt. Abbildung 7.12 zeigt das Ergebnis einer Segmentierung nach diesem Verfahren.

Aufgrund der Verwendung der Ortsinformationen ist beim Übergang vom Zeitschritt t zum Zeitschritt $t + 1$ eine Anpassung der gespeicherten Ortsinformationen der einzelnen Stützstellen

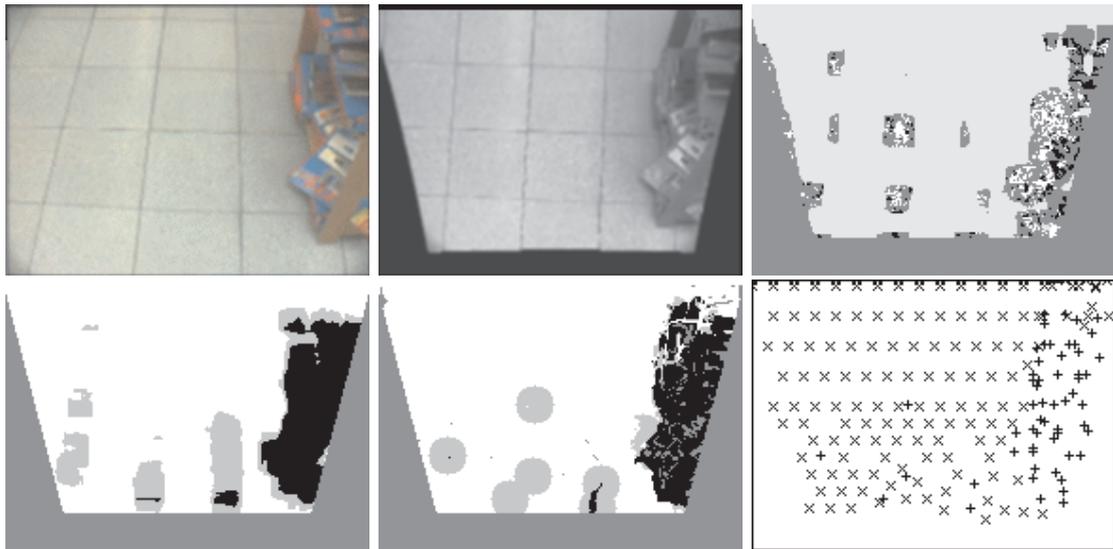


Abbildung 7.12: Ergebnis der vektorquantisierten Bildsegmentation (nach [MB01]). Oben (v.l.n.r.): Original Kamerabild zum Zeitpunkt t , Invers-perspektivisches Kamerabild C_t und Ergebnisbild $R_{OF}(t)$ der Analyse des optischen Flusses. Unten (v.l.n.r.): Ergebnisbild $R_{ST}(t)$ der Struktur- und Differenzanalyse, Ergebnisbild $R_{NG}(t)$ der Bildsegmentation mit den adaptiven Farbmodellen für Untergrund und Hindernisse (hier kodieren wiederum die Grauwerte Schwarz Hindernis, Weiß Freiraum und Grau unsicheres Terrain) und die Position der Stützstellen der beiden Vektorquantisierer (Kreuz steht dabei für Freiraum und Plus steht für Hindernis).

notwendig. Es erfolgt hierbei eine Transformation anhand der Odometriedaten des Roboters, die ähnlich der Kompensation der Eigenbewegung des Roboters ist. Stützstellen, die sich bei der Transformation aus dem sichtbaren Bildbereich des Roboters herausbewegt haben, werden hierbei gelöscht, da für diese keine sinnvolle Verwendung mehr besteht.

Weiterhin hat es sich als sinnvoll erwiesen, dass zu jeder Stützstelle ein *Alter* vermerkt wird. In jedem Zeitschritt werden diese Alterswerte dann inkrementiert. Sobald eine Stützstelle vom aktuellen Trainingsdatensatz verwendet wird, wird deren Alter auf 0 zurückgesetzt. Stützstellen, die ein vorgegebenes Maximalalter überschreiten, werden gelöscht.

Fusion der Einzelcues und Segmentationsergebnisse Als Resultat der in den letzten Abschnitten vorgestellten Verfahren stehen zur Berechnung des Gesamtergebnisses nun insgesamt drei Zwischenergebnisse bezüglich der einzelnen Cues zur Verfügung. Diese drei Zwischenergebnisse werden gemäß der Gleichung (7.24) zu einem Gesamtergebnis $R(t)$ fusioniert.

$$R(t) = a_1 * R_{OF}(t) + a_2 * R_{ST}(t) + a_3 * R_{NG}(t) + a_4 * R(t-1) \quad (7.24)$$

mit:

$$\sum_i a_i = 1$$

R_{OF} steht für das Ergebnis der Analyse des optischen Flusses, R_{ST} für das Ergebnis der Strukturanalyse und R_{NG} für das Segmentierungsergebnis mittels der Vektorquantisierung. $R'(t-$

1) symbolisiert das bezüglich der Eigenbewegung des Roboters kompensierte Ergebnisbild zum Zeitschritt $t - 1$. Durch die Einbeziehung des Resultates $R(t - 1)$ in diese Berechnung wird eine zeitliche Tiefpassfilterung der Untergrund-Hindernis-Segmentation realisiert. Damit können zum einen kurzzeitige Störungen im Resultat wirkungsvoll unterdrückt werden. Weiterhin führt diese Tiefpassfilterung zu einer deutlichen Stabilisierung des finalen Segmentierungsergebnisses, aus dem sich eine „scan-äquivalente“ Repräsentation, wie in Abbildung 7.13 illustriert, erzeugen lässt.

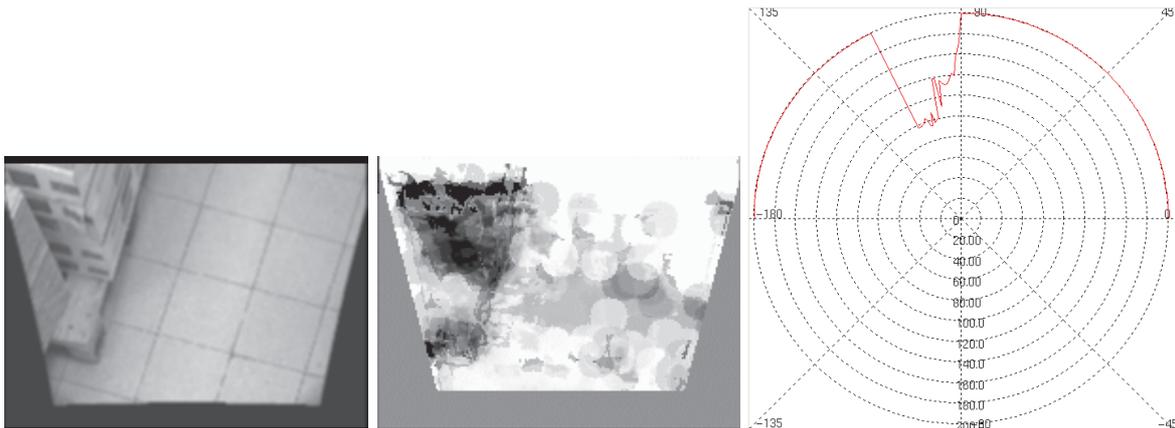


Abbildung 7.13: Aus dem Segmentierungsergebnis der visuell-basierten Hindernisvermeidung (Mitte) gewonnenes Scanprofil (rechts), das den durch die Kamera erfassten Bereich vor dem Roboter abdeckt. Das entsprechende invers-perspektivisch transformierte Originalbild ist links dargestellt (nach [MB01]).

Mittels Sensorfusion, die das sonarbasierte und das visuell erzeugte Scanprofil integriert, sollen im Rahmen aktuell laufender Untersuchungen sowohl die Hindernisvermeidung als auch der Aufbau der lokalen / globalen Umgebungskarte hinsichtlich Robustheit entscheidend verbessert werden. Experimentelle Ergebnisse lagen zum Abschluss dieser Arbeit leider noch nicht vor. Weiterhin ist vorgesehen, das eben beschriebene Verfahren auch auf die Bilder der Omnikamera anzuwenden, wobei dann nahezu der gesamte für die Navigation relevante Umgebungsbereich des Roboters auf mögliche Hindernisse untersucht werden kann.

Probleme und Limitierungen Auch das visuell-basierte Verfahren zur Hindernisvermeidung besitzt logischerweise eine Reihe von Limitierungen, die nachfolgend kurz erörtert werden sollen:

- Reflexionen auf dem Untergrund führen zu großen Unterschieden sowohl bei der Differenzbildanalyse als auch bei der Berechnung des optischen Flusses. Somit werden Regionen des Untergrundes fälschlicherweise der Kategorie Hindernis zugeordnet.
- Eine weitere Einschränkung des hier vorgestellten Verfahrens ist durch die Größe der Hindernisse bzw. deren Erhebung im Vergleich zum befahrbaren Untergrund gegeben, die durch das Verfahren noch wahrgenommen werden können. So zeigt sich, dass vor allem sehr flache Hindernisse (Höhe $< 10\text{cm}$) ein Problem darstellen. Bei der Analyse

des optischen Flusses kann der Geschwindigkeitsbetrag solcher Objekte kaum von dem des befahrbaren Untergrundes unterschieden werden. Auch die Strukturanalyse versagt hier meistens, da flache Objekte beim Vergleich des aktuellen und des bewegungskompensierten vorhergehenden Bildes kaum Unterschiede hervorrufen.

- Hindernisse, die eine unstrukturierte Oberfläche bzw. eine homogene Farbe besitzen, können ebenfalls nur schwer detektiert werden, da aufgrund der fehlenden Strukturen hier die Analyse des optischen Flusses keine signifikanten Ergebnisse liefern kann. Durch die relativ homogene Farbe versagt auch die Differenzbildung und damit auch die Strukturanalyse. Lediglich die Kanten und Ränder solcher Objekte können gut detektiert werden.
- Schlussendlich muss angemerkt werden, dass die derzeitige Implementierung etwa 1s Rechenzeit pro Bildpaar benötigt. Um auch bei zügiger Fahrt schritthaltend visuell Hindernisse zu detektieren, ist eine weitere Performanzsteigerung durch eine effizientere Implementierung, den Einsatz leistungsfähigerer Hardware sowie die Einbeziehung geeigneter Heuristiken notwendig und Gegenstand aktueller Forschungsarbeiten am Fachgebiet.

7.2 Aufbau einer globalen Umgebungskarte

Um den Aufbau der globalen Umgebungskarte möglichst einfach und effizient zu gestalten, wurde versucht, die Abfolge der lokalen Umgebungskarten (siehe Abschnitt 7.1.2), die während der Fahrt durch die Einsatzumgebung berechnet werden, entsprechend der Odometriedaten zu einer globalen Karte aneinander zu fügen. Voraussetzung dafür ist jedoch, dass die Odometriedaten hinreichend korrekt sind bzw. dass diese mit einem geeigneten Verfahren korrigiert werden können. Abbildung 7.14 zeigt eine globale Karte (Ausschnitt aus der BauMarkt-Einsatzumgebung), die ausschließlich auf der Basis der Odometriedaten aufgezeichnet wurde. Dabei befuhr der Roboter einen geschlossenen Kurs um mehrere Regalreihen. Während sich der Roboter nach der Fahrt real wieder an seiner Ausgangsposition befindet, sind in der aufgebauten Karte deutliche Inkonsistenzen zu erkennen, die auf den sich akkumulierenden Odometriefehler zurückzuführen sind.



Abbildung 7.14: Globale Umgebungskarte, die ohne zusätzliche Mechanismen zur Odometriekorrektur aufgebaut wurde. Hellgrau kodiert Freiraum (Belegtheitswert nahe 0), Schwarz kodiert Hindernis (Belegtheitswert nahe 1). Deutlich zu erkennen ist die Inkonsistenz der Karte, da Anfangs- und Endposition in der realen Umgebung identisch sind, jedoch in der aufgebauten Karte räumlich weit auseinander liegen.

Um die Odometriedaten des Roboters anhand von extern zugänglichen Informationen zu korrigieren, wurde das nachfolgend beschriebene visuelle Odometriekorrekturverfahren entwickelt. Es nutzt die Tatsache aus, dass der Fußboden des BauMarktes, der als Testumgebung dient, eine Fliesenstruktur aufweist, die sich einheitlich über den gesamten Markt erstreckt.

7.2.1 Visuelle Odometriekorrektur

Da die Odometrie von PERSES nur eine Positionsschätzung und keine exakte Positionsbestimmung ermöglicht und sich Odometriefehler über die Dauer der Roboterbewegung aufakkumulieren, kommt es beim rein odometrie-basierten Aufbau der globalen Karte zu entsprechenden Problemen. So erscheinen lange gerade Gänge gekrümmt, und geschlossene Wege haben keine Verbindung oder überkreuzen sich frühzeitig (siehe Abbildung 7.14).

Um dieses Problem zu beseitigen, gibt es prinzipiell zwei Vorgehensweisen. So kann man die globale Karte zunächst direkt aus den aufgezeichneten Messdaten (Odometrie und Sensorwerte) erstellen und sich alle wichtigen Daten (z.B. Landmarken) zwischenspeichern, um später eine Korrektur von aktueller Position und aktueller Karte zu ermöglichen.

Einen sehr eleganten Ansatz zum sonarbasierten Kartenaufbau mit Odometriefehlerkorrektur stellt der Expectation-Maximization(EM)-Algorithmus [DLR77] dar. Der EM-Algorithmus behandelt das Problem des Kartenlernens als Wahrscheinlichkeitsmaximierungsproblem (siehe auch [TBF98]). Dazu werden zunächst ein Bewegungs- sowie ein Sensormodell experimentell bestimmt. Das Bewegungsmodell spiegelt die statistische Verteilung der Odometriefehler wider und beschreibt nichts anderes als die Wahrscheinlichkeitsverteilung der tatsächlich zurückgelegten Strecke des Roboters in Bezug auf den laut Odometrie zurückgelegten Weg. Das Sensormodell beinhaltet die Wahrscheinlichkeitsverteilung zwischen der von einem Ultraschallsensor gemessenen Entfernung und der tatsächlichen Entfernung. Der eigentliche Algorithmus läuft dann in zwei Schritten ab. In einem ersten Schritt wird für alle nach dem Bewegungsmodell und den Odometriedaten möglichen Positionen der globalen Karte die Wahrscheinlichkeit berechnet, dass sich der Roboter dort befindet. Dazu wird aus den aktuellen Sonarinformationen und der bisher aufgebauten Karte anhand des Sensormodells die Wahrscheinlichkeit für jede mögliche Position ermittelt und mit der Wahrscheinlichkeit für die jeweilige Position verrechnet. Die Position mit der höchsten Gesamtwahrscheinlichkeit wird dann dazu genutzt, die lokale Karte mit der bisher erstellten globalen Karte zu verrechnen und diese damit anzupassen. Dieses Verfahren ermöglicht in seiner eigentlichen Form zunächst nur eine lokale Optimierung und kann global gesehen nicht das optimale Ergebnis hervorbringen. Dies kann durch einen veränderten Algorithmus weitgehend behoben werden, indem die Wahrscheinlichkeiten über mehrere Schritte berechnet werden und erst zu einem späteren Zeitpunkt die Karte aus den aufgezeichneten Daten rekonstruiert wird. Dies könnte z.B. beim Erreichen einer Stelle der Umgebung, an der sich der Roboter vor längerer Zeit schon einmal befunden hat, geschehen. Dieser abgewandelte Algorithmus liefert sehr robuste Ergebnisse und ist von der Einsatzumgebung relativ unabhängig. Nachteilig sind jedoch der hohe Rechenaufwand und der enorme Speicherverbrauch, der das Verfahren nur bedingt echtzeitfähig macht. Weiterhin ist das Ergebnis dieses Algorithmus sehr stark von der Güte des Bewegungsmodells und des Sensormodells abhängig und damit auch von den Eigenschaften des verwendeten Roboters. Eine genaue Beschreibung des Verfahrens ist neben [TBF98] auch in [BFJ⁺99] zu finden. Für aktuelle Erweiterungen, insbesondere auch hinsichtlich der Umsetzung der Echtzeitfähigkeit, sei auf [TBF01] verwiesen.

Bei der zweiten Vorgehensweise versucht man, die Odometriefehler direkt zu korrigieren, indem man zusätzliche Informationen der Umgebung aus weiteren Sensorinformationen berücksichtigt. Beispielsweise könnte man mit einem Laserscanner, der sehr genaue Entfernungswerte liefert, anhand der Änderungen zur vorherigen Messung eine recht exakte Bewegungsschätzung ermitteln, sofern dies die Struktur der Umgebung zulässt (z.B. würden bei langen geraden Gängen Probleme bei der möglichst eindeutigen Passung aufeinander folgender Scans auftreten). Ähnliche Ergebnisse könnte man auch durch die Auswertung von Kamerabildern erreichen.

Da das größte Problem beim Kartenaufbau ist die stark fehlerbehaftete Rotationsangabe der Odometrie, was dazu führt, dass der Positionsfehler hauptsächlich im Rotationsfehler begründet liegt. Deshalb wird zunächst versucht, den Rotationsfehler zu minimieren oder ganz zu eliminieren. Die Entwicklung des nachfolgend beschriebenen Verfahrens rührt maßgeblich aus der Tatsache, dass der Fußboden des konkreten BauMarktes, in dem die experimentellen Untersuchungen durchgeführt wurden, über eine regelmäßige Fliesenstruktur verfügt, die sich einheitlich über das gesamte Marktareal erstreckt. Die Idee war nun, die Fliesenstruktur des Fußbodens mit Hilfe einer Kamera aufzunehmen und anschließend auszuwerten. Dadurch lässt sich die Ausrichtung des Roboters bzw. die Ausrichtung des Roboters relativ zum Fußboden bestimmen.

Eine Kamera ist dazu in ca. 1.2 m Höhe nach unten gerichtet und liefert ein Fußbodenbild (siehe Abbildung 7.15). Dieses Bild wird einer Analyse der lokalen Nachbarschaften nach der Strukturtenormethode (vgl. [Jae97]) unterzogen, die für jeden Bildbereich die lokal dominante Orientierung liefert.

Um den Strukturtenor zu erhalten, wird das Bild zunächst einer Bandpassfilterung unterzogen. Anschließend bildet man die horizontale und die vertikale Ableitung entsprechend (D_x, D_y) . Im nächsten Schritt werden diese Ableitungen jeweils quadriert $(D_x * D_x, D_y * D_y)$. Zusätzlich wird das Produkt aus horizontaler und vertikaler Ableitung gebildet $(D_x * D_y)$. Dies geschieht jeweils pixelweise. Anschließend werden diese 3 Bilder mit einem Binomialfilter geglättet $(J_{xx} = R(D_x * D_x), J_{yy} = R(D_y * D_y), J_{xy} = R(D_x * D_y))$.

Nun können x -Komponente $(J_{xx} - J_{yy})$ und y -Komponente $(2 * J_{xy})$ des Orientierungsvektors berechnet werden. Über den Arcustangens lässt sich daraus der Orientierungswinkel von $[-\pi/2, \pi/2]$ berechnen. Neben dem Orientierungswinkel wird noch der quadrierte Betrag des Gradienten $(J_{xx} + J_{yy})$ als Wichtungsfaktor benötigt. Dieser gibt an, wie stark (und damit zuverlässig) die lokal dominante Orientierung für den entsprechenden Bildpunkt berechnet werden konnte.

Um daraus die Ausrichtung des Roboters zu ermitteln, wird nun der Orientierungswinkel jedes Bildelements mit seiner entsprechenden Gewichtung in ein Histogramm eingetragen. Da die Fliesenstruktur zwei nicht unterscheidbare Hauptrichtungen aufweist, die senkrecht aufeinander stehen, werden die Orientierungen mittels Modulo-Operator auf den Bereich $[0, \pi/2]$ abgebildet, um die zwei Häufungen des Histogramms zu einer einzigen zu addieren, was die Robustheit des Verfahrens erhöht. Aus dem Maximum des zuvor geglätteten Histogramms kann man anschließend die Orientierung des Fliesenbodens ermitteln. Da hier nur ein Winkel zwischen 0 und $\pi/2$ als Ergebnis erscheint, muss dieser Winkel mit Hilfe der aktuellen Odometriedaten noch umgerechnet werden. Dies geschieht, indem aus dem ermittelten Winkel alle vier möglichen Winkel $(+0^\circ, +90^\circ, +180^\circ, +270^\circ)$ berechnet werden und der am besten zum

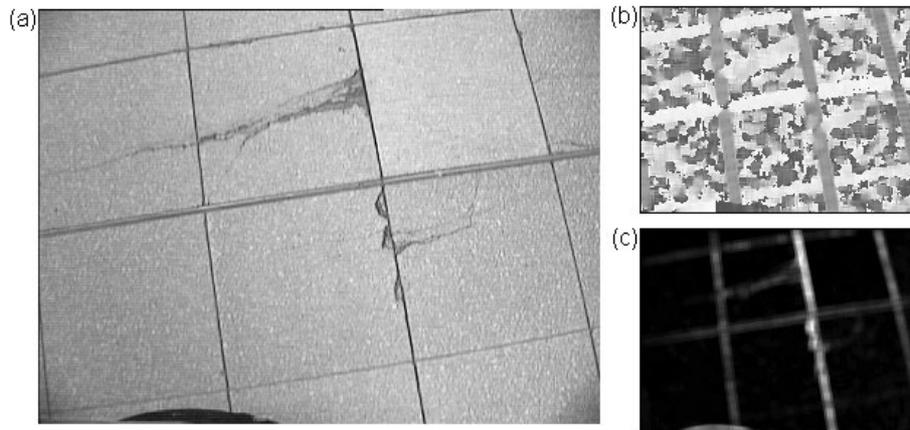


Abbildung 7.15: Strukturtensorberechnung: (a) aufgenommenes Fußbodenbild, (b) ermittelte dominante Orientierungen (als Grauwerte kodiert), (c) Wichtung (aus [Lie00] bzw. [BGKW00]).

aktuellen Odometriewinkel passende Winkel ausgewählt wird.

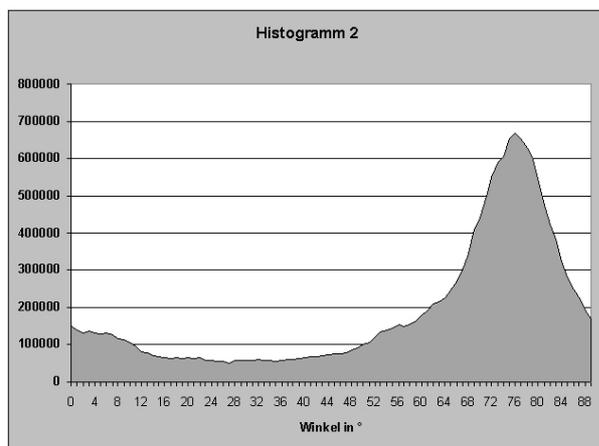


Abbildung 7.16: Histogramm der Orientierungswinkel (aus [Lie00] bzw. [BGKW00]).

Dadurch können Rotationsfehler von bis zu $\pm 45^\circ$ eliminiert werden. Größere Fehler führen dazu, dass bei der Winkelauswahl der falsche Quadrant gewählt wird und der Gesamtrrotationsfehler, der eigentlich Null sein sollte, ein Vielfaches von 90° erreicht. Wenn der Abstand von aufeinanderfolgenden Odometriefehlerkorrekturen relativ klein gehalten wird, was in der vorliegenden Implementierung mit einer Performanz von etwa 3.5 Bildern pro Sekunde gegeben ist, kann dies praktisch kaum auftreten. Mit dem beschriebenen Verfahren kann der Rotationsfehler kontinuierlich sehr genau ermittelt und damit die Odometriedaten korrigiert werden.

Die Korrektur der Roboterposition geschieht dann lediglich mit Hilfe des ermittelten Rotationsfehlers. Dazu wird für jede ausgeführte Bewegung, die als geradlinig angesehen wird, die Differenz der aktuellen zur vorherigen Position berechnet $(x_2; y_2) - (x_1; y_1)$. Dieser Bewegungsvektor wird anschließend einfach um den in diesem Schritt ermittelten Rotationsfehler α rotiert und die korrigierten Koordinaten $(x_2; y_2)$ berechnet (siehe Abbildung 7.17).

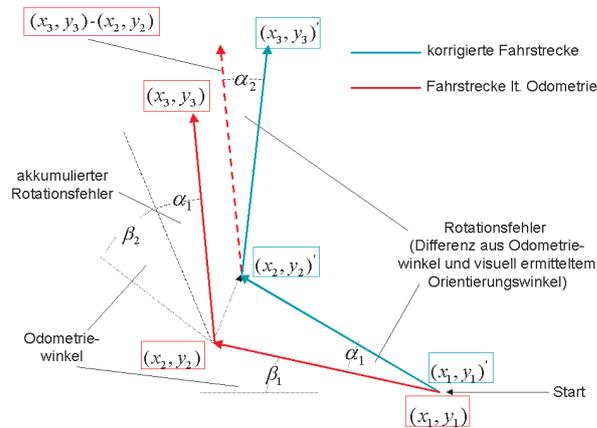


Abbildung 7.17: Funktionsweise der Positionskorrektur der Odometriedaten.

7.2.2 Eigentlicher Kartenaufbau

Die globale Karte wird sukzessive aus den lokalen Karten mit den zugehörigen Positionsinformationen des Roboters zum Zeitpunkt der Messung aufgebaut. Die Positionsinformation, die notwendigerweise zu jeder erstellten lokalen Karte existieren muss, wird von der Odometrie des Roboters geliefert. Diese enthält ausgehend von der absoluten Startposition (Position zum Zeitpunkt des Einschaltens) die x- und y-Koordinaten (in Meter) der aktuellen Roboterposition und die Ausrichtung des Roboters. Diese Odometriedaten unterliegen abhängig von der gefahrenen Strecke recht großen Fehlern, die sich stark auf die globale Karte auswirken. Um das Prinzip der Verrechnung der lokalen Karten zur globalen Karte zu erläutern, werden diese Odometriefehler vorerst vernachlässigt. Es wird von einer optimalen Odometrie ausgegangen.

Zunächst erfolgt eine einheitliche Ausrichtung der lokalen Karten, indem diese entsprechend der Roboterorientierung um ihren Mittelpunkt rotiert werden. Nachdem alle lokalen Karten gleichartig ausgerichtet sind, müssen die Positionen und damit die Überlappungen dieser „Teilkarten“ im globalen Raster ermittelt werden. Dazu werden die zu den lokalen Karten gehörigen Positionskoordinaten durch ganzzahlige Division durch die Zellgröße in Zellkoordinaten der globalen Karte umgerechnet und die lokalen Karten in das globale Raster eingepasst, wobei auch das globale Raster entsprechend angepasst wird.

Da die Belegungswahrscheinlichkeiten der Zellen der lokalen Karte mit zunehmenden Abstand zum Roboter immer unsicherer werden, ist es sinnvoll, den Abstand der Zellen bei der Fusion der Karten zu berücksichtigen. Belegungswerte von Zellen mit sehr kurzem Abstand zum Roboter müssen höher bewertet werden, als Belegungswerte weiter entfernter Zellen. Dies wird mit Gleichung 7.25 erreicht. Für jede Zelle der lokalen Karte wird dazu der Abstand zum Roboter berechnet (d in m) und daraus ein Faktor ermittelt der in Gleichung 7.26 eingesetzt wird. Mit dieser Gleichung wird aus dem aktuellen Belegungswert ($Prob(occ_{x,y})$) der Zelle abhängig vom Abstand zum Roboter die neue Belegungswahrscheinlichkeit ($Prob(occ_{x,y})_n$) berechnet. Für Zellen, deren Abstand null ist, ändert sich die Belegungswahrscheinlichkeit nicht. Für Zellen, die einen Abstand von 5 m oder mehr aufweisen, ergibt diese Gleichung eine Belegungswahrscheinlichkeit von 0,5 (unbestimmt). Die Belegungswahrscheinlichkeiten bewegen sich also mit zunehmendem Abstand stärker zum Wert 0,5, d.h. sie werden unbestimmter. Die Übertragungsfunktion in Gleichung 7.25 beschreibt dabei im Bereich von 0 m bis 5 m eine halbe Cosinusschwingung, die in y-Richtung verschoben und gestaucht ist, so dass bei 0 m ein

Wert von 1.0 und bei 5 m ein Wert von 0.0 erreicht wird.

$$factor = (\cos(d * \pi/5) + 1)/2 \quad (7.25)$$

$$Prob(occ_{x,y})_n = (Prob(occ_{x,y}) - 0,5) * factor + 0,5 \quad (7.26)$$

Um aus den lokalen Karten eine einzige konsistente globale Karte zu erhalten, werden für jede Zelle der globalen Karte die Belegungswahrscheinlichkeiten aller Zellen der lokalen Karten, die der selben Zelle der globalen Karte zugeordnet sind, nach folgender Gleichung verrechnet:

$$Prob(occ_{x,y}|s^{(1)}, s^{(2)}, \dots, s^{(T)}) = 1 - \left(1 + \frac{Prob(occ_{x,y})}{1-Prob(occ_{x,y})} \prod_{\tau=1}^T \frac{Prob(occ_{x,y}|s^{(\tau)})}{1-Prob(occ_{x,y}|s^{(\tau)})} \frac{1-Prob(occ_{x,y})}{Prob(occ_{x,y})}\right)^{-1} \quad (7.27)$$

$Prob(occ_{x,y})$ beschreibt die vorherige Belegungswahrscheinlichkeit der Zelle (x,y), welche anfangs mit 0.5 (unbekannt) initialisiert wird. $Prob(occ_{x,y}|s^{(\tau)})$ ist die Belegungswahrscheinlichkeit der Zelle (x,y), die sich aus der Sensorinformation zum Zeitpunkt τ ($s^{(\tau)}$) ergibt. Eine genauere Beschreibung und Herleitung dieser Gleichung ist z.B. in [Mor88] nachzulesen.

Diese Gleichung kann so umgestellt werden, dass man die globale Karte auch iterativ aufbauen kann und nicht erst, wenn alle lokalen Karten vorliegen. Dazu wird Gleichung 7.27 in die zwei Teilausdrücke 7.28 und 7.29 zerlegt. Die Ergebnisse von 7.28 werden parallel zu den Belegungswahrscheinlichkeiten der globalen Karte in einem Array abgelegt und mit jeder neuen lokalen Karte aktualisiert. Daraus kann dann nach 7.29 die Gesamtbelegungswahrscheinlichkeit berechnet werden.

$$Product(s^{(1)}, \dots, s^{(T)}) = \frac{Prob(occ_{x,y})}{1-Prob(occ_{x,y})} \prod_{\tau=1}^T \frac{Prob(occ_{x,y}|s^{(\tau)})}{1-Prob(occ_{x,y}|s^{(\tau)})} \frac{1-Prob(occ_{x,y})}{Prob(occ_{x,y})} \quad (7.28)$$

$$Prob(occ_{x,y}|s^{(1)}, \dots, s^{(T)}) = 1 - (1 + Product(s^{(1)}, \dots, s^{(T)}))^{-1} \quad (7.29)$$

Die Aufgabe von Gleichung 7.27 besteht darin, die aufgrund fehlerbehafteter Sonarmessungen teilweise recht unterschiedlichen Belegungswahrscheinlichkeiten der Zelle (x,y) zu einer einzigen zu verrechnen. Mit anderen Worten ergibt sich bei ermittelten Belegungswahrscheinlichkeiten, die eine leichte Tendenz zu einem der beiden Zustände „Hindernis“ bzw. „kein Hindernis“ aufweisen, auf Dauer gesehen eine sehr große Wahrscheinlichkeit für den entsprechenden Zustand. Dabei spielt es keine Rolle, wie gering diese Tendenz ist. Eine Belegungswahrscheinlichkeit von dauerhaft knapp über 0.5 (z.B. 0.50001) lässt die Gesamtwahrscheinlichkeit gegen 1 konvergieren und eine Wahrscheinlichkeit von knapp unter 0.5 lässt die Gesamtwahrscheinlichkeit gegen 0 konvergieren. Lediglich die Geschwindigkeit der Konvergenz wird von dem Betrag der Wahrscheinlichkeiten beeinflusst.

Um dem Problem zu begegnen, dass sich nach langer Aufintegration die Belegungswahrscheinlichkeiten der globalen Karte so sehr gefestigt haben, dass eine Anpassung der Karte an Veränderungen in der Umgebung in der Regel genauso lange dauert, wie der Roboter für den

aktuellen Zustand benötigt hat, werden die Zwischenergebnisse aus Gleichung 7.28 nach oben und unten begrenzt, da diese in der einen Richtung gegen Null und in der anderen gegen Unendlich konvergieren. Der Wert dieser Grenzen bestimmt die Geschwindigkeit des „Ablernens“ bereits gefestigter Belegungswerte, d.h. mit Hilfe dieser Parameter kann festgelegt werden, wie lange es dauert, bis sich Veränderungen in der bisher gelernten Umgebung auf die globale Karte auswirken.



Abbildung 7.18: Karte ohne (links) und mit Odometriekorrektur (rechts), Hindernisse sind schwarz, Freiraum grau und unexplorierte Bereiche weiß dargestellt (nach [BGKW00]).

Abbildung 7.18 stellt die globale Karte, die mit Hilfe des beschriebenen Odometriekorrekturverfahrens aufgebaut wurde, der globalen Karte ohne zusätzliche Mechanismen zur Odometriekorrektur gegenüber. Der Roboter legte dabei eine Strecke von ca. 200 m zurück. Im Startbereich, den der Roboter nach ca. 170 m wieder passierte, ist kein Versatz mehr zu erkennen. Der ermittelte Odometriefehler betrug nach Beendigung der Fahrt wenige cm, war also deutlich kleiner als die verwendete Gitterauflösung.

Auch nach Aufzeichnung einer Karte, die nahezu den gesamten BauMarkt umfasste (siehe Abbildung 7.19), wurde ein Gesamtodometriefehler gemessen, der kleiner als die Auflösung des Belegtheitsgitters (25×25 cm) war. Das dabei kartierte Gebiet hatte eine Größe von ca. $60 \text{ m} \times 100 \text{ m}$.

→ hier wünscht sich Klaus eine Grafik im Sinne eines PAP, um den roten Faden beim Aufbau der globalen Umgebungskarte besser zu verdeutlichen

7.2.3 Exploration versus manuelle Führung

Für einen autonomen Roboter sollte das selbstständige Erlernen seiner Umgebungsrepräsentation möglichst ohne notwendigen menschlichen Eingriff angestrebt werden. Das bedeutet, dass der Roboter seine Umgebung autonom explorieren muss. Für diese Aufgabenstellung wurden bislang verschiedene Algorithmen vorgeschlagen (z.B. [Thr93]). Allerdings birgt die autonome Exploration für den praktischen Einsatz eine Reihe von Problemen. Diese sind insbesondere:

- Die Exploration kann sehr lange dauern, was hinsichtlich der limitierten Funktionsfähigkeit der Akkumulatoren problematisch sein kann.

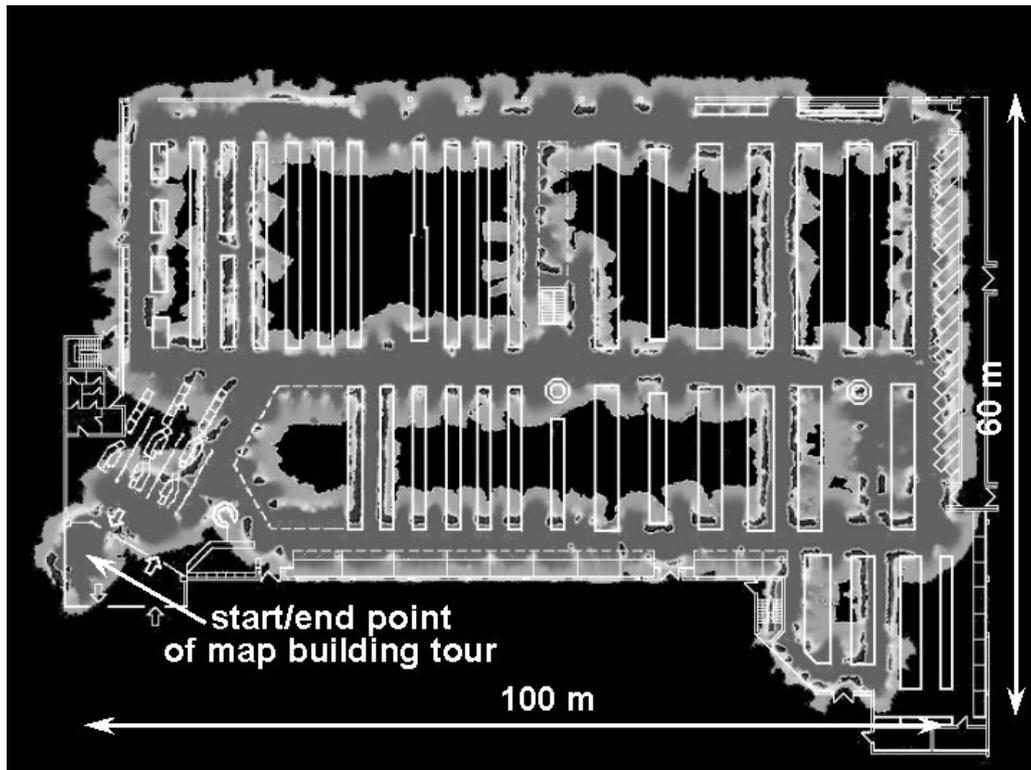


Abbildung 7.19: Eine erlernte probabilistische Karte, die sich über den gesamten BauMarkt erstreckt und die mit der vorliegenden Bauplan-Karte abgeglichen wurde, um den Eintrag von Artikelstandorten und Marktbereichen zu ermöglichen (aus [GB01]).

- Es kann nie hundertprozentig sichergestellt werden, dass wirklich das gesamte verfügbare Terrain kartiert wurde.
- Jedes derzeit bekannte Robotersystem verfügt über begrenzte sensorische Fähigkeiten, so dass nie davon ausgegangen werden kann, dass alle kritischen Situationen (z.B. abwärts führende Treppen, Ausfahrt vom Markt zum Parkplatz etc.) absolut sicher wahrgenommen werden können, weshalb man meist um die Per-Hand-Markierung von so genannten „verbotenen Bereichen“ nicht umhin kommt.

Aus den genannten Gründen wurde der manuellen Führung des Roboters für den Kartenaufbau mittels Joystick der Vorzug gegeben.

7.2.4 Probleme und Limitierungen

Das bislang beschriebene Verfahren eignet sich sehr gut zum Aufbau einer konsistenten Umgebungsrepräsentation, besitzt jedoch auch eine Reihe von zum Teil prinzipiellen Limitierungen, die nachfolgend diskutiert werden:

- Für den Kartenaufbau wird die Kenntnis der initialen Roboterpose vorausgesetzt, was aus rein pragmatischen Gründen auch durchaus sinnvoll ist.

- Erfasst der Roboter mit der zum Fußboden geneigten Kamera z.B. neben den für die Odometriekorrektur entscheidenden Fliesen noch weitere Bildbereiche, so liefert die Odometriekorrektur fehlerhafte Werte. Ist dies über einen längeren Zeitraum (mehrere Sekunden während der Fahrt) der Fall, so führt dies zwangsläufig dazu, dass der Roboter einen fehlerhaften aktuellen Zustand annimmt. Um diesem Fall zu begegnen, muss während des Kartenaufbaus sichergestellt werden, dass der vom Roboter analysierte Fußbodenbereich möglichst keine Störungen durch andere Objekte aufweist. Auch dies konnte während der experimentellen Untersuchungen stets garantiert werden.
- Es konnte zwar nachgewiesen werden, dass durch die Nutzung der visuell-basierten Odometriekorrektur die konsistente Kartierung des gesamten Einsatzareals möglich ist, dies darf jedoch nicht darüber hinwegtäuschen, dass keinerlei Kenntnis über den tatsächlichen aktuellen Fehler bezüglich der aktuellen Roboterpose vorliegt. Dies führt dazu, dass Fehler beim Kartenaufbau nur durch einen externen Beobachter festgestellt werden können. Auch dies ist unter rein pragmatischen Gesichtspunkten noch vertretbar.
- Schwerer wiegt dagegen die Tatsache, dass mit dem bislang beschriebenen Verfahren unmöglich ein Betrieb des Roboters über längere Zeit und damit längere Fahrstrecken bei hinreichend exakter Zustandsschätzung realisierbar ist. Dies implizierte unmittelbar die Entwicklung des im folgenden Abschnitt beschriebenen Ansatzes zum Aufbau eines visuell-basierten Umgebungsmodells in Verbindung mit leistungsfähigen und robusten Verfahren zur Zustandsschätzung.

7.3 Visuell-basierte Zustandsschätzung (Selbstlokalisierung)

Bevor in den folgenden Abschnitten das eigentliche Verfahren zur visuell-basierten Selbstlokalisierung detaillierter erläutert wird, sollen zunächst das Grundproblem der Roboter-Lokalisierung und die Motivation für den Einsatz eines visuellen Verfahrens dargestellt werden.

Angestrebt wird eine globale Lokalisierung des Roboters (siehe auch [Fox98]) bezüglich der Karte, die dieser mittels des vorab beschriebenen Verfahrens bereits erlernt hat. Lokalisierung bedeutet dabei die Schätzung des aktuellen Zustands des Roboters anhand der aktuell gewonnenen Sensordaten (Odometrie und externe Sensorik) und des (zeitinvarianten) Umgebungsmodells. Naheliegender wäre zunächst, die Ultraschallsensoren auch für die Lokalisation des Roboters zu benutzen. Analysiert man die für einen BauMarkt typische Struktur der Umgebung, so stellt man fest, dass diese durch eine in weiten Bereichen regelmäßige Anordnung der Regalreihen gekennzeichnet ist (siehe Abbildung 7.20).

Das würde bei einer ultraschall-basierten globalen Lokalisierung dazu führen, dass eine Vielzahl an Hypothesen von über Kartenpositionen (mit der entsprechenden Orientierung) existiert, an denen sich der Roboter potentiell befinden könnte. Auch die rekursive Schätzung des aktuellen Roboterzustands über die zurückliegenden Daten und Lokalisierungsergebnisse, wie dies im nachfolgend beschriebenen Verfahren der Fall ist, hilft zunächst nicht weiter, da sich bei einer Fahrt entlang eines Kurses durch zwei Regalreihen diese Mehrdeutigkeiten bezüglich des aktuellen Zustandes des Roboters auch über einen längeren Zeitraum nicht auflösen lassen. Aus den genannten Gründen erscheint es sinnvoll, visuelle Information als Grundlage für die globale Lokalisierung zu nutzen, da die hier gegebene Vielfalt an sensorischen Eindrücken das

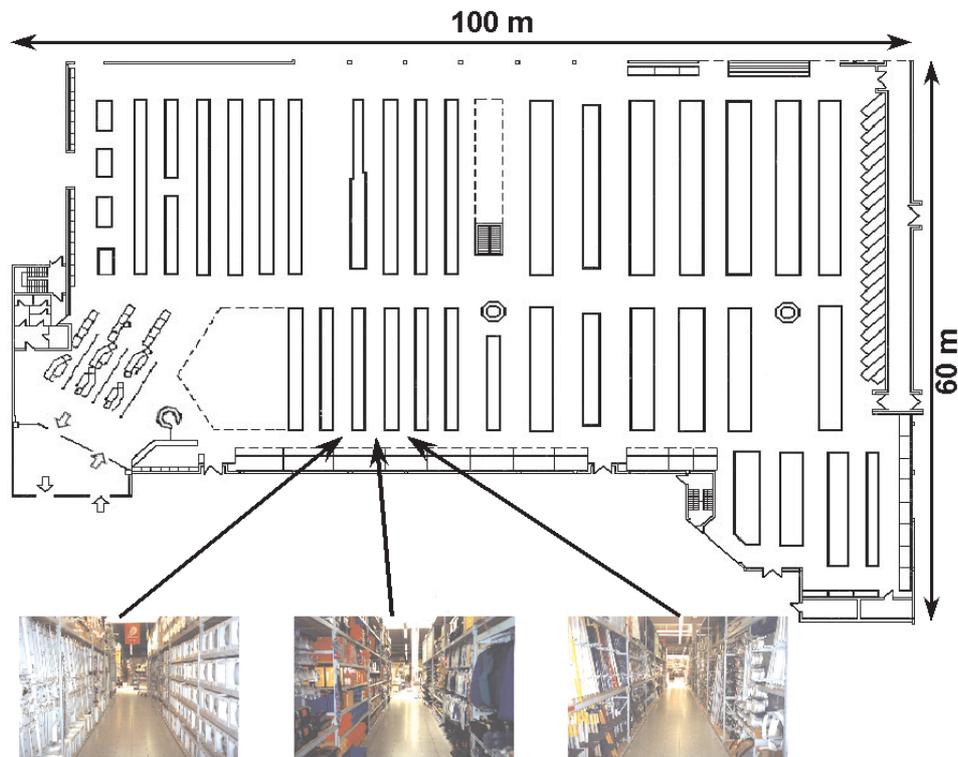


Abbildung 7.20: Motivation der visuell-basierten globalen Navigation. Für die drei markierten Positionen des BauMarktes würde sich auf der Basis der Ultraschallinformation ein sehr ähnlicher sensorischer Eindruck ergeben, was im Falle einer notwendigen globalen Selbstlokalisierung zu vielen Mehrdeutigkeiten bezüglich des aktuell vom Roboter eingenommenen Zustandes führen würde. Betrachtet man hingegen die visuellen Ansichten, so lassen sich diese drei Positionen sehrwohl anhand dieser Information unterscheiden. Genau dieser Umstand soll für die angestrebte globale Navigation des Shopping-Assistenten ausgenutzt werden.

Auftreten der o.g. Mehrdeutigkeiten wirkungsvoll verhindern kann (siehe Abbildung 7.20).

7.3.1 Markov-Lokalisation und Monte-Carlo-Verfahren

Den Ausgangspunkt bildet ein unter dem Begriff der Markov-Lokalisation bekanntes Verfahren, welches in [Fox98] ausführlich behandelt und diskutiert wird. Dabei handelt es sich um einen auf die konkrete Robotik-Problemstellung übertragenen, teilweise beobachtbaren Markov-Entscheidungsprozess (Partially Observable Markov Decision Process, POMDP), in dem der Roboter ein in seiner Umwelt agierender Agent ist, wobei der Zustand der Welt durch den Zustand des Roboters innerhalb seiner Umgebung determiniert wird. Lokalisierung bedeutet, wie oben bereits betont, die Schätzung des aktuellen Zustands des Roboters anhand der Sensordaten (externe Sensorik) und der Aktionen des Roboters (Odometrie) bezüglich des Umgebungsmodells.

Das Monte-Carlo-Verfahren stellt eine Methodik zur Verfügung, diesen Schätzungsprozess effizient zu implementieren, was eine Grundvoraussetzung für dessen Anwendung in der mobilen Robotik darstellt.

Die nachfolgend verwendete Notation orientiert sich an [Fox98], wobei eine Reihe von Begriffen nicht ins Deutsche übersetzt werden.

7.3.1.1 Markov-Lokalisation

Der Zustand L des Roboters zum Zeitpunkt t sei beschrieben durch eine Zufallsvariable L_t , die sowohl die x, y -Position als auch die Orientierung ϕ beschreibt. Der Belief $Bel(L_t = l)$ repräsentiert, mit welcher Wahrscheinlichkeit der Roboter annimmt, zum Zeitpunkt t in Zustand l gewesen zu sein, und spannt eine Wahrscheinlichkeitsdichtefunktion über alle L auf. $Bel(L_t = l)$ wird mit jedem neuen sensorischen Eindruck S bzw. bei jeder Aktion A des Roboters aktualisiert.

Im allgemeinen Fall hängt der aktuelle Belief von allen vorangegangenen Beobachtungen und Aktionen des Roboters nach $Bel(L_t | S_1, \dots, S_t, A_1, \dots, A_{t-1})$ ab, wobei die Berechnungskomplexität exponentiell mit der Zahl der Vorbedingungen wächst. Um diese Berechnungskomplexität einzuschränken, werden zwei Unabhängigkeitsannahmen (Markov-Annahmen) eingeführt, die besagen, dass für die Prädiktion des Zustands L_t die Kenntnis des Zustands L_{t-1} und der dort ausgeführten Aktion A_{t-1} ausreichend ist (Unabhängigkeit der Aktionen), und dass die sensorische Beobachtung S_t nur vom Zustand der Welt zum Zeitpunkt t abhängt (Unabhängigkeit der Wahrnehmung). Damit sind alle Aspekte der Welt durch die Zustandsvariable L und die bedingten Wahrscheinlichkeiten $P(L | L_{t-1}, A_{t-1})$ und $P(S_t | L_t)$ modellierbar.

Den Kern der Markov-Lokalisation bilden drei Komponenten: das Bayes-Filter, das Sensormodell (Wahrnehmungsmodell) und das Bewegungsmodell.

Bayes-Filter Die Rekursionsvorschrift für das Bayes-Filter ergibt sich aus der Anwendung des Bayes-Theorems zur Schätzung von A-posteriori-Wahrscheinlichkeiten und den o.g. Markov-Annahmen und lautet:

$$Bel(L_t = l) = \eta P(s_t | l) \int P(l | L_{t-1}, a_{t-1}) \cdot Bel(L_{t-1}) dL_{t-1} \quad (7.30)$$

Dabei repräsentiert η einen Normierungsterm mit $\eta = P(s_t | a_{t-1}, \dots, s_0)^{-1}$, der sicherstellt, dass die Summe über alle Beliefs den Wert 1 ergibt. In Gleichung 7.30 stellt der Term $P(s_t | l)$ das Sensormodell, der Term $P(l | L_{t-1}, a_{t-1})$ das Bewegungsmodell und $Bel(L_{t-1})$ den Belief zum unmittelbar vorangegangenen Zeitpunkt dar.

Kleinbuchstaben für die Aktionen bzw. die sensorischen Wahrnehmungen deuten an, dass es sich dabei um Messungen handelt, die entsprechend der nachfolgenden Abschnitte modelliert werden.

Bewegungsmodell Das Bewegungsmodell stellt eine probabilistische Beschreibung der Roboterkinematik in Form einer Wahrscheinlichkeitsverteilung über mögliche Folgezustände des Roboters dar, die dieser ausgehend von einem Initialzustand erreichen kann. Abbildung 7.21 stellt das Bewegungsmodell, das physikalisch begründete Unsicherheiten bei der Roboterbewegung berücksichtigt, für zwei Aktionen des Roboters B21 nach [FDBT99] dar. Dieses Modell wird experimentell bestimmt und nachfolgend als zeitinvariant angenommen. Da es sich bei der gegenwärtig verwendeten Experimentalplattform PERSES um den gleichen Robotertyp handelt,

gilt das dargestellte Bewegungsmodell ebenso wie das folgende Sensormodell zumindest qualitativ auch hier.

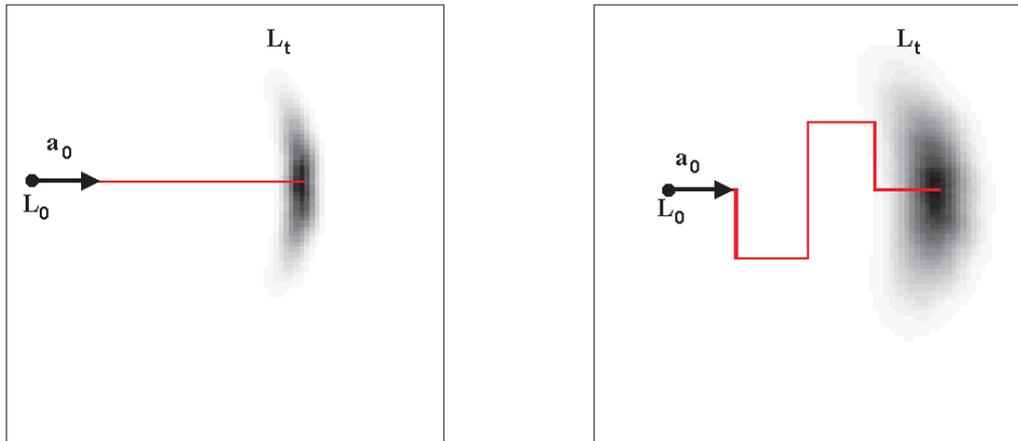


Abbildung 7.21: Probabilistische Bewegungsmodelle für eine geradlinige Fahrt über 40m (links) und eine Sequenz aus Translation und Rotation über insgesamt 80m (rechts) (nach [FDBT99]).

Sensormodell Das Sensormodell stellt wiederum eine probabilistische Beschreibung in Form einer Wahrscheinlichkeitsverteilung über mögliche Observations dar. Auch dieses zeitinvariante Modell, das in Abbildung 7.22 illustriert wird, muss vorab experimentell bestimmt werden. Alternativ dazu besteht die Möglichkeit, das Sensormodell mittels eines neuronalen Netzes zu erlernen, wie dies im Abschnitt 7.1.2 beschrieben wurde. Es berücksichtigt in erster Linie das Rauschverhalten des Sensors und gegebenenfalls auch externe Störungen.

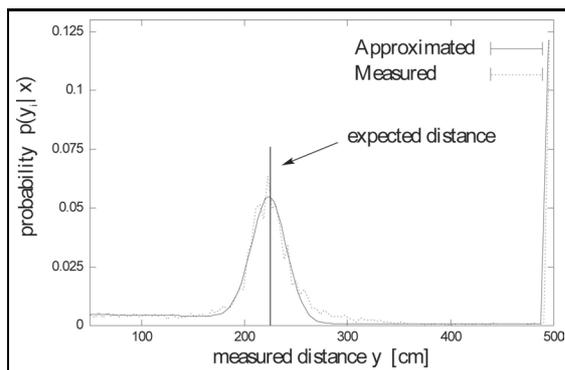


Abbildung 7.22: Sensormodell für einen Ultraschallsensor (aus [FDBT99]).

7.3.1.2 Monte-Carlo-Lokalisation und Condensation-Algorithmus

Die Wahrscheinlichkeitsverteilung $Bel(L_t = l)$ stellt eine Funktion über dem gesamten Zustandsraum dar (Positionen in der Karte). Die Berechnungskomplexität für die kontinuierliche Aktualisierung dieser Verteilungsfunktion wächst mit zunehmender Dimension des Bewegungsmodells exponentiell an. Gesucht sind also Verfahren, die eine effiziente Implementation

ermöglichen, ohne dabei die Funktion über deren gesamten Definitionsbereich in jedem Schritt neu berechnen zu müssen.

ISARD und BLAKE [IB98] schlagen im Kontext des visuellen Trackings von Objekten einen unter dem Namen CONDENSATION (conditional density propagation) bekannt gewordenen Algorithmus vor, der trotz einer spärlichen Abtastung der Wahrscheinlichkeitsverteilung $Bel(L_t = l)$ eine hinreichend akkurate und robuste Approximation dieser Verteilungsfunktion ermöglicht. Das Verfahren wurde mittlerweile auch erfolgreich auf die Problematik der Roboterlokalisierung übertragen [FDBT99]. Es bezieht seine Leistungsfähigkeit in erster Linie durch eine gewichtete Abtastung der Verteilungsfunktion, was dazu führt, dass Bereiche mit hohem Funktionswert dicht und Bereiche mit niedrigem Funktionswert entsprechend grober abgetastet werden. Ein weiterer Vorteil des Condensation-Algorithmus liegt in seiner Fähigkeit, beliebig geformte Verteilungsfunktionen approximieren zu können, was gerade für die Roboterlokalisierung mit den hier nahezu zwangsläufig auftretenden multiplen Positionshypothesen von Bedeutung ist. Abbildung 7.23 veranschaulicht zunächst die prinzipielle Idee zur Abtastung des aktuellen Belief entlang einer Bewegungssequenz des Roboters.

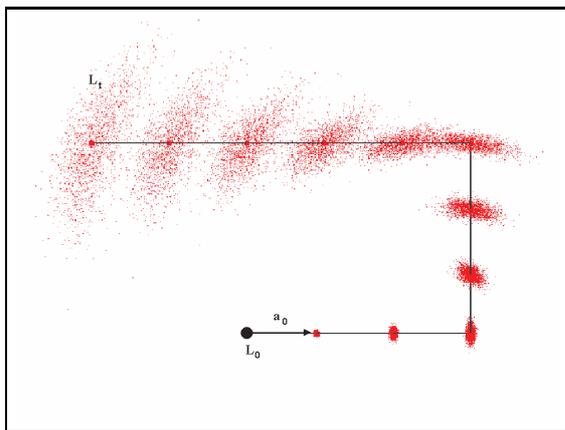


Abbildung 7.23: Abtastung des aktuellen Belief entlang einer Bewegungssequenz des Roboters (in Anlehnung an [Thr00b]).

Abbildung 7.24 soll das Grundprinzip des Condensation-Algorithmus anhand einer eindimensionalen Verteilungsfunktion verdeutlichen. Ziel des Verfahrens ist es, eine unbekannte, möglicherweise multimodale Verteilungsfunktion $f(x, t)$ zu approximieren, ohne diese im gesamten Definitionsbereich berechnen zu müssen. Diese Verteilungsfunktion repräsentiert die Wahrscheinlichkeit, dass sich das zu verfolgende Objekt zum Zeitpunkt t an Position x befindet. Begonnen wird mit einer definierten (ausreichend großen) Anzahl an Stützstellen, an denen der Funktionswert $f(x, t)$ berechnet wird, wobei die initiale Verteilung der Stützstellen (initialer Belief) von A-priori-Annahmen abhängt. Ohne A-priori-Wissen sind alle Positionen gleich wahrscheinlich, was sich in einer homogenen initialen Verteilung der Stützstellen über dem gesamten Definitionsbereich niederschlägt (Abbildung 7.24 Schritt 1). Liegen bereits Hypothesen über mögliche Positionen vor, können diese durch entsprechend dichtere initiale Platzierung der Stützstellen berücksichtigt werden. Anschließend erfolgt die erste Gewichtung der Stützstellen anhand des Wahrnehmungsmodells (Abbildung 7.24 Schritt 2), welches bestimmt, mit welcher Wahrscheinlichkeit sich der Roboter in einem bestimmten Zustand befindet oder umgekehrt, wie wahrscheinlich die aktuelle Beobachtung für alle möglichen Zustände ist. Bewegt sich der Roboter, wird dessen Bewegung auch auf die Stützstellen entsprechend des Bewegungsmodells übertragen (Abbildung 7.24 Schritt 3). Dabei ist die Wahrscheinlichkeit, dass eine Stützstelle

auch im Folgeschritt für die Approximation verwendet wird, proportional zu ihrem Gewichtswert im vorhergehenden Abtastschritt, was dazu führt, dass die Verteilungsfunktion vorrangig an den „wichtigen“ Stellen approximiert wird, an denen der Belief bereits einen hohen Wert erreicht hat. Nun schließt sich wiederum ein Bewertungsschritt für alle Stützstellen an (Abbildung 7.24 Schritt 4). Zu beachten ist, dass die Anzahl der Stützstellen über alle Folgeschritte konstant bleibt, und dass die Bewertungen der Stützstellen nach jedem Aktualisierungsschritt so normiert werden, dass ihre Summe den Wert eins ergibt.

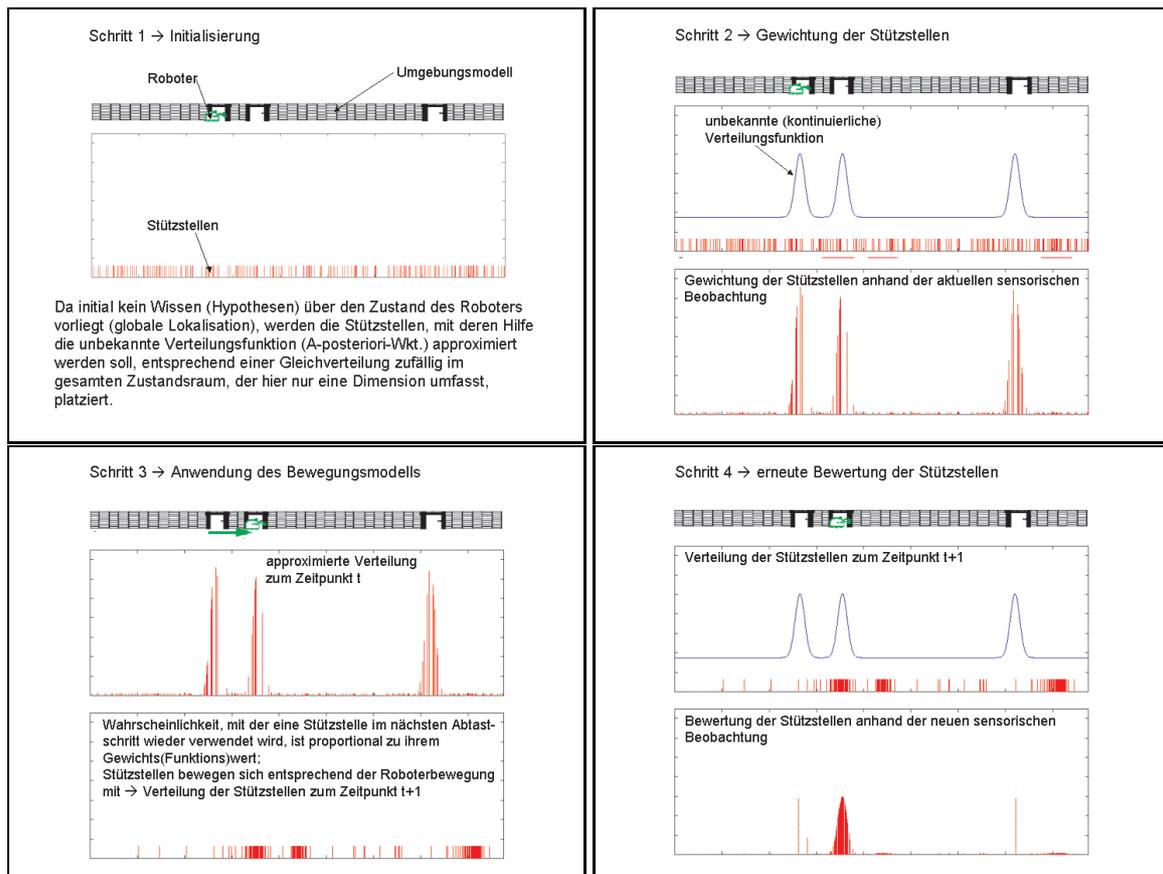


Abbildung 7.24: Teilschritte des Condensation-Algorithmus für eine eindimensionale Verteilungsfunktion (in Anlehnung an [Thr00b]). Eine detailliertere Erläuterung erfolgt im Text.

Abschließend sei an dieser Stelle darauf hingewiesen, dass es sich bei einigen der in den nachfolgenden Abschnitten gezeigten Verteilungen jeweils um Projektionen eines dreidimensionalen Zustandsraumes auf eine zweidimensionale Darstellung handelt.

7.3.2 Visuelle Monte-Carlo-Lokalisation (MCL)

Abbildung 7.25 gibt einen Überblick über das Gesamtsystem zur visuellen MCL (siehe auch [KGB01]). Dargestellt sind die wesentlichen Teilsysteme für den MCL-Algorithmus, die z.B. die Vorverarbeitung der sensorischen Daten (Umweltwahrnehmung) und die Bereitstellung eines Umweltmodells übernehmen.

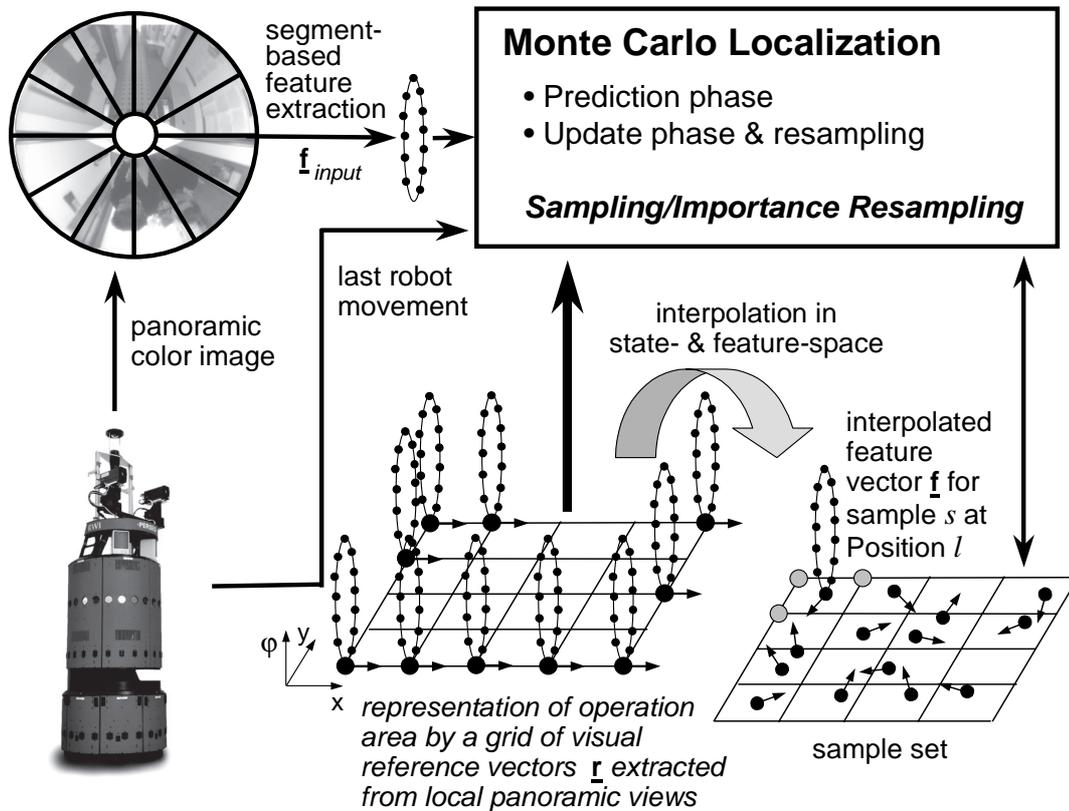


Abbildung 7.25: Überblick über die Gesamtarchitektur zur visuellen Monte-Carlo-Lokalisation. Der Ansatz basiert auf einem gridbasierten Umgebungsmodell, welches mit den visuellen Sensoreindrücken an den diskreten Gitterpunkten attribuiert wird (aus [GB00]).

Den Ausgangspunkt bildet ein gridbasiertes Umgebungsmodell. Dieses wurde für die ersten experimentellen Untersuchungen, die hier beschrieben werden (siehe auch [GB00] [KKG00] [KGB01]), noch vorgegeben und soll im nächsten Schritt mit dem erlernten Umgebungsmodell (siehe Verfahren aus Abschnitt 7.2) kombiniert werden. An allen Gitterpunkten des Umgebungsmodells erfolgt jetzt die Aufnahme eines omnidirektionalen Bildes durch die 360°-Kamera, wobei der Roboter an jedem Aufnahmepunkt die gleiche Orientierung bezüglich der Umgebungskarte aufweist. Um die Dimension der Daten zu reduzieren, erfolgt jetzt die Transformation des Bildes in einen zyklischen Merkmalsvektor (siehe Abschnitt 7.3.2.1), mit dem dann der entsprechende Gitterpunkt attribuiert wird. Anhand der Odometriedaten des Roboters und der jeweils gewonnenen aktuellen omnidirektionalen Bilder kann dann der eigentliche MCL-Algorithmus ablaufen.

Die folgenden Abschnitte erläutern detailliert die einzelnen Verarbeitungsschritte, bevor dann abschließend erste experimentelle Ergebnisse zur visuellen MCL präsentiert werden.

7.3.2.1 Visuelle Umgebungsrepräsentation

Zunächst wird das omnidirektionale Bild über eine Polarkoordinatentransformation in ein kartesisches Bild umgewandelt, um die nachfolgenden Verarbeitungsschritte zu vereinfachen (siehe

Abbildung 7.26).

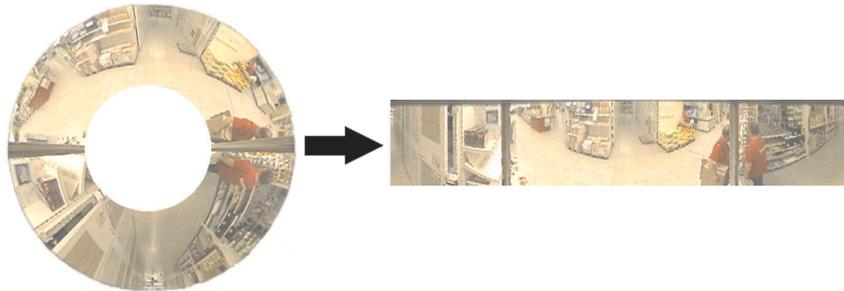


Abbildung 7.26: Transformation des omnidirektionalen Bildes in ein kartesisches Bild.

Für die Auswahl geeigneter bildbeschreibender Merkmale waren die folgenden Kriterien maßgeblich:

- Die Merkmale sollten sich möglichst einfach und damit effizient bestimmen lassen, um eine hinreichende Performanz des gesamten MCL-Verfahrens zu ermöglichen.
- Hinsichtlich ihrer Invarianzeigenschaften ist eine Abhängigkeit der Merkmalsbeschreibung von der Roboterorientierung einer Rotationsinvarianz vorzuziehen, da so neben der x, y -Position auch die Orientierung mit geschätzt werden kann.
- Die Merkmalsbeschreibung sollte robust gegenüber Störungen sein, wie sie z.B. durch Personen in der unmittelbaren Umgebung des Roboters hervorgerufen werden können.

Um all den genannten Kriterien Rechnung zu tragen, wurde eine Unterteilung des kartesischen Bildes in zehn gleich große Segmente vorgenommen (siehe Abbildung 7.27), in denen dann jeweils der mittlere Farbwert der Farbwerte pro Farbband im RGB-Farbraum als Merkmale für dieses Segment bestimmt werden.



Abbildung 7.27: Einteilung des omnidirektionalen Bildes in zehn Segmente (nach [KGB01]).

Fasst man die pro Bild(Kreis)segment berechneten Merkmale als eine Komponente auf, liegt an jedem Gitterpunkt des Umgebungsmodells ein zehndimensionaler, zyklischer Merkmalsvektor vor, der fortan als Referenzvektor bezeichnet werden soll und die dort aufgenommene visuelle Szene beschreibt. Um die verschiedenen möglichen Orientierungen des Roboters als dritte Dimension des Zustandsraumes des Roboters zu berücksichtigen, müssen die Referenzvektoren lediglich um den entsprechenden Winkel rotiert werden.

Weiterhin wurde experimentell untersucht, ob auf Basis der Referenzvektoren eine Gewinnung von Sensoreindrücken für Positionen zwischen den Gitterpunkten möglich ist. Dabei wurde festgestellt, dass bis zu einem Gitterabstand von 90×90 cm eine lineare Interpolation hinreichend exakte Ergebnisse liefert.

Um sowohl für die Rotation als auch die Interpolation der Referenzvektoren eine möglichst stetige Kodierung zu erreichen, erfolgt die oben beschriebene segmentweise Berechnung der

Merkmale mittlerer Grauwert und Varianz gewichtet mit einer Gaußfunktion. Damit einher geht eine Überlappung benachbarter Segmente, und auch starke Änderungen des Bildinhalts führen nicht zu extremen Unstetigkeiten in der Merkmalskodierung.

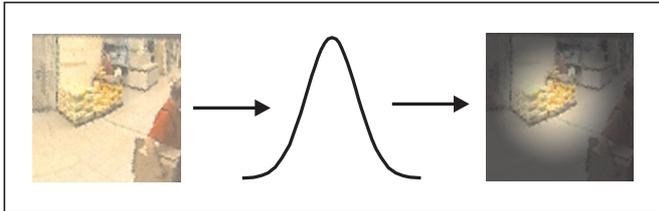


Abbildung 7.28: Gaußförmig gewichtete, segmentweise Bestimmung der bildbeschreibenden Merkmale (nach [KGB01]).

7.3.2.2 Visuell-basiertes Sensormodell

Für den visuellen Sensor muss nun noch ein entsprechendes Sensormodell $P(s_t | L_t = l)$ bestimmt werden, was sich als nicht trivial erweist, da hier mehrere Faktoren eine Rolle spielen. Das aufgenommene Bild hängt vom Pixelrauschen, der Fokus- und der Blendenautomatik, dem automatischen Weißabgleich und noch einigen anderen Parametern der Kamera ab, welche in der Regel nicht unmittelbar erfasst oder berechnet werden können. Desweiteren gibt es noch andere äußere Einflüsse, wie z.B. die Schwankung des Tageslichtes in der Helligkeit und der spektralen Zusammensetzung, welche im allgemeinen die Hauptursachen für unterschiedliche Umwelteindrücke s_t in ein und dem selben Zustand $L_t = l$ sind. Um dennoch zumindest ein einfaches Sensormodell zu bestimmen, wurden experimentelle Untersuchungen über die Änderung der szenenbeschreibenden Merkmale durchgeführt. Der Roboter befand sich dabei an einer festen Position mit konstanter Orientierung ($l = konstant$), und über einen Zeitraum von mehreren Stunden wurden die sensorischen Inputs s_t aufgezeichnet. Mit diesen Daten wurde die in Abbildung 7.29 dargestellte Verteilung berechnet.

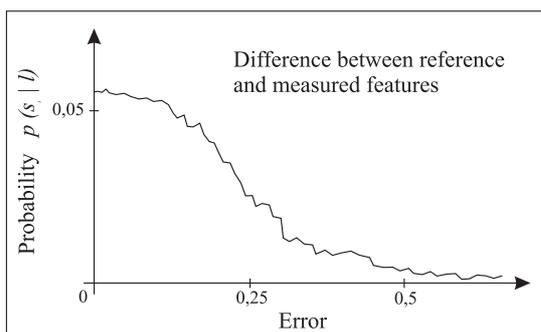


Abbildung 7.29: Wahrscheinlichkeitsverteilung $P(s_t | l)$ für alle im Zustand l aufgenommenen Observationen (Bilder bzw. Merkmalsvektoren) s_t . Der experimentell ermittelte Verlauf (Euklidischer Abstand zwischen allen aufgezeichneten Merkmalsvektoren und dem Referenz-Merkmalvektor für Zustand l) kann durch eine entsprechend parametrisierte Gaußfunktion einfach und effizient modelliert werden (nach [KGB01]).

Die Bewertung der Stützstellen erfolgt dann entsprechend des Sensormodells über den Euklidischen Abstand zwischen dem an der Stützstelle erwarteten Merkmalsvektor und dem anhand der tatsächlichen Beobachtung ermittelten Merkmalsvektor.

7.3.2.3 Bewegungsmodell

Das experimentell ermittelte Bewegungsmodell entspricht qualitativ dem, welches in Abbildung 7.21 bereits dargestellt wurde, da es sich bei PERSES um den gleichen Robotertyp (RWI

B21) handelt.

Sensor- und Bewegungsmodell sind für die Monte-Carlo-Lokalisation von entscheidender Bedeutung. Erst mit der Kenntnis der Unsicherheit bezüglich Wahrnehmung und Bewegung ist eine robuste Lokalisation möglich.

7.3.2.4 Experimentelle Ergebnisse

Abbildung 7.30 zeigt das Ergebnis eines Lokalisationsexperimentes. Der im Umgebungsmodell erfasste Bereich erstreckte sich über drei Regalreihen der realen BauMarkt-Umgebung (ca. 28×17 Meter, 1363 Gitterpunkte beziehungsweise Referenzvektoren). In dem hier beschriebenen Experimenten wurde ein Gitterabstand von 60×60 cm verwendet.

Da initial kein Wissen über die Position des Roboters vorlag, wurden alle Stützstellen entsprechend uniform im Zustandsraum verteilt (Abbildung 7.30 (a)). Nachdem der Roboter seinen ersten Sensoreindruck wahrgenommen hat, kann der initiale Belief bezüglich aller möglichen Positionen ermittelt werden, mit dem dann eine Gewichtung der Stützstellen möglich wird. Stützstellen mit niedrigem Gewicht werden gelöscht und in der Umgebung von Stützstellen mit hohem Gewicht eingefügt. Anschließend erfolgt die erste Bewegung des Roboters, die entsprechend dem Bewegungsmodell auf alle Stützstellen übertragen wird, der mit dem neuen Sensoreindruck entsprechend des Sensormodells zu einem neuen Belief verrechnet wird und zu einer erneuten Gewichtung der Stützstellen führt. Abbildung 7.30 verdeutlicht, wie die iterative Anwendung dieser Prozedur sukzessive zu einer Konzentration der Stützstellen an den vom Roboter angenommenen wahrscheinlichsten Positionen führt, bis nach 19 Bewegungs- und Wahrnehmungszyklen (Gesamtfahrstrecke ca. 6m) die korrekte Position mit einer Abweichung von ca. 7cm ermittelt werden konnte.

Ein weiteres Experiment, bei dem zwischen Aufnahme des Umgebungsmodells und der eigentlichen Lokalisation etwa 2 Monate (!) lagen, soll die Robustheit des vorgestellten Ansatzes verdeutlichen: Nach etwa der gleichen Zahl an Bewegungsschritten konnte ausgehend von einem initial homogen verteilten Belief ebenfalls eine korrekte Zustandsschätzung erreicht werden, wobei hier der verbleibende Fehler etwa 15cm betrug. Obwohl aus der großen Zeitspanne zwischen der Erstellung des Umgebungsmodells und dem eigentlichen Experiment neben den veränderten Beleuchtungsbedingungen auch Veränderungen in der BauMarkt-Umgebung selbst resultiert haben dürften, lag die Genauigkeit der Zustandsschätzung zumindest nicht um Größenordnungen schlechter als in dem in Abbildung 7.30 dargestellten Fall. Auf eine analoge Darstellung dieses Experiments wurde verzichtet, da sich grafisch keine erkennbaren Unterschiede zu Abbildung 7.30 ergeben würden.

Abschließend sei noch auf einen weiteren erwarteten Vorteil der verwendeten Merkmalskodierung für die Bildsegmente hingewiesen, der ebenfalls experimentell manifestiert werden konnte: durch die sehr grobe Merkmalskodierung in Verbindung mit der gaußförmigen Gewichtung hat die während der Roboterbewegung auftretende Bewegungsunschärfe kaum Einfluss auf die Zustandsschätzung. Damit kann das Verfahren auch während der Fahrt kontinuierlich zum Einsatz kommen. Würde man stattdessen eine Beschreibung der Bildsegmente verwenden, die jeweils scharfe (fokussierte) Bilder voraussetzt, müsste der Roboter für jeden Schritt der Zustandsschätzung jeweils anhalten und gegebenenfalls noch die Zeit bis zur automatischen

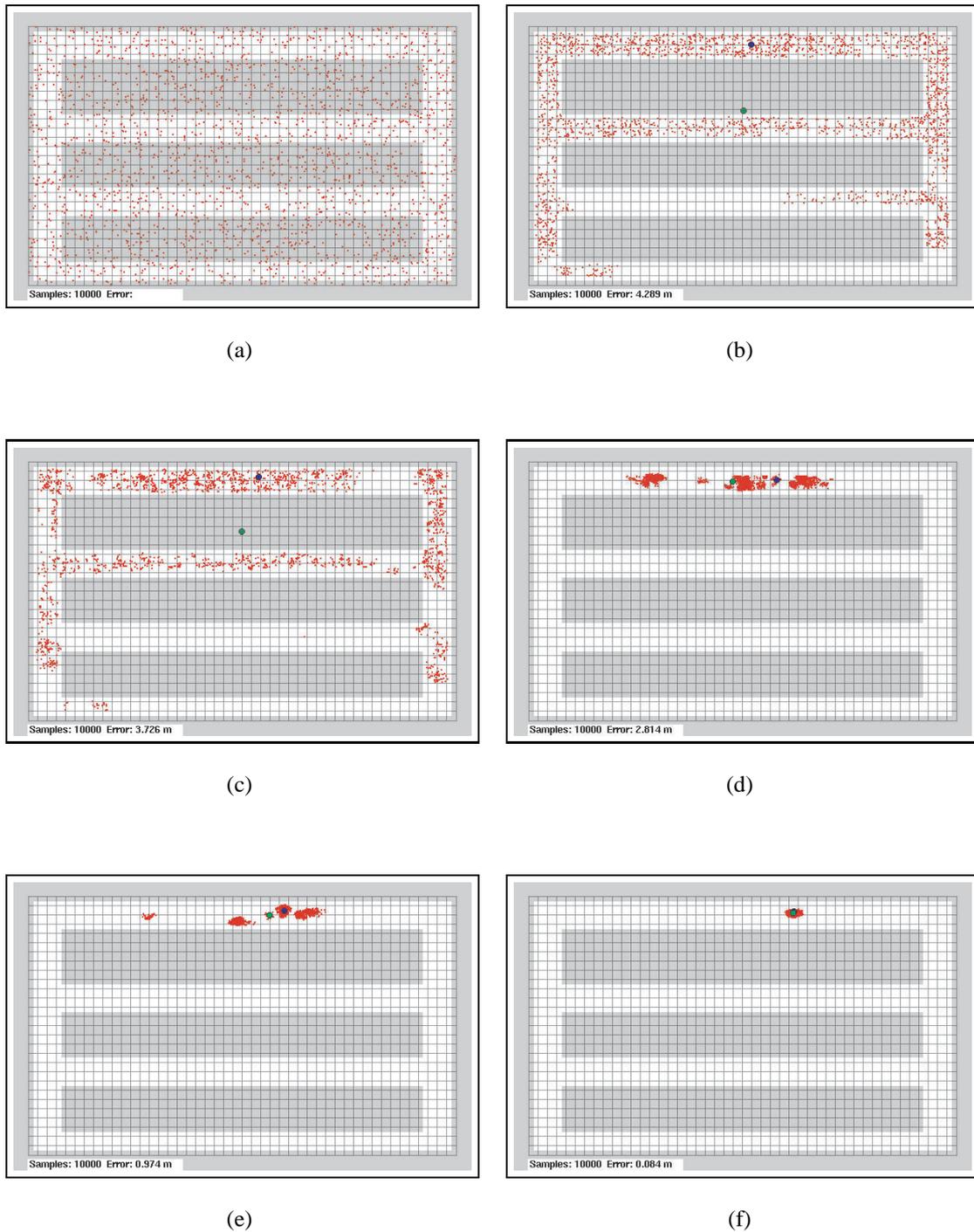


Abbildung 7.30: Sequenz einer visuell-basierten Monte-Carlo-Lokalisation; blauer Punkt → korrekte Position des Roboters; grüner Punkt → geschätzte Position; rote Punkte markieren die Stützstellen, mit denen die Verteilungsfunktion abgetastet wird. (a) Anfangssituation; (b) nach einem Schritt; (c) nach 2 Schritten; (d) nach 13 Schritten; (e) nach 15 Schritten; (f) Position korrekt geschätzt nach 19 Schritten (Positionsfehler ca. 10cm).

Fokussierung der Kamera bis zur Bildakquisition abwarten, was zu enormen Einschränkungen beim praktischen Einsatz des Verfahrens führen dürfte.

7.3.2.5 Probleme und Limitierungen

Obwohl das Verfahren zur visuell-basierten Zustandsschätzung in den ersten experimentellen Untersuchungen seine Robustheit unter Beweis stellen konnte, stehen für eine abschließende Bewertung des Verfahrens noch eine Reihe von Untersuchungen aus. Diese sind nachfolgend kurz zusammengefasst.

- Zunächst muss das in Abschnitt 7.2 beschriebene Verfahren zum Aufbau einer globalen Umgebungsrepräsentation erweitert werden, um dessen Attributierung mit den entsprechenden visuellen Ansichten (Merkmalen) zu realisieren.
- Die bisherigen Experimente bezogen sich lediglich auf einen sehr begrenzten Marktbe- reich. Weitere Untersuchungen sollen zeigen, wie das Verfahren skaliert, wenn das ge- samte BauMarkt-Areal einbezogen wird.
- Die verwendete Interpolation der Merkmalsvektoren für die Generierung des erwarteten sensorischen Inputs an Positionen zwischen den Gitter(referenz)punkten stellt derzeit den aufwendigsten Berechnungsschritt dar. Aktuell laufende Arbeiten versuchen, eine optimale Balance zwischen der Dichte der Referenzvektoren und der Interpolation der Merkmalsvektoren zu ermitteln.
- Auch die Variabilität der visuellen Information stellt noch ein gewisses Problem dar. Deshalb werden gegenwärtig weitere Untersuchungen zu besser geeigneten Merkmalen durchgeführt, die auch unter unterschiedlichsten Beleuchtungsbedingungen noch besser reproduzierbar sind.

7.4 Pfadplanung und Bewegungssteuerung

Da der Roboter sich autonom und selbständig in seinem Einsatzszenario zurechtfinden soll, muss es ihm möglich sein, beliebige Punkte der Umgebung anzufahren. („Beliebige Punkte“ be- zieht sich hierbei zunächst lediglich auf Welt-Koordinaten, eine etwaige Erkennung von Land- marken etc. wird nicht betrachtet). Voraussetzung dafür ist natürlich, dass zuvor durch eigen- ständige oder durch geführte Exploration (siehe Abschnitt 7.2.3) eine Karte der Umgebung aufgebaut wurde, welche sowohl das jeweilige Navigationsziel als auch (mindestens) einen möglichen Weg von der aktuellen Position dorthin enthält.

7.4.1 Pfadplanung

Zunächst soll das Problem betrachtet werden, den kürzesten Pfad zum festgelegten Ziel zu finden. Nachfolgend werden dann Überlegungen zu zusätzlichen Nebenbedingungen bzgl. des gewünschten Verhaltens angeführt und in die Pfadplanung einbezogen.

Ausgangspunkt der Pfadplanung sind folgende bekannten Daten: Eine Grid-basierte Karte, in der jede Zelle einen Belegtheitswert $0 \leq occ(x, y) \leq 1$ besitzt (bzw. -1 für „unbekannt“),

sowie die Koordinaten der aktuellen Position und des Zieles, welche sich jeweils auf eine Zelle dieser Karte abbilden lassen. Als Resultat soll ein Pfad als Abfolge von zu durchfahrenden Zellen berechnet werden. Mit diesem Pfad ist es dann möglich, an jeder Position auf dem Weg zum Ziel die zu steuernde Fahrtrichtung mit geringem Aufwand und daher schnell zu berechnen.

7.4.1.1 Dijkstra-Algorithmus

Der Dijkstra-Algorithmus ist ein bekannter Algorithmus der Graphentheorie, der das *Single Source Shortest Paths Problem* für gewichtete Graphen löst. Er erhält als Eingabe einen gewichteten Graphen $G = (V, E)$ sowie einen ausgezeichneten Knoten $v_0 \in V$ und berechnet für alle Knoten $v \in V$ den kürzesten Pfad zwischen v und v_0 . Zu beachten ist dabei, dass die Kantengewichte niemals negativ sein dürfen. Negative Kantengewichte können zu undefinierten kürzesten Pfaden führen, wenn Kreise mit Gesamtlänge < 0 existieren. Dadurch kann jeder Pfad zwischen 2 Punkten beliebig „kurz“ (negative Länge) werden, indem dieser Kreis beliebig oft durchlaufen wird.

Da die erlernte Karte im Gegensatz zu topologischen Modellen nicht unmittelbar als Graph-Struktur vorliegt, muss diese in einen entsprechenden Graphen überführt werden. Dazu werden die Zellen der Karte in „freie“ und „belegte“ Zellen unterteilt. Alle freien Zellen bilden jetzt die Knoten des Graphen, wobei Zellen dann als frei betrachtet werden, wenn diese einen Belegtheitswert ≤ 0.5 aufweisen. Zellen mit höherem Belegtheitswert und unbekannte Zellen sind folglich nicht Teil des Graphen. Kanten werden jeweils zwischen 2 Knoten eingefügt, die direkt (horizontal/vertikal/diagonal) benachbart sind. Das Gewicht einer Kante zwischen 2 diagonalen „Nachbarknoten“ ist dabei jeweils 1.4, das Gewicht einer anderen Kante 1, um das tatsächliche Verhältnis der Entfernungen zwischen diesen Zellen abzubilden. Da davon ausgegangen werden kann, dass immer ein Weg von einer freien Zelle in eine benachbarte freie Zelle existiert, entspricht dieser Graph einer diskretisierten Darstellung der erreichbaren Positionen und der möglichen Übergänge in unserer realen Umgebung.

Der Ansatz geht davon aus, dass Wege innerhalb von Zellen nicht berechnet werden müssen. In gewisser Weise wird eine einzelne Zelle als geometrischer Punkt abstrahiert. Bei genügend kleiner Ausdehnung der Zellen (hier 20 cm * 20 cm) erscheint diese Vereinfachung als annehmbar.

Da es sich um einen ungerichteten Graphen handelt und man insbesondere bei einem holonomen Roboter davon ausgehen kann, dass jede Strecke, die in der realen Welt in einer Richtung befahrbar ist, auch in umgekehrter Richtung befahren werden kann, ist es prinzipiell unbedeutend, welcher Punkt als Startpunkt festgelegt wird. Unter dem Gesichtspunkt, dass man den Pfad *zum Ziel* sucht und unter Berücksichtigung der Tatsache, dass der Algorithmus gleichzeitig mit Entfernung den Pfad zum Ziel ermittelt, ist es jedoch sinnvoll, den Zielknoten als Ausgangsknoten für den Planungsalgorithmus zu wählen.

Die umgekehrte Festlegung von Start und Ziel kann z.B. dann sinnvoll sein, wenn verschiedene alternative Ziele zur Auswahl stehen und die Entfernungen verglichen werden sollen. Da die Pfadlängen zu *allen* Knoten *gleichzeitig* berechnet werden, kann dieses Problem effizient gelöst werden, indem die aktuelle Position als ausgezeichneter Knoten v_0 benutzt wird. Andererseits ist es prinzipiell mit dem selben Algorithmus ohne wesentliche Änderungen (lediglich mit anderer Initialisierung) auch möglich, mehrere alternative Zielknoten zu verarbeiten (d.h. es existiert eine Menge V_0). Es wird dann jeweils die Entfernung und der Pfad zum nächstgele-

genen Ziel berechnet. Allerdings lässt sich nicht ermitteln, um welches der möglichen Ziele es sich dabei handelt, was u.U. Probleme für die jeweilige Anwendung aufwirft. Dieses Vorgehen kann jedoch genutzt werden, wenn mehrere gleichwertige Zielpunkte existieren, von denen ein beliebiger angefahren werden soll, ohne dass weitere Informationen über das gewählte Ziel benötigt werden, wie das z.B. für die Realisierung eines Explorationsverhaltens der Fall sein kann (siehe [Lie00]).

Der eigentliche Algorithmus ist relativ einfach. V sei die Menge der Knoten, E sei die Menge der Kanten, $w(x, y)$ sei die Gewichtsfunktion, die jeder Kante $(x, y), (x, y) \in E$ eine reelle Zahl zuweist. Außerdem verwaltet der Algorithmus eine Liste d der besten (niedrigsten) Schätzung der kürzesten Weglänge zu jedem Knoten sowie eine Liste p , die zu jedem Knoten dessen „Vorgänger“ enthält.

```

1. Initialisierung:
    $\forall v \in V : d(v) = \infty, p(v) = \emptyset$ 
    $d(v_0) = 0, Q = V$ 
2. solange  $Q \neq \emptyset$ :
   suche  $u \in Q$  mit  $d(u)$  minimal
    $\forall v, (u, v) \in E$  :
     falls  $d(u) + w(u, v) < d(v)$ :
        $d(v) = d(u) + w(u, v), p(v) = u$ 
    $Q = Q \setminus \{u\}$ 

```

Abbildung 7.31: Ablauf des Dijkstra-Algorithmus.

Nachdem der Algorithmus beendet ist, enthält d zu jedem Knoten die Länge des kürzesten Pfades zum Zielknoten und p jeweils den nächsten Knoten auf dem Pfad zum Zielknoten. p beschreibt also den vollständigen Pfad von einem beliebigen Knoten zum Zielknoten (ohne dass eine erneute Berechnung notwendig wäre). Interessant dabei ist, dass selbst dann, wenn der berechnete Weg nicht eingehalten wird, trotzdem immer der kürzeste Pfad zum Ziel bekannt ist, da für *alle* Knoten dieser Pfad berechnet wurde.

Die Rechenzeit-Komplexität liegt offensichtlich in $O(n^2)$ mit $n = |V|$: die Schleife in Schritt 2. (Abbildung 7.31) wird n mal durchlaufen, das Schleifeninnere wird in $O(n)$ abgearbeitet. In der Literatur wird darauf hingewiesen, dass mit einer komplizierteren Organisation der Liste Q die Komplexität auf $O(n \cdot \log n + |E|)$ verringert werden kann. ($|E|$ ist bei der vorliegenden Gitterstruktur wiederum in $O(n)$, so dass sich insgesamt $O(n \cdot \log n)$ ergibt).

Eine alternative und deutlich einfachere Möglichkeit der Beschleunigung besteht darin, den Algorithmus vorzeitig abubrechen. Im Grunde genommen genügt es, die Entfernung des eigentlichen Startknotens vom Zielknoten zu ermitteln. Sobald dieser in Schritt 2. (Abbildung 7.31) als Knoten mit minimalem Abstand ausgewählt wird, braucht für alle weiteren Knoten die Berechnung nicht fortgesetzt zu werden. Besonders effizient ist diese Verkürzung, wenn die Strecke zwischen Start und Ziel nur einen kleinen Teil der Karte ausmacht, so dass andernfalls ein großer Overhead an ungenutzten Daten berechnet werden würde.

Wie wirkt sich diese Modifikation auf die oben erwähnte Eigenschaft der Insensitivität gegenüber Pfad-Abweichungen aus? Um diese Frage beantworten zu können, werden folgende Tatsachen ausgenutzt:

- Der Dijkstra-Algorithmus wählt in jedem Schritt als aktuellen Knoten denjenigen Knoten mit minimaler Entfernung. Daher sind zum Zeitpunkt t , an dem der Knoten v_t gewählt wird, bereits alle Knoten v mit $d(v) < d(v_t)$ bearbeitet. Für jeden bearbeiteten Knoten ist der Weg zum Zielknoten bekannt.
- Für jeden Knoten v_p auf dem kürzesten Pfad zwischen dem aktuellen Startknoten v_S und dem Zielknoten v_0 gilt: $d(v_p) < d(v_S)$.

Weiterhin sollen 2 Behauptungen ohne Beweis aufgestellt werden:

- Nahe beieinanderliegende Knoten haben nur gering differierende Entfernungen zum Zielknoten. Daraus folgt, dass geringe Abweichungen vom vorgegebenen Pfad zu geringem Anstieg der tatsächlichen Entfernung führen.
- Die Gesamtwahrscheinlichkeit für Abweichungen vom Pfad bzw. der Erwartungswert der Gesamtabweichung (durch nicht kalkulierte Störungen wie dynamische Hindernisse und Kartenungenauigkeit) steigt mit der längerer Fahrt und damit mit geringerer Zielentfernung. Anders ausgedrückt: In weiter Entfernung zum Ziel ist mit relativ genauer Einhaltung des geplanten Weges zu rechnen, in näherer Umgebung des Zieles mit stärkeren Abweichungen.

Geht man davon aus, dass beide Behauptungen zutreffen, erscheint die Annahme legitim, dass auch nach unerwarteten Korrekturen mit hoher Wahrscheinlichkeit weiterhin ein (ausgehend von der neuen Position) optimaler Weg gefahren wird. Allerdings kann *keinesfalls ausgeschlossen* werden, dass der Roboter den Bereich, in dem die kürzesten Pfade bekannt sind, verlässt. Ein solcher Fall könnte beispielsweise eintreten, wenn in der als Grundlage der Pfadplanung genutzten Karte Wände eingezeichnet sind, die in der realen Umgebung nicht (mehr) existieren. Obwohl jenseits der „Wand“ liegende Zellen als weit entfernt angenommen werden, sind sie vom Roboter bei Ausweichmanövern leicht zu erreichen. Dies kann, je nachdem ob die entsprechende Zelle im Algorithmus bereits einmal mit einer Entfernung versehen wurde oder nicht, dazu führen, dass der fortgesetzte Weg nicht ideal ist oder dass überhaupt kein Weg zum Ziel mehr bekannt ist, was eine Neuberechnung erforderlich macht und somit mit erhöhtem Aufwand verbunden ist.

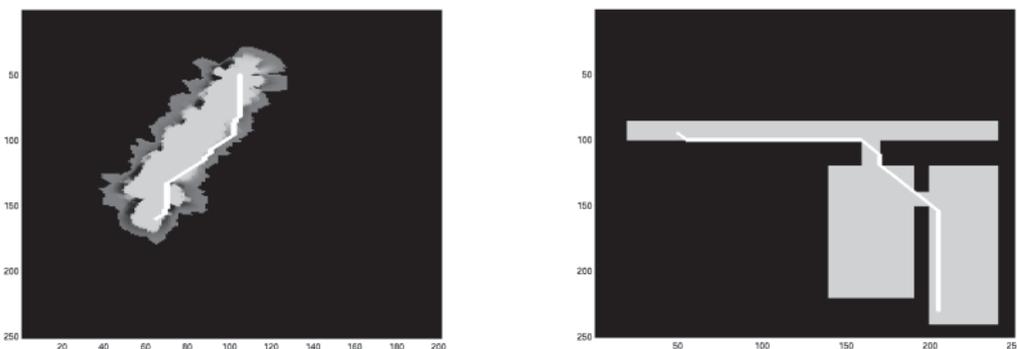


Abbildung 7.32: Pfadplanung in einer explorierten und einer künstlich erzeugten Karte.

Abbildung 7.32 zeigt das Ergebnis der Pfadplanung jeweils in einer künstlich erzeugten und einer tatsächlich vom Roboter aufgenommenen Karte. Ein großes Problem ist deutlich zu erkennen: der Pfad nähert sich in beiden Fällen den Wänden sehr stark an. In der Tat wird der kürzeste Pfad in den meisten Fällen so nah wie möglich an Hindernissen vorbeiführen, da jedes Ausweichmanöver zur Umgehung von Hindernissen einen Umweg darstellt und damit vom idealen Pfad abweicht.

7.4.1.2 Pfadoptimierung

Bei genauerer Überlegung wird klar, dass die dargestellten Pfade eigentlich gar nicht gefahren werden können: Die Positionsangabe bezieht sich jeweils auf den *Mittelpunkt* des Roboters. In der bisherigen Pfadplanung wurde der Roboter als rein punktförmig angenommen. In Wahrheit hat natürlich jedes reale Objekt eine Ausdehnung. Der B21 als Shopping-Assistent überdeckt mit seiner Grundfläche ca. 6 Zellen des Gitters der erlernten Karte, d.h. er kann sich einem Hindernis auf minimal 2 Zellen Entfernung nähern. Um dieses Problem zu lösen, wäre es z.B. möglich, vor der eigentliche Pfadplanung alle Hindernisse um einen Sicherheitsbereich von 2 Zellen auszuweiten, indem ein Erosions-Filter auf die belegten Zellen angewendet wird. Der Effekt wäre, dass der gefundene Pfad immer mindestens 2 Zellen Abstand zu allen Hindernissen hält.

Allerdings führt auch dies nicht unbedingt zu befriedigendem Verhalten. Der gefundene Pfad ist dann immer noch derart optimiert, dass er möglichst eng (unter Berücksichtigung des Sicherheitsabstandes) um Hindernisse und Ecken herumführt. Dieses Ergebnis ist unter zwei Gesichtspunkten nicht wünschenswert: zum Einen muss immer davon ausgegangen werden, dass durch ungenaue Positions- und Abstandsmessungen und durch zeitliche Veränderungen (im einfachsten Fall: vorbeilaufende Menschen) die reale Hinderniskonfiguration von der in der Karte abgebildeten abweicht. Deshalb wird der berechnete Pfad nicht blind abgefahren, sondern es existiert eine weitere Ebene der *lokalen* Navigation, welche unter Berücksichtigung der Richtung zum Ziel die Hindernisvermeidung realisiert. Die lokale Navigation soll Kollisionen vermeiden, indem Ausweichpfade um Hindernisse berechnet werden. Stärkere Annäherungen an Hindernisse führen dazu, dass die Hindernisvermeidung öfter in die Robotersteuerung eingreifen muss und damit nicht nur zu höherem Rechenaufwand, sondern auch zu ineffizienterer (langsamerer und weniger geradliniger) Fahrt. Ein Ziel der Pfadplanung sollte also sein, von vornherein die Hindernisannäherung zu beschränken und damit die lokale Navigation zu erleichtern. Zum Anderen lässt sich dieses Prinzip der Hindernisumgehung auch bei Menschen beobachten: ein Mensch, der einen langen Gang entlanggeht um an dessen Ende zu einer Seite in einen Raum abzubiegen, wird in der Mitte des Ganges entlanggehen, bis er direkt vor der Tür steht. Wenn er um Ecken herumgeht, wird er in den seltensten Fällen versuchen, die Kante in möglichst engem Bogen zu umkurven, sondern einen „vernünftigen“ Abstand einhalten. Im Grunde genommen bildet dieser Weg also nicht wirklich den *kürzesten* Pfad. Im Gegenteil, ein starres Festhalten am kürzesten Pfad wirkt „unnatürlich“.

Das neu formulierte Ziel lautet also: Finde einen möglichst kurzen Pfad vom Start zum Ziel, der gleichzeitig einen möglichst großen Abstand zu Hindernissen hält. „Möglichst großer Abstand“ lässt sich noch etwas genauer spezifizieren: natürlich macht es für die Aufgabe der Navigation eines ca. Menschen-großen Roboters keinen Unterschied, ob ein Hindernis im Abstand 10m oder 15m umfahren wird. Jeder Abstand größer als 2m kann daher prinzipiell als

unendlich groß angenommen werden.

Diese Aufgabe stellt somit eine Mehrzieloptimierung dar. Eine grundlegende Eigenschaft eines solchen Problems ist, dass keine einzelne optimale Lösung existiert. Zur Lösung ist in jedem Fall eine Gewichtung der verschiedenen Ziele notwendig. Diese Gewichtung explizit auszudrücken erscheint jedoch relativ schwierig.

Eine weitere Möglichkeit wäre, eines der beiden Ziele als Nebenbedingung zur Optimierung des anderen zu betrachten. Die Aufgabe lautet dann entweder „Finde den Pfad, der maximalen Abstand zu allen Hindernissen beibehält und dabei nicht länger als n Meter ist“ oder „Finde den kürzesten Pfad unter der Bedingung, dass dieser Pfad mindestens n Meter Abstand zu allen Hindernissen wahr“. Die erste Alternative würde wohl bedeuten, eine Menge von Pfaden zu generieren und diese hinsichtlich ihrer Hindernisannäherung zu vergleichen. Hierbei wäre nicht nur der Rechenaufwand relativ hoch, es ist auch nicht klar, wie eine sinnvolle maximale Weglänge festgelegt werden soll. Der zweite Fall wäre mittels der genannten Hindernis-Erosion und des Dijkstra-Algorithmus effizient berechenbar, allerdings ist auch hier unklar, wie die minimale Hindernisannäherung bestimmt werden kann. Man könnte zwar den Mindestabstand iterativ so lange erhöhen, bis kein Pfad mehr zwischen Start und Ziel existiert bzw. die Pfadlänge signifikant steigt (weil ein völlig anderer Weg gefunden wurde), allerdings wäre dies extrem aufwendig. Außerdem wäre dann die minimale Hindernisannäherung über die gesamte Fahrstrecke starr festgelegt, wohingegen es wünschenswert sein kann, enge Passagen durchqueren zu können und trotzdem in größeren Räumen einen weiteren Abstand zu Hindernissen zu halten.

In der Optimierungs-Theorie sind auch Methoden mit sogenannten Straftermen bekannt, die eine implizite Gewichtung der Ziele darstellen. Auch das nachfolgend vorgeschlagene Verfahren zur Pfadoptimierung gehört zu dieser Kategorie.

Da der eigentliche Dijkstra-Algorithmus gewichtete Graphen verarbeitet, liegt es nahe, die Kantengewichte zu nutzen, um die Hindernisentfernung abzubilden. Dazu wird zunächst für jede Zelle der Abstand zur nächsten belegten Zelle berechnet. Da die Zellen jeweils Knoten entsprechen, diese jedoch selbst keine Gewichte tragen können, wird zu jedem Kantengewicht der Strafterm des dem Ziel näheren Knotens addiert. Dadurch wird die Pfadlänge insgesamt um die Summe der Strafterme aller durchlaufenen Knoten erhöht. Es sei $d(v)$ die Distanz eines Knotens(bzw. der zugehörigen Zelle) zur nächsten belegten Zelle (als Anzahl von Zellen \rightarrow benachbarte Zellen haben Abstand 1). Der jeweilige Strafterm berechnet sich nach Gleichung 7.31:

$$penalty(v) = \begin{cases} \frac{100}{d(v)^2} & \text{falls } d(v)^2 \leq 100 \\ 0 & \text{sonst} \end{cases} \quad (7.31)$$

Der Strafterm kann also einen Maximalwert von 100 erreichen, wenn die betrachtete Zelle einer belegten Zelle direkt benachbart ist, d.h. wenn der Weg durch diese Zelle als genau so lang bewertet wie durch 100 Zellen ohne Hindernisse in der Umgebung.

Die Benutzung dieser Strafterme führt dazu, dass Pfade, die sehr nah an Hindernissen und Wänden entlangführen, als relativ lang bewertet werden und dadurch andere Wege, die sonst länger erscheinen würden, bevorzugt werden. Allerdings wird nicht verhindert, dass auch diese langen Wege erkundet werden. Dies ist wichtig, um trotzdem die Möglichkeit zu erhalten, enge Durchfahrten oder schmale Gänge zu passieren.

Die explizite Berechnung der Hindernisentfernung für jede Zelle löst auch das bereits angesprochene Problem der Roboter-Ausdehnung: Zellen, deren Hindernisabstand kleiner als der Roboter-Radius ist, werden nun ebenfalls als belegt betrachtet und stehen damit für die Pfadplanung nicht zur Verfügung.

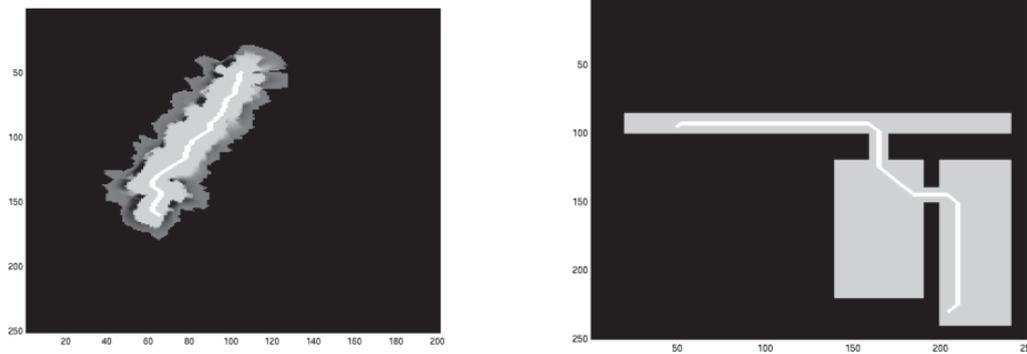


Abbildung 7.33: Pfadplanung unter Berücksichtigung des Hindernisabstandes.

Die in Abbildung 7.33 gezeigten Pfade veranschaulichen den Erfolg dieser Methode (vgl. Abb 7.32). Der vorgegebene Weg weicht Hindernissen in einem Bogen aus und versucht, Abstand von Wänden zu halten, was dazu führt, dass der Pfad in schmalen Gängen in der Mitte des Ganges entlangführt. Es ist deutlich zu erkennen, dass der abgewandelte Algorithmus nicht den absolut kürzesten, sondern einen „natürlich“ wirkenden Pfad ermittelt.

Abschließend soll betrachtet werden, welche Auswirkungen zu erwarten sind, wenn mehrere Wege zum Ziel möglich erscheinen (Abbildung 7.34). Dazu ist hier eine künstlich erzeugte Karte abgebildet, welche einen schmalen und einen breiten Durchgang enthält. Der Roboter muss einen der beiden Wege wählen, wobei der Weg durch den schmalen Gang deutlich kürzer ist.

Wie zu erwarten, wird der kürzere Weg gewählt, solange der entsprechende Durchgang relativ breit ist. Die Verringerung der Durchgangsbreite ändert nichts am gewählten Weg, bis eine gewisse Grenze unterschritten wird. Bei einer bestimmten Breite erscheint dieser Weg wegen der starken Annäherung an die Seitenwände jedoch nicht mehr als lohnend, obwohl er im Prinzip immer noch genügend Platz für eine unbehinderte Durchfahrt bietet. Der Algorithmus wählt nun eine „Umweg“, um nicht den schmalen Gang durchfahren zu müssen.

Dieses Ergebnis entspricht durchaus dem erwarteten und auch gewünschten Verhalten. Es erscheint sinnvoll, alternative Wege zu betrachten, wenn der kürzeste Pfad mit einer Durchfahrt durch eine enge Passage verbunden ist und andere Wege existieren. Da keine minimale Hindernisannäherung (außer dem Roboter-Radius) fest vorgegeben ist, erfolgt in jedem Fall eine Abwägung zwischen dem zu fahrenden Umweg und dem Nachteil des geringeren Hindernisabstandes. Eine feine Ausbalancierung dieser beiden Ziele kann über die Einstellung der Höhe des Straftermes vorgenommen werden. In den bisher vorgenommenen Versuchen hat sich Gleichung 7.31 für den Strafterm als gute Lösung erwiesen.

Eine weitere Motivation für das gezeigte Verhalten ergibt sich aus folgender Überlegung: Je enger und je länger eine Passage ist, umso größer ist die Wahrscheinlichkeit, dass dieser Weg unerwartet durch ein mobiles Hindernis (Menschen, Kisten, etc.) blockiert und für den Roboter

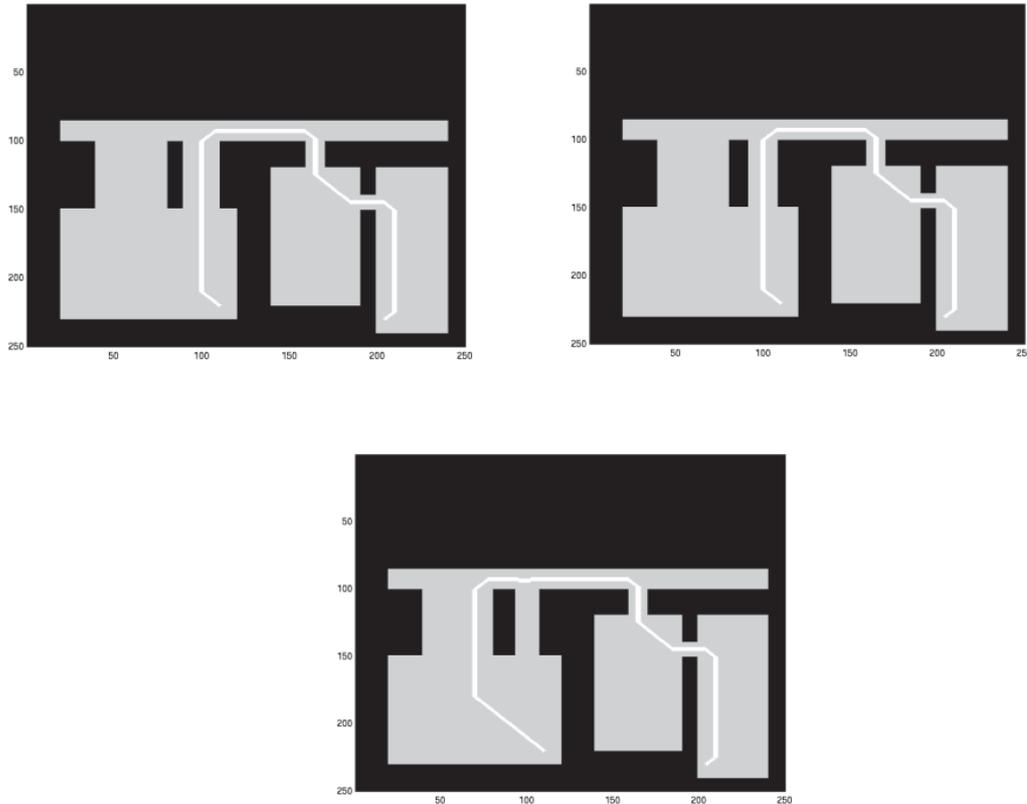


Abbildung 7.34: Unterschreitet die Wegbreite eine gewisse Grenze, so wird ein längerer Weg gewählt.

völlig unpassierbar ist. Das Meiden solcher Fahrwege bedeutet also eine Verringerung des Risikos, in einer temporären Sackgasse zu enden und umdrehen zu müssen, um einen anderen Pfad zu suchen.

7.4.2 Bewegungssteuerung

Obwohl durch den Algorithmus zur Pfadplanung ein Weg berechnet wird, der bekannte, d.h. in der gelernten Umgebungskarte repräsentierte Hindernisse umgeht, muss trotzdem stets mit unerwartet auftauchenden Störungen gerechnet werden: Einerseits können Karten- und Positionsinformationen des Roboters niemals als exakt angenommen werden. Unberechenbare bewegliche Objekte wie Menschen oder Fahrzeuge können natürlich in keinem Umweltmodell abgebildet werden. Daher ist es selbst bei idealer Kenntnis der Umgebung und perfekter Orientierung nötig, einen Algorithmus zu entwickeln, der den Roboter in die Lage versetzt, während der Fahrt zu seinem Ziel eventuellen Hindernissen auszuweichen.

Die Strategie zur Hindernisvermeidung geht davon aus, dass eine gewünschte Richtung zu einem anzusteuernenden globalen Ziel vorgegeben wird. Diese kann sowohl aus der Pfadplanung resultieren, aber ebenso auch in der perspektivischen Anwendungen durch MRI-Module gene-

riert werden, indem der Roboter z.B. auf einen potentiellen Nutzer zufährt. Ziel ist allerdings zunächst nicht, beweglichen Objekten, die sich auf Kollisionskurs dem *stehenden* Roboter nähern, auszuweichen, obwohl auch dies mit minimalen Erweiterungen möglich ist. Die Funktion der Bewegungssteuerung besteht somit derzeit darin, die Vorgabe einer gewünschten Fahrtrichtung unter Berücksichtigung der momentan wahrgenommenen Hindernisse in entsprechende Steuerkommandos umzusetzen.

Bei einem Gewicht von 160 kg und einer (zumindest anzustrebenden) Fahrgeschwindigkeit von etwa 1m/s können bereits leichte Berührungen während der Fahrt zu Schäden an Umgebungsobjekten oder dem Roboter selbst bzw. dessen teilweise recht empfindlichen Aufbauten führen. Die Sicherheit, mit der Kollisionen vermieden werden, ist daher ein entscheidendes Kriterium.

Eine naive Strategie zur Bewegungssteuerung könnte zunächst so aussehen, dass der Roboter bei einer detektierten Annäherung an ein Hindernis ab einem minimalen Abstand stoppt und eine Ausweichrichtung bestimmt. Der Roboter würde sich dann in diese Ausweichrichtung drehen und seine Fahrt wieder aufnehmen. Dieses Strategie würde jedoch zu einem relativ unruhigen Fahrverhalten führen, wenn bei Kurskorrekturen jeweils angehalten wird. Außerdem steigt natürlich die Zeit, die benötigt wird, um ein bestimmtes Ziel zu erreichen. Es wird daher nach Methoden gesucht, die eine sichere Kollisionsvermeidung gewähren, dabei aber keine Stopps erzwingen und in einer möglichst gleichmäßigen Bewegung resultieren.

7.4.2.1 Potentialfeld-Verfahren

Ein verbreiteter Ansatz zur Bewegungssteuerung ist die sogenannte Virtuelle Potentialfeldmethode (vgl. auch [BK91]). Sie wird u.a. auch zur Berechnung von optimalen Pfaden benutzt, soll hier aber nur als Strategie zur Hindernisvermeidung betrachtet werden. Bei dieser Methode wird angenommen, dass jedes wahrgenommene Objekt der Umgebung eine abstoßende Kraft auf den Roboter ausübt. Die wirkende Kraft steigt mit sinkendem Abstand exponentiell an. Für praktische Belange kann die Kraft jenseits einer gewissen Distanz (bspw. 1m) als verschwindend gering und damit zu Null angenommen werden.

Praktisch wird der Roboter jedoch keine Objekte identifizieren (siehe auch 2.2.4), und es wäre auch nicht offensichtlich, wie die Form und Größe der Objekte sich auf die ausgeübte Kraft auswirken sollte. Statt dessen wird die Messung jedes Abstandssensors als eigenständige Wahrnehmung aufgefasst, so dass aus jeder einzelnen gemessenen Entfernung eine Kraft berechnet wird (Abbildung 7.35).

Durch die Annahme solcher Kräfte entsteht um den Roboter herum ein virtuelles Potentialfeld. Um maximalen Abstand zu den Hindernissen einzuhalten, muss lediglich das (lokale) Minimum dieses Potentialfeldes gesucht werden. Dies wird erreicht, indem man der momentan wirkenden Gesamtkraft folgt, welche sich als die Summe aller auftretenden Kräfte berechnet. Der Roboter führt also einen Gradientenabstieg in einem Potentialgebirge aus.

Auf diese Weise kann der Roboter die Position mit der geringsten Hindernisannäherung auch bei dynamisch veränderlicher Hinderniskonfiguration finden. Dabei ist jedoch noch nicht berücksichtigt, dass ein festgelegtes Ziel angefahren werden soll, dessen Richtung durch die Pfadplanung vorgegeben wird. Um sich im Potentialfeld in (ungefähr) dieser Richtung zu bewegen, wird eine vom Ziel ausgehende, den Roboter anziehende Kraft eingeführt (Abbildung 7.36). Durch Summation aller auftretenden Kräfte erhält man die Gesamtkraft, deren Richtung

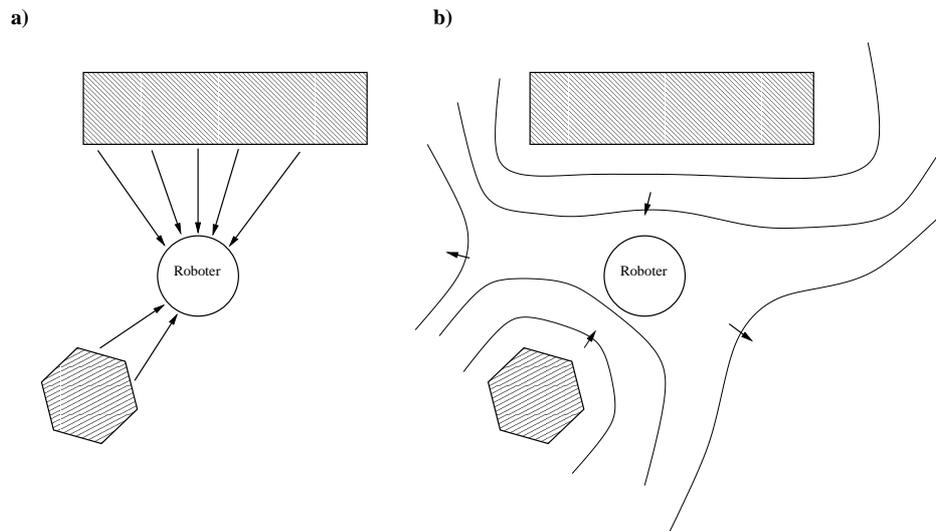


Abbildung 7.35: a) Die Hindernisse üben virtuelle Kräfte auf den Roboter aus. b) Aus den Kräften resultiert ein virtuelles Potentialfeld. Pfeile zeigen jeweils die Richtung geringerer Hindernisannäherung.

die Richtung der aktuellen Bewegung darstellt. Abbildung 7.37 zeigt die zu erwartende idealisierte Fahrtroute zwischen den dargestellten Hindernissen. Um die Fahrdynamik des Roboters zu beachten, der wegen seiner hohen Masse relativ schwerfällig auf abrupte Geschwindigkeitsänderungen reagiert, ist es sinnvoll, die Fahrgeschwindigkeit vom Abstand des nächsten Hindernisses abhängig zu machen. Aus dem selben Grund ist es notwendig, die Fahrt zu glätten, indem die Fahrtrichtungsangaben mit einem Tiefpass gefiltert werden. Dadurch wird zwar eine langsamere Reaktion auf plötzlich auftauchende Objekte in Kauf genommen, dafür werden aber zu starke Richtungsänderungen unterdrückt.

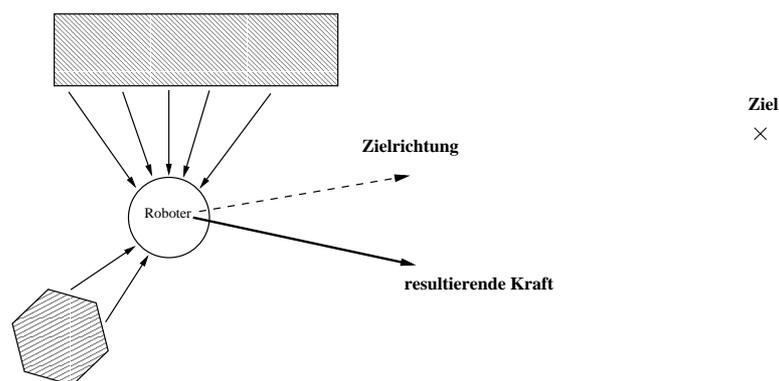


Abbildung 7.36: Die Gesamtkraft ergibt sich als Summe aller abstoßenden und anziehenden Kräfte.

Die Performanz kann durch geringe Modifikationen erhöht werden. So ist es sinnvoll, Hindernisse, die in Fahrtrichtung gesehen *hinter* dem Roboter liegen, nicht in die Berechnung der

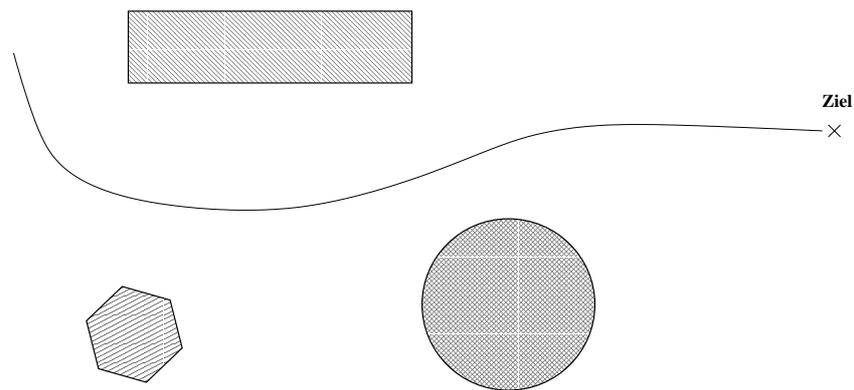


Abbildung 7.37: Der resultierende Kurs zwischen den dargestellten Objekten.

Gesamtkraft einzubeziehen, da die Bewegung von diesen wegführt. Auch die vorderen Hindernisse können nach ihrer Relevanz gewichtet werden, indem ein Faktor benutzt wird, der berücksichtigt, wie weit diese Objekte von der augenblicklichen Fahrtrichtung abweichen.

$$F' = F * c \quad (7.32)$$

$$c = \begin{cases} \cos(\alpha) & \text{falls } |\alpha| \leq 90^\circ \\ 0 & \text{sonst} \end{cases} \quad (7.33)$$

$$\text{mit } -180^\circ \leq \alpha \leq 180^\circ$$

Dadurch lässt sich eine höhere Fahrgeschwindigkeit, z.B. bei Bewegungen parallel zu einer nahen Wand erreichen. Auch verfolgende Benutzer werden nun nicht mehr als Hindernisse betrachtet.

Grundlegende Schwierigkeiten bleiben jedoch bestehen. Wie in Abbildung 7.38 zu sehen, bewirkt ein langgestrecktes Objekt (wie z.B. auch eine Wand) dadurch, dass es wegen seiner Länge durch mehr Sensoren wahrgenommen wird, eine wesentlich größere Kraft als ein kleines Objekt. Bei der Durchfahrt zwischen diesen beiden Hindernissen führt das dazu, dass der Roboter diese Hindernisse nicht etwa mittig passiert, so dass nach beiden Seiten genügend Platz bleibt, sondern stark auf das kleinere Objekt zu getrieben wird.

Ein weiteres wesentliches Problem des Potentialfeld-Ansatzes besteht in der notwendigen Parametrisierung, wobei sich insbesondere der Festlegung der anziehenden Kraft in der globalen Bewegungsrichtung als kritisch erweist. Einerseits muss diese Kraft groß genug sein, um auch bei seitlich vorn auftauchenden Hindernissen eine Bewegung auf das Ziel zu aufrecht zu erhalten, andererseits muss die abstoßende Kraft eines Hindernisses ausreichen, um diese ziehende Kraft auszugleichen und damit einen Mindestabstand zum Hindernis zu wahren. Dies wird durch die oben demonstrierte Abhängigkeit der entstehenden Kräfte von der Objektgröße zusätzlich erschwert. Besonders problematisch kann die Durchfahrt durch Türen oder ähnliche enge Passagen werden. Diese sind meist recht schmal, so dass der Roboter von den links und rechts angrenzenden Wänden starke Abstoßungen erfährt (Abbildung 7.39). Für einen solchen Fall wird also eine sehr hohe anziehende Kraft benötigt, welche in anderen Situationen wiederum riskant sein kann. Das Grundproblem ist hier, dass durch die reine Summierung

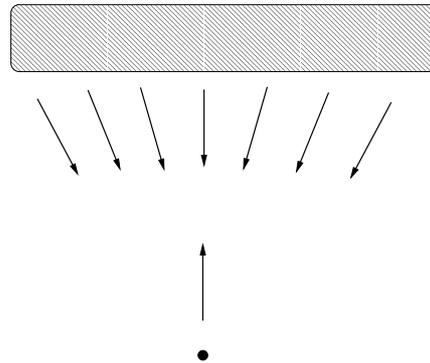


Abbildung 7.38: Die Größe des Objektes hat einen sehr starken Einfluss auf die ausgeübte Kraft.

der Kräfte die Information darüber verloren geht, wo die entsprechenden Objekte sich befinden. Daher ist der Roboter nicht mehr in der Lage, den zwischen den Wänden existierenden Freiraum, den er bedenkenlos befahren könnte, zu erkennen. Die Information ist reduziert auf „Hindernis vorn“.

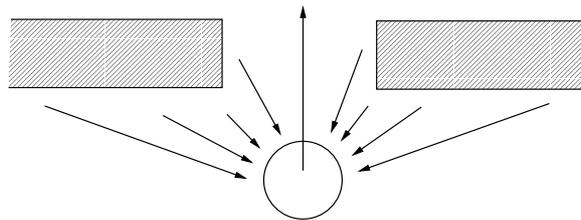


Abbildung 7.39: Schmale Durchfahrten erweisen sich als äußerst schwierig, da der eigentlich zu durchfahrende Freiraum aufgrund der wirkenden abstoßenden Kräfte vom Roboter nicht mehr wahrgenommen werden kann.

Experimente im BauMarkt zeigten auch, dass es in schmalen Gängen zu einem pendelnden Fahrverhalten kommen kann (Abbildung 7.40), wenn der Roboter auf eine Wand zu fährt, von dieser stark abgestoßen wird und sich dadurch der anderen Wand nähert, von dieser wiederum abgestoßen wird etc. Bei geradliniger Fahrt in Gangmitte heben sich im Idealfall die Kräfte von beiden Seiten gegenseitig auf, so dass eine geradlinige Fahrt erzielt wird. Selbst bei geringen Abweichungen von dieser Ideallinie ist jedoch das erneute Erreichen des Balance-Zustandes kaum wieder zu gewährleisten.

Alle diese Probleme sind in der Methode selbst begründet und treten auch dann auf, wenn eine exakte Repräsentation der Umwelt vorliegt. Sie werden jedoch noch verschärft durch die fehlerbehafteten Sensor(Sonar)-Messungen. Für alle bisherigen experimentellen Untersuchungen wurden als Abstandssensoren lediglich die Sonar-Sensoren verwendet. Die Tatsache, dass diese relativ ungenau arbeiten, muss bei der Parametrisierung zusätzlich berücksichtigt werden. Zum Beispiel ist davon auszugehen, dass durch Messungenauigkeiten Hindernisse zumindest zeitweilig „übersehen“ werden können. Abweichungen des gemessenen Abstandes von der tatsächlichen Entfernung sind kritisch für das angestrebte, möglichst kontinuierliche und glatte

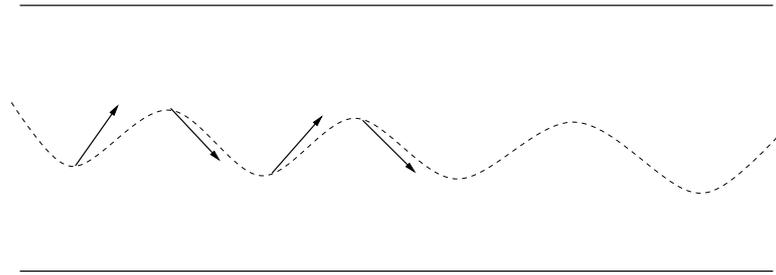


Abbildung 7.40: In engen Korridoren können oszillierende Änderungen der Fahrtrichtung auftreten, die sich in einem pendelnden und damit wenig natürlich und zielgerichtet wirkenden Fahrverhalten äußern.

Fahrverhalten: bei zu lang gemessenem Abstand wird das Objekt als zu weit entfernt eingeschätzt, was im Extremfall zur Kollision führen kann. Bei zu kurz gemessenem Abstand erscheinen „Phantom“-Hindernisse. Unnötige Ausweichversuche führen dann zu sehr unruhiger und langsamer Fahrt. Es ist daher nötig, möglichst genaue Informationen über die tatsächlichen Abstände zu extrahieren.

7.4.2.2 Vektorfeld-Histogramm

Resultierend aus den Problemen des Potentialfeld-Ansatzes wird in [BK91] ein alternatives Verfahren, das sogenannte Vektorfeld-Histogramm, vorgeschlagen. Man benutzt dabei eine dreistufige Datenrepräsentation: Auf der ersten Stufe wird eine detaillierte Repräsentation der Umgebung des Roboters aufgebaut. Diese arbeitet mit einem Gittermodell und entspricht im Prinzip einer vereinfachten Version der in 7.2 beschriebenen Belegtheitsgitterkarte. Als Resultat enthält jede Zelle eine Belegungswahrscheinlichkeit als diskrete Anzahl der für diese spezielle Zelle aufgenommenen Hindernisbeobachtungen (in der Karte bilden sich also maximale Belegtheitswahrscheinlichkeiten an den Positionen der tatsächlichen Hindernisse heraus). Aus diesem Umgebungsmodell wird die zweite Stufe der Umgebungsrepräsentation berechnet, indem der gesamte Umkreis des Roboters in Sektoren unterteilt und für jeden dieser Sektoren wiederum eine von außen einwirkende Kraft berechnet wird. Jede einzelne Zelle übt dabei eine Kraft aus, die proportional zum Quadrat ihres Belegtheitswertes und zu ihrer (negierten) Entfernung ist. Auf diese Weise haben sicher belegte Zellen einen hohen, unsichere Zellen einen sehr niedrigen Einfluss. Nahe liegende Zellen üben größere Kräfte aus als weiter entfernte. Die resultierende Kraft eines Sektors ergibt sich als Summe der Kräfte aller Zellen, die in diesem Sektor liegen.

Auf der zweiten Stufe der Umgebungsrepräsentation existiert nun ein Polar-Histogramm als diskretisierte Darstellung der Hinderniswahrscheinlichkeit und -entfernung für jede Blickrichtung (Abbildung 7.41). Aus dieser Darstellung soll nun unter Berücksichtigung einer Richtung zum Ziel die zu steuernde Fahrtrichtung ermittelt werden.

Das Histogramm besteht normalerweise aus „Bergen“ und „Tälern“, wobei die Berge hohe Hinderniswahrscheinlichkeiten darstellen, die Täler dagegen mit hoher Wahrscheinlichkeit Freiraum repräsentieren. Für die Weiterverarbeitung werden die Sektoren in die beiden Klassen „frei“ und „belegt“ eingeteilt (Abbildung 7.42). Dazu wird eine Schwelle Θ_0 festgelegt, Sektoren mit $\Theta > \Theta_0$ gelten als belegt, andere entsprechend als frei. Die Höhe der Schwelle ist

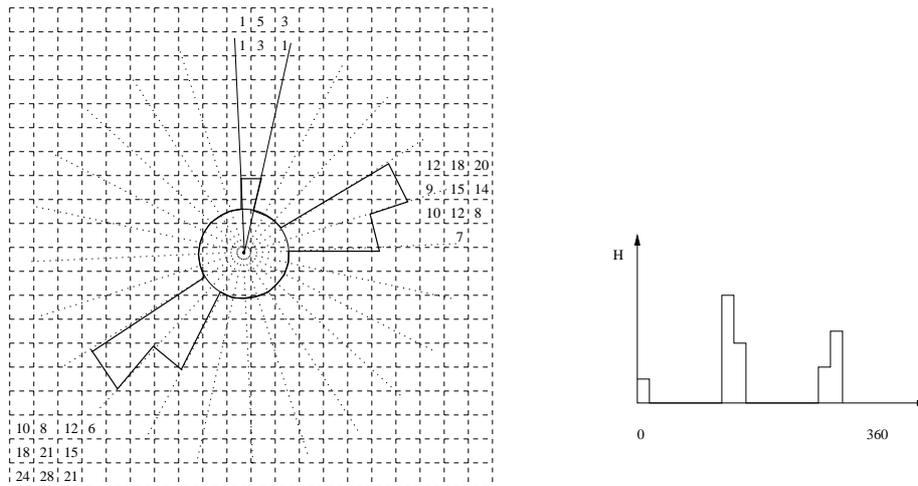


Abbildung 7.41: Aus der Belegungskarte als erster Repräsentationsebene wird ein Polar-Histogramm ermittelt.

abhängig von der zugelassenen Annäherung an Hindernisse. Eine genauere Analyse der Sensitivität auf die Höhe der Schwelle folgt im Anschluss an die komplette Beschreibung. Eine Anzahl zusammenhängender freier Sektoren wird als Tal bezeichnet. Die Breite eines Tales beschreibt den Abstand zwischen den entsprechenden begrenzenden Hindernissen (bzw. deren Projektion auf den Betrachter).

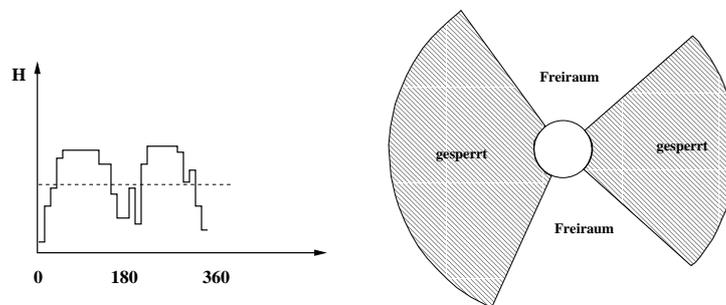


Abbildung 7.42: Durch eine Schwellenoperation werden freie und belegte Sektoren ermittelt.

Nachdem nun festgelegt ist, welche Richtungen befahrbar sind, kann im nächsten Schritt die exakte Fahrtrichtung berechnet werden. Normalerweise existiert eine Auswahl mehrerer Täler. Die Klassifizierung wird allerdings mit Hilfe mehrerer aufeinander folgender Scans vorgenommen, also mehrerer signifikant unterschiedlicher Richtungen, die Wegen zwischen den Hindernissen hindurch entsprechen. Es wird zunächst dasjenige Tal ausgewählt, welches der gewünschten Fahrtrichtung (der Richtung zum Ziel) am nächsten liegt. Falls der direkte Weg zum Ziel frei ist, wird die Zielrichtung in einem Tal des Histogramms liegen, andernfalls wird ein in eine andere Richtung weisendes Tal gewählt, was ein Ausweichen um das detektierte Hindernis bewirkt.

Für Auswahl des Tales wird zwischen engen und breiten Tälern unterschieden: bei einer

Winkelbreite von mehr als 90° gilt das Tal als breit, sonst als eng. Breite Täler entstehen dabei durch weit verstreute Hindernisse, oder aus Situationen, in denen lediglich ein Hindernis (z.B. eine Wand) in der Nähe des Roboters liegt. Ein enges Tal bedeutet nahe beieinander liegende Hindernisse. Bei einem breiten Tal wird der der Zielrichtung nähere Rand gesucht und von diesem aus ein Winkel von typischerweise 45° angesteuert (Abbildung 7.43). Dies entspricht genau der halben Breite eines maximal großen engen Tals. Bei starker Annäherung an das Hindernis führt dies zu einer Fahrtrichtung, die die Entfernung vergrößert, bei hohem Abstand dagegen kann sich der Roboter weiter annähern. Bei einem bestimmten Abstand d_B wird der Roboter genau parallel zum Hindernis fahren, wobei kleine Abweichungen wie oben beschrieben korrigiert werden. Die Länge dieses Balance-Abstandes d_B hängt dabei hauptsächlich vom Winkel ab, der zwischen Talrand und resultierender Fahrtrichtung gewählt wird (die beschriebenen 45°). Durch Änderung dieses Winkels kann der Hindernisabstand bei Parallelfahrt variiert werden, größere Winkel resultieren auch in größerer Entfernung d_B .

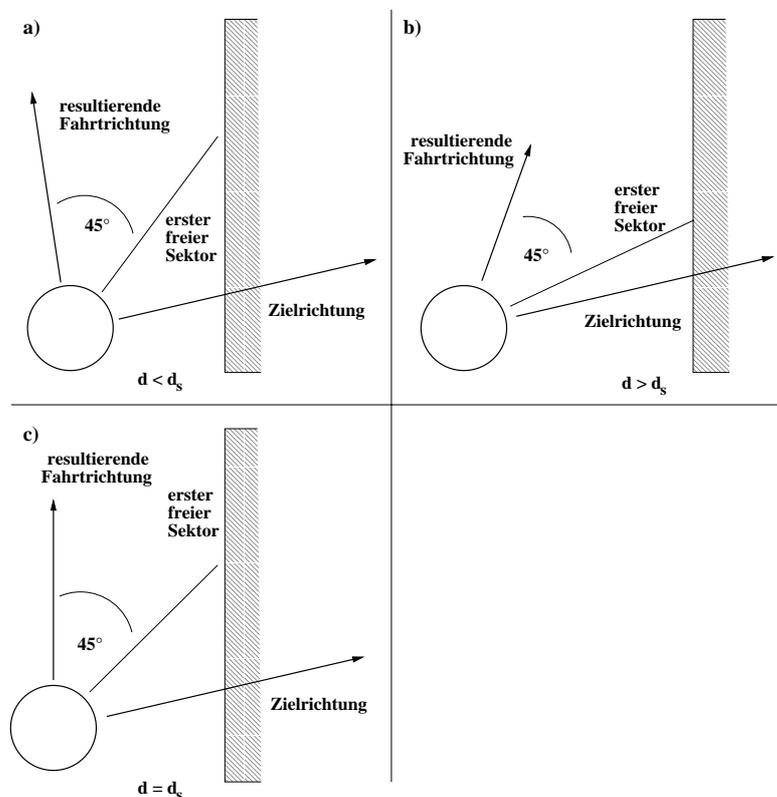


Abbildung 7.43: Bei Abstand d_s fährt der Roboter stabil parallel zur Wand.

Wenn das dem Ziel nächstliegende Tal ein enges Tal ist, wird dagegen die Mitte des Tals angesteuert (Abbildung 7.44). Dies sichert die Einhaltung des maximalen Abstandes zu allen Hindernissen.

Ein Vorteil dieses Verfahrens liegt u.a. darin, dass lediglich ein einziger Parameter wirklich einzustellen ist: die Schwelle, welche bestimmt, ob ein Sektor als frei oder belegt betrachtet wird. Allerdings ist die Einstellung dieser Schwelle relativ unkritisch. Eine Veränderung der Schwelle ändert lediglich die Breite der wahrgenommenen Täler. Geringe Änderungen wirken

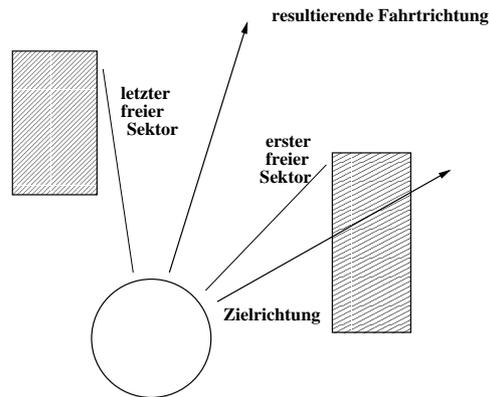


Abbildung 7.44: Zwischen nahen Objekten wird maximaler Abstand nach beiden Seiten gewahrt.

sich daher im Wesentlichen auf den Abstand aus, in dem der Roboter an Hindernissen entlangfährt. Falls die Schwelle deutlich zu niedrig ist, wird der Roboter auf Hindernisse zufahren. Bei stärkerer Annäherung steigt allerdings die abstoßende Kraft, so dass die Histogrammbelegung des entsprechenden Sektors die Schwelle übersteigt und der Roboter auf einen Ausweichkurs einschwenken sollte. Bei hohen Geschwindigkeiten kann es passieren, dass es nicht mehr rechtzeitig gelingt, auszuweichen. Eine deutlich zu hohe Schwelle kann bedeuten, dass Täler vollständig geschlossen werden und dadurch kein Weg mehr zwischen 2 nahe beieinander liegenden Objekten gefunden wird, so dass ein großer Umweg gefahren werden muss.

Der große Vorteil dieser Methode gegenüber der Verwendung von Potentialfeldern ist die genauere Information, die im Polar-Histogramm aufgehoben und für die Ermittlung der Fahrtrichtung benutzt wird. Während die Potentialfeldmethode alle Sensorwerte in Kräfte abbildet und diese direkt aufsummiert, bleibt hier das Wissen über freie und gesperrte Richtungen erhalten. Erst dadurch ist es möglich, relativ einfach und ohne aufwendige und extrem kritische Parametrisierung auch schmale Durchgänge zu identifizieren und zu durchfahren.

Wesentliches Merkmal der Vektorfeld-Histogramm-Methode ist die Verwendung einer Umgebungskarte als oberste Repräsentationsebene. Diese wird während der Fahrt aufgebaut und ständig aktualisiert, es handelt sich also um eine globale Karte. Einerseits trägt diese wesentlich dazu bei, Sensorungenauigkeiten auszugleichen und dadurch stabilere Werte für die Belegungswahrscheinlichkeiten in jeder Richtung zu erhalten, andererseits bedeutet dieses Vorgehen auch einen erheblichen Mehraufwand. Das Verfahren wurde von den Autoren [BK91] nur zur lokalen Navigation auf relativ kurzen Strecken getestet. Bei längeren Navigationsfahrten stellt sich auch hier die Frage nach der Konsistenz der globalen Karte. Im Prinzip würde sich eine Kopie des internen Weltmodells entwickeln, was keineswegs als sinnvoll erscheint. Daher wurde hier nach einer Möglichkeit gesucht, das Vektorfeld-Histogramm auch ohne diese Karte zu benutzen. Es werden nun nicht mehr die sich ergebenden Kräfte der einzelnen Zellen für jeden Sektor aufsummiert, sondern ein Sektor enthält lediglich die Information über die in dieser Richtung gemessene Entfernung. Bei der ursprünglichen Methode war die Sektoraufteilung nicht explizit festgelegt, eine feinere Aufteilung führte zu feiner abgestuften Ausweichmanövern um Hindernisse. Im konkret vorliegenden Fall ist nun eine Aufteilung entsprechend der Sensoranzahl bzw. Sensoranordnung ratsam. Anstelle der Schwelle Θ_0 wird demzufolge lediglich ein Min-

destabstand d_S definiert. Ein Vorteil ist natürlich die intuitivere Festlegung dieser Schwelle. Um trotz der fehlenden Glättungsfunktion durch die Sensor-Integrations-Ebene eine gewisse Stabilität in den Sensormessungen zu erzielen, wird folgendes Messprinzip angewandt: Es werden jeweils 5 Messungen des Abstandes an jedem Sensor vorgenommen, falls mindestens 3 davon unterhalb der Entfernungsschwelle liegen, wird dieser Sektor als belegt bezeichnet, ansonsten als frei. In Abbildung 7.46, und 7.47 ist zu beachten, dass jeweils nur einer dieser Scans dargestellt ist. Für die tatsächliche Bewertung als freie oder versperrte Richtung (jeweils rechts) stehen also mehr Informationen zur Verfügung als abgebildet.

Abbildung 7.45 veranschaulicht, dass d_S nicht automatisch identisch mit dem oben erläuterten Parallelfahrt-Abstand d_B ist.

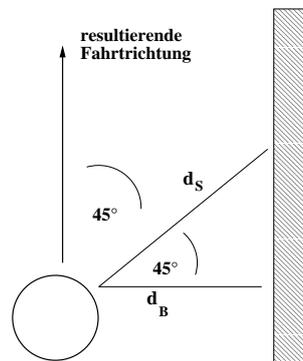


Abbildung 7.45: Parallelfahrt bei explizit festgelegter Distanz-Schwelle.

Es gilt (unter Vernachlässigung der Roboter-Ausdehnung):

$$2 \cdot d_B^2 = d_S^2 \quad (7.34)$$

$$d_B = \sqrt{\frac{1}{2}} \cdot d_S \quad (7.35)$$

Diese Beziehung sollte beachtet werden, wenn ein bestimmter einzuhaltender Mindestabstand zu Hindernissen gewünscht wird. In der Implementation des Algorithmus wird hier $d_S = 1m$ benutzt. Damit wird eine sehr ruhige und geradlinige Fahrt in der Mitte eines ca. 2m breiten Ganges erzielt.

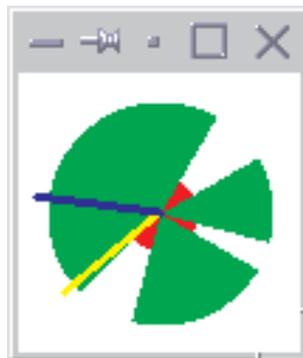
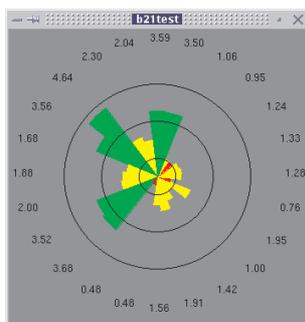


Abbildung 7.46: Sonar-Scan (links) und resultierende Einteilung in freie und belegte Sektoren (rechts). Der hell hervorgehobene Strahl zeigt die Richtung zum globalen Ziel, dunkel ist die resultierende Fahrtrichtung dargestellt.

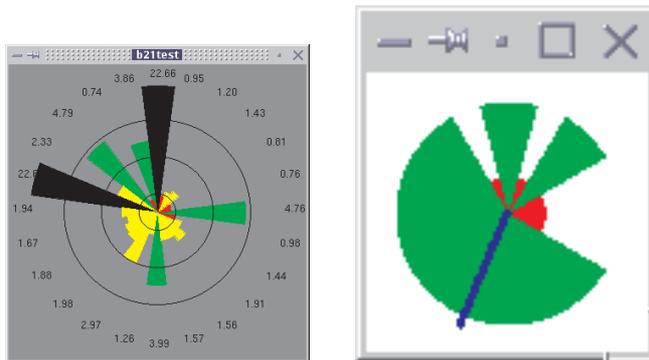


Abbildung 7.47: In diesem Beispiel ist genug Freiraum in Zielrichtung vorhanden, die tatsächliche Fahrtrichtung entspricht daher der Richtung zum globalen Ziel.

7.5 Zusammenfassung

Zum Abschluss dieses Kapitels soll der Gesamtkomplex Navigation in der BauMarkt-Umgebung anhand des aktuellen Implementationsstandes resümiert und auf einige Schwachpunkte sowie unmittelbar anstehende offene Probleme hingewiesen werden. Abbildung 7.48 fasst die wesentlichen Komponenten des vorgestellten Systems zur Roboternavigation in der BauMarkt-Umgebung zusammen.

Basisrepräsentation

Die Modellierung der Umgebung des Roboters durch Belegtheitsgitterkarten hat sich als sinnvoll und weitgehend praktikabel erwiesen. Der ursprüngliche Nachteil dieser Art der Umgebungsrepräsentation, der im hohen Speicherplatzbedarf bei der Kartierung weiträumiger Umgebungen bei geringer Zellengröße bestand, tritt mit dem heutigen Stand der Rechentechnik mehr und mehr in den Hintergrund. Gleiches gilt für die effiziente Pfadplanung auf einer solchen Repräsentation, die für zwei diametral gelegene Positionen des BauMarktes und einer Zellengröße von 20×20 cm nur wenige Sekunden in Anspruch nimmt. Vorteilhaft ist weiterhin die Bewahrung der metrischen Informationen der Umgebung, was die Attributierung der globalen Umgebungskarte (siehe dazu auch Abschnitt 9.3) mit Artikelstandorten und Marktberichen deutlich vereinfacht.

Lokale Umgebungskarte und Hinderniswahrnehmung

Ultraschall-basiertes Verfahren Zunächst wurde ein aus der Literatur bekanntes Verfahren implementiert, dass die Messwerte der Sonarsensoren mit Hilfe eines neuronalen Netzwerks auf Belegheitswahrscheinlichkeiten der Zellen in der egozentrisch repräsentierten lokalen Umgebung des Roboters abbildet. Allerdings erscheinen in der so aufgebauten lokalen Karte nur die Hindernisse, die mit der vorliegenden Anordnung der Sonarsensoren auch tatsächlich wahrgenommen werden können. Bei alleiniger Verwendung der Sonarsensoren würden daraus starke Einschränkungen für die Umgebung resultieren, die gerade im BauMarkt mit der hohen Vielfalt an möglichen Hindernisgeometrien unmöglich eingehalten werden können. Dies führte zur Entwicklung eines visuell-basierten Verfahrens zur Hinderniswahrnehmung, mit dessen Hilfe der Aufbau einer lokalen Umgebungsrepräsentation, die die tatsächlich vorliegende dreidimensionale Hindernissituation besser beschreibt, angestrebt wird.

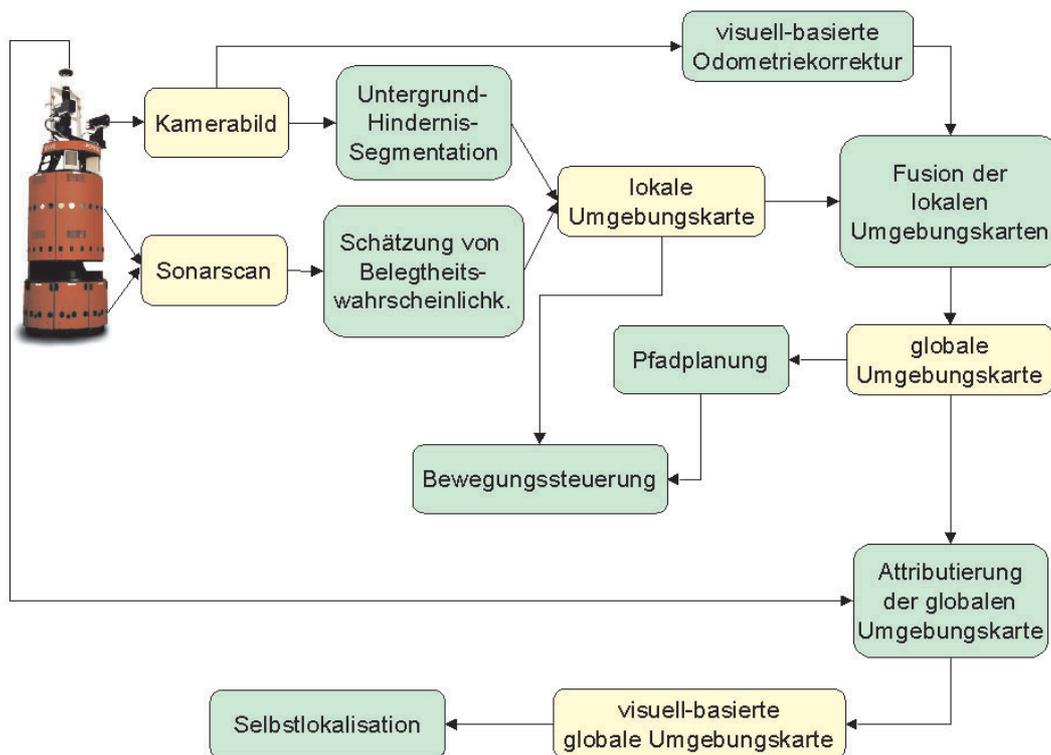


Abbildung 7.48: Überblick über die wesentlichen Komponenten des vorgestellten Systems zur Roboternavigation. Repräsentationen sind gelb, informationsverarbeitende Module grün dargestellt.

Visuell-basiertes Verfahren Das visuelle Verfahren zur Hinderniswahrnehmung beruht auf dem Prinzip der invers-perspektivischen Transformation. Es nutzt die Tatsache aus, dass sich der Roboter ausschließlich auf ebenem Terrain bewegt, und verwendet eine Kombination aus verschiedenen Merkmalen (optischer Fluss, Differenz / Struktur, Farbe), um den mittels Kamera erfassten Bereich in Fahrtrichtung des Roboters in befahrbaren Untergrund und Hindernisse zu segmentieren. Voraussetzung für die Funktionsweise des Verfahrens ist, dass die Hindernisse genügend Struktur aufweisen, was zwar oft gegeben ist, was allerdings auch zu prinzipiellen Einschränkungen für die robuste Funktion führt. Erste experimentelle Ergebnisse bestätigen, dass das Verfahren für die reale BauMarkt-Umgebung einsetzbar ist.

Angestrebt wird eine kombinierte Nutzung der Sonarsensoren und der visuellen Untergrund-Hindernis-Segmentation mit dem Ziel, damit den Gegebenheiten der realen Einsatzumgebung besser Rechnung tragen zu können. Experimentelle Untersuchungen zur Integration beider Verfahren sind Gegenstand aktueller Arbeiten am Fachgebiet und lagen zum Abschluss dieser Arbeit noch nicht vor.

Globale Navigation

Kartenaufbau Hinsichtlich des Aufbaus konsistenter Umgebungskarten konnte ein Stand erreicht werden, der eine sichere und effiziente Handhabung gestattet. Genutzt wurde wiederum

ein aus der Literatur bekannter probabilistischer Algorithmus, der die Belegtheitsinformation der lokalen Umgebungskarten zu einer konsistenten globalen Umgebungskarte integrativ fusioniert. Der Einsatz einer Odometriekorrektur, die auf der kamerabasierten Auswertung der Fußbodenstruktur der Einsatzumgebung beruht, sichert, dass der Odometriefehler klein genug gehalten werden kann, um eine konsistente Karte des gesamten BauMarktes aufzubauen. Der Vorteil dieser Vorgehensweise besteht aus pragmatischer Sicht darin, dass aufwendigere und konzeptionell mächtigere Algorithmen zum Kartenaufbau bei gleichzeitiger Schätzung der aktuellen Roboterpose bezüglich der gerade aufgebauten Karte (z.B. EM-Algorithmus) umgangen werden können.

Da letztlich jedoch keine Informationen über den tatsächlich vorliegenden Odometriefehler vorliegt, kann mit diesem Verfahren kein Betrieb des Roboters über längere Zeit und damit längere Fahrstrecken bei hinreichend exakter Zustandsschätzung realisiert werden. Dies führte unmittelbar zur Entwicklung eines Verfahrens zur visuell-basierten Zustandsschätzung.

Visuell-basierte Zustandsschätzung Neben den o.g. Aspekten lieferte die Struktur der Einsatzumgebung (regelmäßige Anordnung weitgehend gleichförmiger Regalreihen über große Bereiche des BauMarktes) eine weitere Motivation für die Entwicklung einer Methode zur visuell-basierten Zustandsschätzung. Die bei Verwendung von Abstandssensorik im Falle der Selbstlokalisierung zwangsläufig auftretenden Mehrdeutigkeiten bezüglich der aktuellen Roboterpose können durch die Verwendung visueller Information wirkungsvoll reduziert werden. Das Grundprinzip des Verfahrens basiert auf dem MCL-Algorithmus, der seit einigen Jahren als robustes und effizientes Werkzeug zur Zustandsschätzung zunehmend Verbreitung findet.

Die experimentellen Ergebnisse zur visuell-basierten Zustandsschätzung sind soweit ermutigend, dass eine erfolgreiche Umsetzung dieses Verfahrens für das gesamte BauMarkt-Areal ohne prinzipielle methodische Probleme möglich erscheint. Auch diese Themenstellung ist Gegenstand einer aktuell laufenden Arbeit, ebenso wie die Verbindung des ultraschall-basierten Kartenaufbaus mit der automatischen Generierung eines visuellen Umgebungsmodells.

Pfadplanung und Bewegungssteuerung

Die eigentliche Pfadplanung beruht auf dem aus der Literatur bekannten Dijkstra-Algorithmus. Zur Gewährleistung eines ruhigen und glatten Fahrverhaltens bei sicherer Kollisionsvermeidung wurde die Hindernisvermeidung implizit in die Pfadplanung mit integriert, indem eine Abbildung des Hindernisabstandes auf die Weglänge des geplanten Pfades erfolgt. Damit konnte der prinzipielle Algorithmus zur Pfadplanung beibehalten werden.

Die ursprünglich eingesetzte Potentialfeld-Methode zur Kollisionsvermeidung bzw. Bewegungssteuerung erwies sich für die vorliegende Einsatzumgebung als nur bedingt geeignet. Mit dem implementierten alternativen Verfahren auf der Basis von Vektorfeld-Histogrammen konnten die Probleme der Potentialfeld-Methode überwunden werden, so dass letztlich ein effizientes Verfahren zur Bewegungssteuerung vorliegt.

Offene Probleme

Trotz aller Fortschritte bezüglich der angestrebten robusten Hinderniswahrnehmung konnte dieser Gesamtkomplex noch nicht so weit entwickelt werden, dass ein unbeaufsichtigtes Agieren

des Shopping-Assistenten im BauMarkt sichergestellt werden kann. Dies liegt jedoch vorrangig in der noch fehlenden Integration des visuellen Verfahrens zur Hinderniswahrnehmung begründet. Aktuell laufende Arbeiten widmen sich genau diesem Aspekt, so dass mit experimentellen Ergebnissen zur Validierung des Gesamtsystems zur Hinderniswahrnehmung in den kommenden Monaten gerechnet werden kann.

Perspektivisch Probleme bereiten dürfte die Tatsache, dass gegenwärtig noch kein Ansatz zur flexiblen Adaption der globalen Umgebungsrepräsentation existiert, der beispielsweise die ständige Neuordnung von bestimmten Marktbereichen berücksichtigt. Hier ist noch konzeptionelle Arbeit zu leisten.

Gleiches gilt für die Behandlung des Falles, dass ein geplanter Pfad aufgrund neuer Hindernissituationen nicht erfolgreich umgesetzt werden kann.

Kapitel 8

Mensch-Roboter-Interaktion

Die folgenden Abschnitte beschreiben detailliert die Methoden, die für die Interaktion des Serviceroboters mit seinen potentiellen Benutzern entwickelt wurden. Wie bereits mehrfach betont, liegt das Hauptaugenmerk darauf, dass sich die Interaktion zwischen Mensch und Maschine möglichst intuitiv und natürlich gestaltet.

Das in Abschnitt 2.3 vorgestellte Paradigma der gezielten, selektiven Informationsaufnahme spielt vor dem Hintergrund der vorrangig verwendeten visuellen Sensorik hier eine wesentliche Rolle. Dieses von der Untersuchung biologischer informationsverarbeitender Systeme abgeleitete Grundprinzip, nur die Information aus den vorhandenen sensorischen Datenströmen zu extrahieren, die im aktuellen Verhaltenskontext wesentlich ist, lässt sich unmittelbar auf das technische System Serviceroboter übertragen. Ebenso wie ein biologisches System verfügt der Serviceroboter nur über limitierte Ressourcen, die möglichst effizient genutzt werden müssen. Während bei biologischen Systemen die entsprechenden Verarbeitungsmechanismen im Verlauf der Evolution entwickelt wurden, müssen diese für das technische System derzeit noch von den Entwicklern entworfen und umgesetzt werden.

8.1 Aufmerksamkeitssteuerung

Der Shopping-Assistent soll nicht passiv darauf warten, dass sich ein Kunde bei ihm einbucht, sondern ganz analog zu einem menschlichen Berater nach potentiellen Interaktionspartnern Ausschau halten und zu diesen dann möglichst gezielt Kontakt aufnehmen. Um ein solches Verhalten zu erreichen, muss der Roboter in die Lage versetzt werden, Menschen in seiner Einsatzumgebung wahrzunehmen, denen er sich dann gezielt zuwenden kann. Zu diesem Zweck wurden sowohl visuell- als auch akustisch-basierte Verfahren implementiert, um Hypothesen über das Vorhandensein von Personen in der Einsatzumgebung des Roboters zu erhalten.

Das angestrebte Verhalten, welches durch die Aufmerksamkeitssteuerung realisiert werden soll, lässt sich durch folgende Aspekte zusammenfassen:

- Solange sich keine Person im Sensorbereich des Roboters aufhält, soll auch keine Aufmerksamkeit erzeugt werden.
- Sobald eine Person im Sensorbereich erscheint, soll diese umgehend wahrgenommen werden, also die Aufmerksamkeit des Roboters auf sich ziehen.

- Beim Auftreten mehrerer Personen soll die Person, welche sich direkt auf den Roboter zu bewegt, eine höhere Aufmerksamkeit erzeugen.
- Wenn Aufmerksamkeit generiert wurde, soll umgehend ein Zuwendungsverhalten zum Ort der höchsten Aufmerksamkeit eingeleitet werden, um dem potentiellen Interaktionspartner eine umgehende Reaktion anzubieten.

Das implementierte System zur Aufmerksamkeitssteuerung wurde so ausgelegt, dass ein potentieller Interaktionspartner sowohl über den visuellen (Bewegungsinformation) als auch den akustischen (Geräuschortung) Kanal wahrgenommen werden kann (Abschnitte 8.1.1 und 8.1.2). In einem daran anschließenden Verfahren werden die so generierten Hypothesen durch eine Personenverifikation abschließend validiert (Abschnitt 8.2). Dies soll sicherstellen, dass sich der Roboter auch wirklich nur Personen zuwendet und möglichst nicht versucht, mit unbelebten Gegenständen, wie etwa vorbeifahrenden Staplern oder Einkaufswagen, in Kontakt zu treten.

Für die Einbettung des Aufmerksamkeitssystems in den gesamten Interaktionszyklus sei auf Kapitel 9 und Abbildung 9.3 verwiesen.

8.1.1 Visuelle Aufmerksamkeit

Ausgegangen wird von einem ruhenden Roboter, der nach potentiellen Interaktionspartnern Ausschau hält. Die omnidirektionale Kamera, die einen vollständigen Überblick über die Einsatzumgebung liefert, wird zur Generierung von Hypothesen über vorhandene Personen in der Umgebung von PERSES herangezogen.

Da sich Personen typischerweise bewegen, erfolgt eine Bewegungsanalyse im omnidirektionalen Datenstrom. Das verwendete Verfahren besitzt Ähnlichkeiten zu der im PFINDER-System [WADP97] eingesetzten Methode, wurde jedoch in erster Linie aus Performanzgründen stark vereinfacht [Hem00] [BWK⁺01a] [BWK⁺01b].

Für die Bewegungsberechnung wird das aktuelle Farbbild ($\underline{a}(x, y)(t)$) von einem über die Zeit berechneten Hintergrundbild ($\underline{h}(x, y)(t)$) bandweise abgezogen. Wenn die Differenz einen definierten Schwellwert (Γ_1) überschreitet, so wird eine Bewegung an der entsprechenden Bildposition unterstellt.

$$m_A(x, y)(t) = \begin{cases} 1; & \|\underline{a}(x, y)(t) - \underline{h}(x, y)(t+1)\| \geq \Gamma_1 \\ 0; & \text{sonst} \end{cases} \quad (8.1)$$

Damit kann ein Binärbild ($m_A(x, y)(t)$) erzeugt werden, in welchem vermerkt ist, ob ein Pixel zum Hintergrund oder zum Vordergrund gehört. Basierend auf diesem Bild wird das Hintergrundbild ähnlich einem Tiefpass entsprechend der folgenden Formel adaptiert.

$$\begin{aligned} \underline{h}(x, y)(t+1) &= \begin{cases} \alpha * \underline{h}(x, y)(t) + (1 - \alpha) * \underline{a}(x, y)(t); & m_A(x, y)(t) = 0 \\ \underline{h}(x, y)(t); & \text{sonst} \end{cases} \\ \underline{h}(x, y)(0) &= 0 \end{aligned} \quad (8.2)$$

Der Wichtungsfaktor α ($0 \leq \alpha \leq 1$) bestimmt, wie stark der Farbwert des aktuellen Pixels in die Adaption des Hintergrundbildes eingeht. Dank dieser Adaption können nicht sprunghaft auftretende Beleuchtungsschwankungen gut kompensiert werden.

Um nicht voraussetzen zu müssen, dass sich zu Beginn der Berechnungen keine bewegten Objekte im Bild befinden dürfen, wurde die mittelwertbasierte Segmentation um ein Historienbild ($history_A(x, y)(t)$) erweitert.

$$\begin{aligned} history_A(x, y)(t) &= \begin{cases} history_A(x, y)(t-1) + 1; & m_A(x, y)(t) = 1 \\ 0; & sonst \end{cases} \\ history_A(x, y)(0) &= 0 \end{aligned} \quad (8.3)$$

In diesem wird verzeichnet, wie lange ein Pixel schon als Vordergrund segmentiert wurde. Überschreitet dieser Wert eine definierte Schwelle, so wird der Farbwert des Pixels, trotz der großen Differenz, als Hintergrund definiert und das Hintergrundmodell entsprechend Gleichung 8.2 angepasst.

$$m_A(x, y)(t) = \begin{cases} 1; & m_A(x, y)(t) = 1 \wedge history_A(x, y)(t) \geq \Gamma_2 \\ 0; & sonst \end{cases} \quad (8.4)$$

Dies kann dazu führen, dass auch in der Szene vorhandene Personen als Hintergrund betrachtet werden, nämlich genau dann, wenn diese sich für eine bestimmte Zeit nicht bewegen. Dieser Effekt wirkt sich jedoch nicht negativ auf das Gesamtverhalten aus, da diese Personen bei deren erneuter Bewegung sofort wieder wahrgenommen werden können.

Ein Dilations-Erosions-Filter, das der eigentlichen Bewegungsdetektion vorgeschaltet ist, reduziert das zwangsläufig auftretende Pixelrauschen und sorgt dafür, dass die Schwelle Γ_1 in Gleichung 8.1 relativ niedrig angesetzt werden kann, um Bewegung im omnidirektionalen Bild möglichst unmittelbar wahrnehmen zu können.

Die große radiale optische Verzeichnung und die geringe Auflösung können problemlos vernachlässigt werden, da Objekte in der näheren Umgebung größere Bewegungssegmente erzeugen und auf die exakte Bestimmung des Radius kein Wert gelegt wird. Lediglich die Information, ob sich das Objekt auf den Roboter zu bewegt oder von ihm weg, ist von Interesse.

Für jeden Winkel wird also eine Bewertung (q_α^{mov}) errechnet, wie gut (oder schlecht) es wäre, in dieser Region nach Interaktionspartnern zu suchen.

$$q_\alpha^{mov} = \prod_{i=0}^n q_\alpha^i \quad (8.5)$$

Jedes q_α^i repräsentiert dabei eine bestimmte Teilqualität, wobei alle möglichst hoch sein müssen, um q_α^{mov} zu maximieren. Diese Teilqualitäten sind

- q_α^0 : Entfernung des Objekts zum Roboter
- q_α^1 : Bewegungsrichtung des Objekts
- q_α^2 : zu erwartender Drehwinkel

Entfernung des Objekts zum Roboter Im BauMarkt-Szenario kann keineswegs davon ausgegangen werden, dass sich immer nur eine Person im Sichtbereich des Shopping-Assistenten

bewegen wird. Aus diesem Grund wird ein Kriterium definiert, das anhand der Größe des Bewegungssegments über die Entfernung der Person zum Roboter und damit über die Qualität des Winkels entscheidet. Die Größe eines Segments im Winkel α ist definiert als Differenz zwischen maximalem und minimalem Radius ($r_\alpha^{max}, r_\alpha^{min}$), auf dem Bewegung detektiert wurde. Die Wurzelfunktion rührt aus der Abbildungsvorschrift des Parabolspiegels der omnidirektionalen Kamera her.

$$q_\alpha^0 = \sqrt{r_\alpha^{max} - r_\alpha^{min} + 1} \quad (8.6)$$

Bewegungsrichtung des Objekts Stellt man sich die gesuchte Bewegung (in Richtung des Roboters) im omnidirektionalen Bild vor, so stellt man fest, dass eine Person, die sich auf das System zu bewegt, in immer kleineren Radien Bewegungsinformation im omnidirektionalen Bild induziert. Somit kann man für jeden Winkel, unter dem Bewegung detektiert wurde, den minimalen Radius bestimmen, und, falls sich dieser Wert in aufeinanderfolgenden Bildern verkleinert, auf eine Bewegung in Richtung des Roboters schließen.

$$q_\alpha^1 = \begin{cases} 1; & r_\alpha^{min}(t) - r_\alpha^{min}(t-1) > 0 \\ 0.5; & \text{sonst} \end{cases} \quad (8.7)$$

Der Wert des Wichtungsfaktors α von 0.5 stellt sicher, dass auch Personen, die sich nicht direkt auf den Roboter zu bewegen, die Aufmerksamkeit von PERSES auf sich ziehen können.

Zu erwartender Drehwinkel Für die Zuwendung des Serviceroboters zu einer interessanten Region (Roboter soll mit den Frontalkameras in die entsprechende Richtung schauen) muss neben der Bewegungsrichtung des Objekts auch die momentane Ausrichtung (Orientierung) des Roboters und damit die Abweichung des Regionen- bzw. Winkelkandidaten α zur aktuellen Orientierung Φ betrachtet werden. Gleichung 8.8 zeigt die entsprechende Berechnungsvorschrift auf.

$$q_\alpha^2 = \begin{cases} 0.75 * \cos\left(\frac{|\alpha - \phi|}{\theta}\right) + 0.25 & ; \quad |\alpha - \phi| \leq \theta \\ 0.25 & ; \quad \text{sonst} \end{cases} \quad (8.8)$$

θ stellt dabei den bevorzugten Winkelbereich im Bereich von $\Phi - \theta/2$ bis $\Phi + \theta/2$ dar. Da das Rotieren des gesamten Roboters eine relativ lange Verzögerung darstellt, während das Drehen der Frontalkameras innerhalb kurzer Zeit abgeschlossen ist, kann mit Hilfe des Parameters θ festgelegt werden, wo das System zwischen alleiniger Drehung der Frontalkamera(s) und Roboterrotation umschalten soll. Ein weiterer Vorteil, nicht sofort das gesamte Robotersystem um nur wenige Grad drehen zu müssen, liegt darin, dass die nach jeder Rotation notwendige Initialisierung der Bewegungsdetektion vermieden werden kann. Durch die Verwendung des Parameters q_α^2 wird somit sichergestellt, dass sich der Roboter bevorzugt Regionen (Personen) zuwendet, die sich in etwa vor ihm aufhalten.

Die beschriebene visuelle Komponente der Aufmerksamkeitssteuerung könnte mit Hilfe eines Laserscanners, der jedoch nicht zur Verfügung stand, in ähnlicher Weise realisiert werden. Experimentelle Untersuchungen zum Einsatz der Sonarsensoren erbrachten jedoch keine zufriedenstellenden Ergebnisse hinsichtlich Winkelauflösung und Robustheit der Richtungsauswertung, da eine zu starke Abhängigkeit des Sonarsignals vom Auftreffwinkel und der vorhandenen Bekleidung vorliegt. Angedacht ist jedoch eine Integration der Sonarinformation, was letztlich wieder zu einem multimodalen Verfahren führen wird.

8.1.2 Akustische Aufmerksamkeit

Ein Verfahren zur Geräuschlokalisierung, welches auf der Modellierung der menschlichen auditorischen Informationsverarbeitung basiert (siehe [SZPG00] [SPG01]), bietet dem Interaktionspartner die Möglichkeit, auch akustisch die Aufmerksamkeit auf sich zu ziehen. Damit stehen dem Roboter zwei alternative Aufmerksamkeitstrigger zur Verfügung, die in einem Fusionsmodul (Abschnitt 8.1.3) integriert wurden, welches letztlich die Richtung determiniert, welcher sich der Roboter dann zuwendet. Die Abbildung 8.1 soll hier nur das grobe Prinzip verdeutlichen.

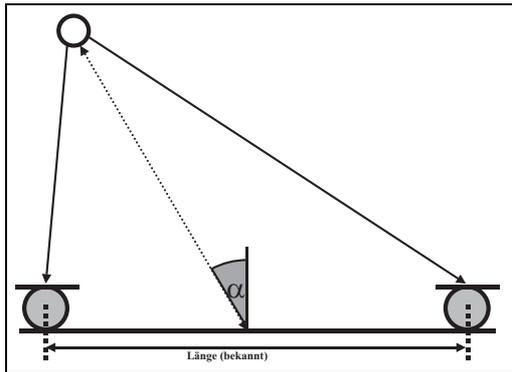


Abbildung 8.1: Prinzipskizze der Geräuschquellenlokalisierung

Aus dem Laufzeitunterschied eines Geräusches zwischen rechtem und linkem Kanal und dem bekannten Basisabstand zwischen den Mikrofonen kann über effiziente Korrelationsverfahren eine Richtungsschätzung vorgenommen werden. Damit diese Korrelation nicht ständig berechnet werden muss, wird auf dem akustischen Datenstrom eine so genannte OnSet-Erkennung ausgeführt. Die nächsten N Samples nach dem OnSet werden dann zur Richtungsschätzung herangezogen.

Die errechnete Richtung (α_{sound}^0) wird im Intervall $[-\Gamma_3 : +\Gamma_3]$ um α_{sound}^0 und $\alpha_{sound}^1 = \alpha_{sound}^0 + 180^\circ$ in Gleichung 8.10 als q_α^{sound} eingekoppelt. Die Gleichung 8.9 beschreibt die Berechnung von q_α^{sound} .

$$q_\alpha^{sound} = \begin{cases} 1; & (\alpha_{sound}^0 - \Gamma_3 \leq \alpha \leq \alpha_{sound}^0 + \Gamma_3) \wedge (\alpha_{sound}^1 - \Gamma_3 \leq \alpha \leq \alpha_{sound}^1 + \Gamma_3) \\ 0; & \text{sonst} \end{cases} \quad (8.9)$$

Γ_3 stellt dabei den Winkelbereich (ca. 5° - 10°) dar, in dem das Ergebnis der Geräuschlokalisierung eingekoppelt wird.

8.1.3 Fusion der Aufmerksamkeitsquellen

Nachdem für jeden einzelnen Winkel sowohl visuelle als auch akustische Aufmerksamkeitswerte ermittelt wurden, gilt es nun, diese zu fusionieren und einen endgültigen Drehwinkel zu bestimmen. Dies geschieht mittels einer einfachen linearen Superposition nach Gleichung 8.10.

$$\tilde{q}_\alpha = w^{mov} \cdot q_\alpha^{mov} + w^{sound} \cdot q_\alpha^{sound} \quad (8.10)$$

Die beiden Wichtungskoeffizienten bestimmen, wie stark die Bewegungsdetektion bzw. die Geräuschlokalisierung in das Ergebnis einfließen.

Bei der letztendlichen Auswahl des anzusteuern Winkels muss noch ein weiterer Aspekt berücksichtigt werden: Die Winkelgüten spiegeln nicht den gesamten von den Frontalkameras erfassbaren Bildbereich (ca. 30°) wider, sondern nur einen genau definierten Winkel. Eine einfache Box-Filter-Operation reicht aus, um den Kontext eines bestimmten Winkelbereiches in einem Wert zu vereinen. Bei sehr großen Bewegungsregionen (die Person steht schon sehr nahe am Roboter) führt dies allerdings dazu, dass ein relativ ausgedehnter zusammenhängender Winkelbereich entsteht, dessen Elemente alle in etwa die selbe Güte aufweisen. Diesem Effekt kann man mit Hilfe einer Schwerpunktberechnung begegnen und den Schwerpunkt um den Winkel der maximalen Qualität (α_{best}) als anzusteuern Winkel definieren. $\tilde{q}(\alpha)$ ist die in Gleichung 8.10 berechnete Gesamtaufmerksamkeit (Fusion aus visueller und akustischer Aufmerksamkeit).

$$\alpha_{best} = \underset{\alpha}{argmax} \{ \tilde{q}(\alpha) \} \quad (8.11)$$

Die nachfolgende Abbildung 8.2 soll verdeutlichen, wie sich das System beim Auftreten zweier Personen in seiner Einsatzumgebung verhält, wobei hier ausschließlich der visuelle Kanal berücksichtigt wurde.

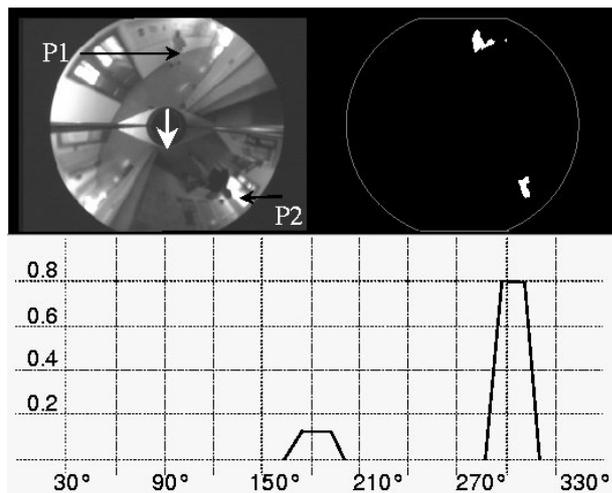


Abbildung 8.2: Bewegungsanalyse im omnidirektionalen Videodatenstrom. Die sich aus etwa 280° -Richtung direkt auf den Roboter zu bewegende Person P2 erzeugt aufgrund der Auswertung der Bewegungsrichtung im Vergleich mit der in ca. 180° -Richtung detektierten Person P1, die sich am Roboter vorbei bewegt, eine deutlich höhere Aufmerksamkeit, die im unteren Diagramm dargestellt ist. Der weiße Pfeil markiert die Orientierung des Roboters, der Winkel läuft im Uhrzeigersinn (aus [BWK⁺ 01a]).

In Abbildung 8.2 ist gut zu erkennen, dass beide Personen bewegungsbasiert wahrgenommen werden und entsprechend im Winkelvektor (Abbildung 8.2 unten) erscheinen. Die Person, die sich auf den Roboter zu bewegt, erzeugt jedoch einen deutlich höheren Aufmerksamkeitswert für die entsprechende Richtung, der sich der Shopping-Assistent anschließend zuwendet.

8.2 Personenverifikation

Nachdem das Aufmerksamkeitssystem Regionen ermittelt hat, in denen bevorzugt nach Interaktionspartnern gesucht werden sollte, und der Roboter sich der aus seiner Sicht erfolgversprechendsten Region zugewandt hat, erfolgt nun eine abschließende Personenverifikation. Dies ist notwendig, da die bislang verwendeten Merkmale zu unspezifisch sind, um daraus sicher auf das Vorhandensein einer Person in der entsprechenden Richtung zu schließen. Genau diese Sicherheit ist jedoch notwendig, um zu vermeiden, dass der Roboter auch andere Objekte seiner

Umgebung fälschlicherweise als Personen identifiziert und versucht, mit diesen in Interaktion zu treten.

Für den Entwurf der Architektur waren folgende Aspekte maßgebend:

- Die Eigenschaften der BauMarkt-Umgebung (siehe Abschnitt 6.2) gebieten nahezu den Einsatz eines Multi-Cue-Systems, da von vornherein nicht zu erwarten ist, dass sich ein einziges personenspezifisches Merkmal unter allen Umgebungsbedingungen als genügend robust erweist. Alle Merkmale sollten parallel berechnet und anschließend miteinander kombiniert werden, um nicht auf das Vorhandensein eines bestimmten Merkmals angewiesen zu sein.
- Für die Kombination der Merkmale und die abschließende Selektion der entsprechenden Position müssen geeignete Mechanismen entwickelt werden, die dies hinreichend effizient und robust gestatten.
- Bei der Personenverifikation handelt es sich um ein dreidimensionales Problem, da lediglich eine Hypothese über die Richtung vorliegt, in der sich eine Person befinden könnte, aber a priori keine Annahmen bezüglich Position und Entfernung möglich sind. Daraus lässt sich die Verwendung einer Multiskalen-Repräsentation ableiten.
- Hinsichtlich der konkret zu verwendenden Merkmale wurde folgende Heuristik herangezogen: Wenn Menschen kommunizieren, wenden sie sich einander zu. Also werden sich Personen, die mit dem Shopping-Assistenten in Interaktion treten möchten, diesem auch zuwenden. Als Konsequenz daraus wurden Merkmale gewählt, die für diesen Fall besonders gut detektierbar sind. Dabei handelt es sich um Hautfarbe, die Silhouette der Kopf-Schulter-Partie und um die Struktur innerhalb des Gesichts.

Im Resultat entstand eine Gesamtarchitektur zur Personenverifikation, die in Abbildung 8.3 überblicksartig dargestellt ist und deren Teilmodule in den folgenden Abschnitten genauer erläutert werden.

Bevor in den Abschnitten 8.2.2 bis 8.2.4 die einzelnen Merkmalsdetektoren motiviert und detailliert beschrieben werden, soll im unmittelbar nachfolgenden Abschnitt auf die pyramidale Repräsentation, die sowohl für die Personenverifikation als auch für das Personentracking (Abschnitt 8.3) eine Basisstruktur darstellt, eingegangen werden.

8.2.1 Pyramidale Repräsentation

Pyramidale Repräsentationen sind eine seit langem in der Bildverarbeitung benutzte praktische und zugleich elegante Weise, bei Nutzung von Operatoren mit abgestuften bzw. gestaffelten spektralen Eigenschaften die Anzahl der auszuführenden Operationen auf ein erheblich vermindertes Maß zu reduzieren. ROSENFELD und JOLION [Ros94] [JR94] geben dazu einen umfassenden Überblick über das spezielle Gebiet der Auflösungspyramiden bzw. des „Multiresolutional Image Processing“ (für eine ausführlichere Diskussion der pyramidalen Repräsentation sei auf [Bra01] verwiesen).

Die Relevanz für die vorliegende Arbeit liegt darin, dass sowohl die Personenverifikation als auch das Personentracking immer das Problem der Entfernungsschätzung einschließen. Als

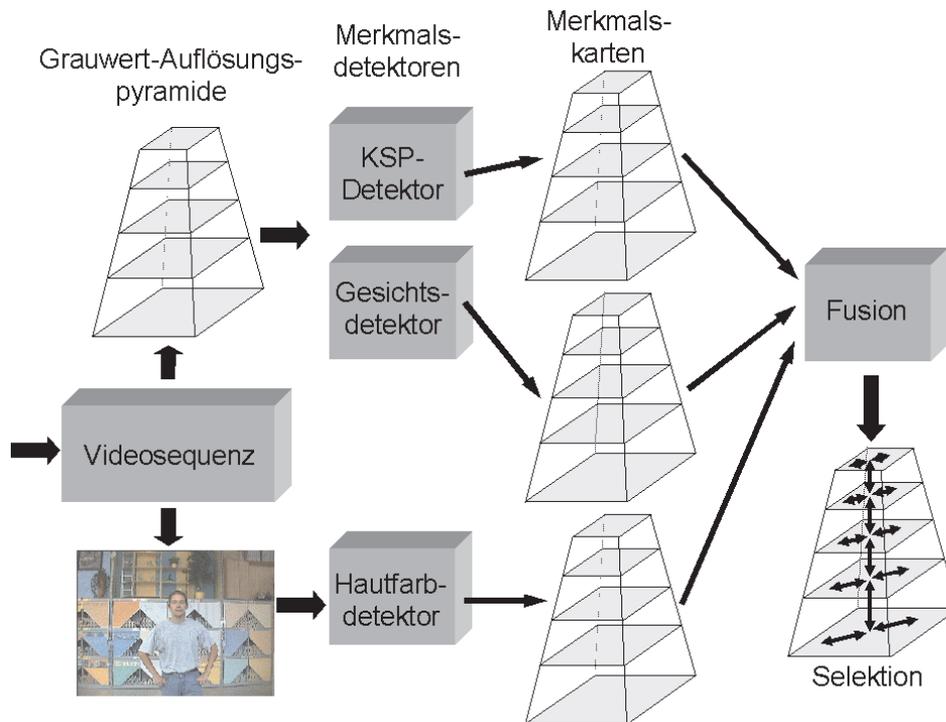


Abbildung 8.3: Gesamtarchitektur zur Personenverifikation. Auflösungs- und Hautfarbklassifizierungspyramiden transformieren das Bild in eine Multiskalenrepräsentation mit 5 Ebenen. Die beiden Cue-Module, die strukturbasiert arbeiten und für die Detektion eines Gesichts bzw. einer Kopf-Schulter-Silhouette verantwortlich sind, operieren auf allen Ebenen der Auflösungs- und Hautfarbklassifizierungspyramide, während die Detektion von Hautfarbe im Originalbild erfolgt. Danach wird das hautfarbklassifizierte Bild ebenfalls in eine Pyramide transformiert, um eine einheitliche Datenstruktur zur Fusion der einzelnen Cues zu erhalten. Anschließend werden die so erhaltenen pyramidalen Merkmalskarten einem Fusionsmodul zugeführt, das die Einkopplung der einzelnen Merkmale in einen wiederum pyramidal organisierten Selektionsprozess vornimmt.

mindestens erforderlich wird angesehen, in einem Entfernungs- und damit Skalierungsintervall von zwei Oktaven diese Entfernungsschätzung vornehmen zu können, was ausgehend von einer festen minimal zulässigen Entfernung einen Spielraum bis zur Vervierfachung dieser Entfernung lässt. Daher rührt die Forderung, zumindest einen Teil der notwendigen Operationen in aneinander anschließenden spektralen Intervallen auszuführen und somit indirekt über die spektrale Lage (Größe) der Merkmale auf die Entfernung der Person zur Kamera schließen zu können.

Das führt zunächst dazu, einen Satz entsprechend abgestufter Operatoren zu entwerfen und die jeweiligen Operationen auszuführen. Dabei entsteht das Paradoxon, dass mit den aus diesem Satz größten und damit für die Rechnung aufwendigsten Operatoren nach den im Bild größten (und damit auch in der Abbildung am größten erscheinenden) enthaltenen jeweiligen Merkmalen gesucht wird.

Um dieses Paradoxon aufzulösen, besteht umgekehrt die Möglichkeit, auch ohne Informationsverlust das Bild zu verkleinern und *danach* die Operation auszuführen, sofern hierfür die durch die Verkleinerung wegfallenden höherfrequenten Bildbestandteile nicht relevant sind. Die

Skalierung wird prinzipiell durch Unterabtastung einer Bildmatrix einer bestimmten Größe vorgenommen und resultiert in einer neuen Bildmatrix mit einer entsprechend verminderten Größe.

8.2.1.1 Abstufung und Glättung

Ein fehlerhaftes Vorgehen kann jedoch zu unerwünschten Ergebnissen führen, da zu beachten ist, dass hierbei das Abtasttheorem erfüllt bleibt. Dieses besagt, dass eine periodische Struktur nur dann aus Abtastwerten fehlerfrei rekonstruierbar bleibt, wenn die Abtastung dabei mindestens zweimal pro Wellenlänge erfolgt. Es muss also gesichert sein, dass *vor* der Unterabtastung idealerweise eine Beseitigung von spektralen Anteilen oberhalb $s \cdot k_{\max}$ erfolgt, wobei s den Skalierungsfaktor und k_{\max} die maximale Ortsfrequenz darstellen. Andernfalls würden Strukturen enthalten bleiben, die nach erfolgter Unterabtastung *nicht* mehr hinreichend oft abgetastet wären. Damit ginge für diese Strukturen dem Theorem entsprechend die Rekonstruierbarkeit verloren, es käme zu falschen Rekonstruktionen (Aliasing-Effekt).

In der Praxis sind spektrale Anteile oberhalb einer bestimmten Frequenz effektiv dämpfbar mit Tiefpassfiltern, die im Ortsraum als Faltung (oder in erweiterter Form als Verknüpfung von Faltungen) mit bestimmten geeigneten Masken ausgeführt werden können. Die Wahl des Tiefpasses hängt jedoch vom Abstufungsintervall der Auflösungsstufen innerhalb der Pyramide ab, also vom Skalierungsfaktor s . Da von vornherein keine Unterscheidung im entfernungsabhängigen Auflösungsbedarf der Entfernungsbestimmung gemacht werden soll, ist darüber hinaus eine *äquidistante* Abstufung der Pyramiden und damit der Einsatz *eines* allgemeinen Skalierungsfaktors sinnvoll.

Ein typisches Abstufungsintervall von Auflösungsstufen, die in der Literatur vorgestellt werden, ist der *Oktavabstand* mit einem Skalierungsfaktor von $s = 2^{-1}$. Dies entspricht also einer Halbierung der Auflösung von Ebene zu Ebene der Pyramide. Vorteil ist, dass der Zuwachs an Bilddaten (über die gesamte Pyramide gesehen) die theoretische Grenze von einem Drittel des Ursprungsdatenumfangs nicht überschreiten kann. Hinzu kommt der Wegfall des Bedarfs von Zwischenraster-Interpolationen. Als entscheidender Nachteil muss allerdings angesehen werden, dass der Oktavabstand sich leider für die Entfernungsbestimmung zu grob erweist, da z.B. die Merkmalsextraktionsmethoden für die Gesichtsdetektion eine geringere Skalierungsinvarianz als über eine Oktave aufweisen.

Dieser Umstand wurde von anderen Autoren (z.B. VIENNET und FOGELMAN SOULIÉ [VF92]) ebenfalls erkannt, die sich stattdessen für den *Halboktavabstand* entschieden haben. Wenngleich damit Aufwand und Datenmenge der Pyramide steigen, liegt damit die Genauigkeit der Entfernungsschätzung potentiell doppelt so hoch. Der Skalierungsfaktor beträgt in diesem Falle $s = 2^{-\frac{1}{2}}$.

Bei der Auswahl eines geeigneten Tiefpasses beschränkt man sich typischerweise auf lineare Operatoren. Als weiteres Auswahlkriterium kommt der Grad der Dämpfung oberhalb der Wellenzahl $k_{\text{grenz}} = s \cdot k_{\max}$ hinzu. Einen akzeptablen Kompromiss der Dämpfungseigenschaften weisen Binomialfilter auf. Es zeigt sich, dass bereits ein Binom 2. Ordnung zur Bildung eines geeigneten zweidimensionalen Binomialfilters ausreicht.

8.2.1.2 Unterabtastung und Interpolation

Nachdem im vorangegangenen Abschnitt das Thema Glättung als Voraussetzung für die Erzeugung einer pyramidalen Repräsentation behandelt worden ist, soll nun mit Unterabtastung und Interpolation der eigentliche Schritt dazu dargestellt werden. Das Bild (einzeln Farbauszug bzw. Grauwertmatrix) auf der n -ten Pyramidenebene $\mathbf{I}^{(n)}$ wird dafür mittels zweidimensionalem Binomialfilter ${}^2\mathbf{B}$ tiefpassgefiltert und dann mit einem Unterabtast- und Interpolationsoperator \mathbf{R} behandelt, so dass sich die nächste Pyramidenebene formal folgendermaßen bilden lässt:

$$\mathbf{I}^{(n+1)} = \mathbf{R} ({}^2\mathbf{B} * \mathbf{I}^{(n)}) \quad (8.12)$$

Was verbirgt sich hinter dem Unterabtast- und Interpolationsoperator \mathbf{R} ? In einem Doppelschritt werden zunächst entsprechend eines Skalierungsfaktors (hier $s = \frac{1}{\sqrt{2}}$) die i. a. auf Zwischengitterplätzen liegenden Abtastkoordinaten bestimmt. Im einfachsten Fall könnten dafür die Werte des jeweiligen nächsten Nachbarn verwendet werden, wobei damit auf eine Interpolation im eigentlichen Sinne verzichtet würde.

Von den möglichen Interpolationsmethoden erscheint ein lineares Verfahren aufwandsmäßig adäquat. Bei der so genannten *bilinearen Interpolation* werden die vier Nachbarn eines solchen Zwischengitterplatzes zur Bestimmung eines Abtastwertes linear gewichtet einbezogen. Diese Interpolation erfolgt für die Koordinate k, l einer reduzierten Matrix an den i. a. auf Zwischengitterplätzen liegenden reellwertigen Koordinaten $s^{-1} \cdot k, s^{-1} \cdot l \in \mathbb{R}$, die daher jeweils zwischen zwei diskreten nichtnegativen ganzzahligen Koordinaten $p, q \in \mathbb{N}$ und $p + 1, q + 1 \in \mathbb{N}$ liegen, so dass gilt $p \leq s^{-1} \cdot k < p + 1$ und $q \leq s^{-1} \cdot l < q + 1$.

Sei $\tilde{i}^{(n)}(p, q)$ ein Bildpunkt eines (bereits geglätteten) Bildes ${}^2\mathbf{B} * \mathbf{I}^{(n)}$ (n -te Pyramidenebene), der zusammen mit den drei weiteren Bildpunkten $\tilde{i}^{(n)}(p + 1, q)$, $\tilde{i}^{(n)}(p, q + 1)$ und $\tilde{i}^{(n)}(p + 1, q + 1)$ die vier unmittelbar benachbarten Eckpunkte eines minimalen Matrixausschnitts bildet. Dann liegen die Koordinaten des abzutastenden Punktes, dessen Wert durch Interpolation gefunden werden muss, genau innerhalb dieses Ausschnitts. Die Vorschrift für die bilineare Interpolation dieses Abtastwertes für den Bildpunkt $i^{(n+1)}(k, l)$ der nächsten Pyramidenebene lautet:

$$\begin{aligned} i^{(n+1)}(k, l) = & \tilde{i}^{(n)}(p, q) \Delta p_2 \Delta q_2 + \\ & \tilde{i}^{(n)}(p, q + 1) \Delta p_2 \Delta q_1 + \\ & \tilde{i}^{(n)}(p + 1, q) \Delta p_1 \Delta q_2 + \\ & \tilde{i}^{(n)}(p + 1, q + 1) \Delta p_1 \Delta q_1 \end{aligned} \quad (8.13)$$

Darin sind $\Delta p_1 = s^{-1} \cdot k - p$ und $\Delta q_1 = s^{-1} \cdot l - q$ sowie $\Delta p_2 = 1 - \Delta p_1$ und $\Delta q_2 = 1 - \Delta q_1$ die jeweiligen Koordinatendistanzen zu den Koordinaten der Gitterpunkte.

Damit steht mit den Gleichungen 8.12 und 8.13 ein komplettes algorithmisches Instrumentarium für die Bildung der pyramidalen Repräsentation zur Verfügung. Abbildungen 8.4 soll diese abschließend für eine Beispielszene veranschaulichen, wobei hier jeder der drei Farbauszüge einzeln auf die oben beschriebene Weise verarbeitet wurde.

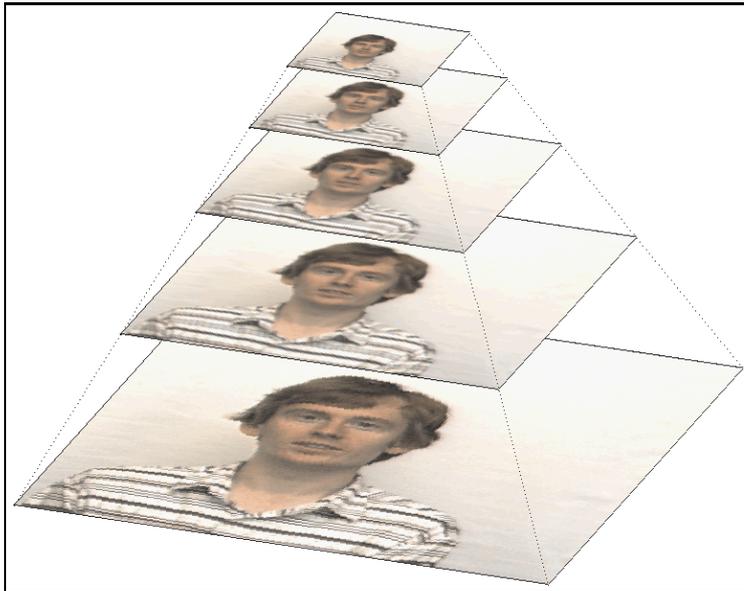


Abbildung 8.4: Auflösungsstufen der pyramidalen Repräsentation: Durch den gewählten Halboktavabstand bedingt, beträgt das Intervall zwischen der jeweils auf der obersten und der untersten Auflösungsebene maximal darstellbaren Wellenzahl k_{\max} genau zwei Oktaven bei fünf umfassten Auflösungsebenen (aus [Bra01]).

8.2.2 Hautfarbanalyse

Hautfarbe spielt in vielen Verfahren der Mensch-Maschine-Interaktion eine tragende Rolle. Jedoch handelt es sich bei der Hautfarbe keinesfalls um ein eindeutiges Merkmal, obgleich es die Hypothese, dass sich an dieser Stelle eine Person befindet, wirkungsvoll unterstützen kann. Dies hat im wesentlichen zwei Ursachen:

- Hautfarbe kann zwar als charakteristisch für Hautpartien angesehen werden, jedoch treten in realen Umgebungen verschiedenste Objekte auf, die ebenfalls ähnliche spektrale Anteile aufweisen.
- Das Erscheinungsbild von Hautfarbe ist stark abhängig von den vorherrschenden Beleuchtungsbedingungen. Würde man alle dabei auftretenden Möglichkeiten in einem Hautfarbmodell berücksichtigen, so wäre dieses für den angestrebten Einsatzfall viel zu unspezifisch.

Gesucht ist also zunächst ein Hautfarbmodell, das eine gute Balance zwischen Spezifität und Sensitivität aufweist. Um ein solches Modell zu generieren, wurde eine Testreihe von Bildern aus der realen BauMarkt-Umgebung aufgenommen, die verschiedene Personen enthält. Anschließend erfolgte eine manuelle Selektion hautfarbener Pixel, die in einem Hautfarbdatensatz zusammengefasst wurden. Analog zu der in [YLW97] vorgeschlagenen Vorgehensweise wird diese Verteilung von Farbwerten mit einem möglichst einfachen Modell mit wenigen Parametern modelliert. Als Hautfarbmodell dienen die intensitätsnormierten Farbanteile r und g (dichromatische Farben) aller Farbtripel (R,G,B) des Hautfarbdatensatzes (Abbildung 8.5), beschrieben durch deren Mittelwert und Kovarianzmatrix.

$$r = \frac{R}{R + G + B} \quad g = \frac{G}{R + G + B} \quad (8.14)$$

Der Mahalanobis-Abstand zwischen dem Farbwert des betrachteten Pixels und dem Hautfarbmodell bestimmt nun die Zugehörigkeit des Pixels zur Klasse **Hautfarbe**.

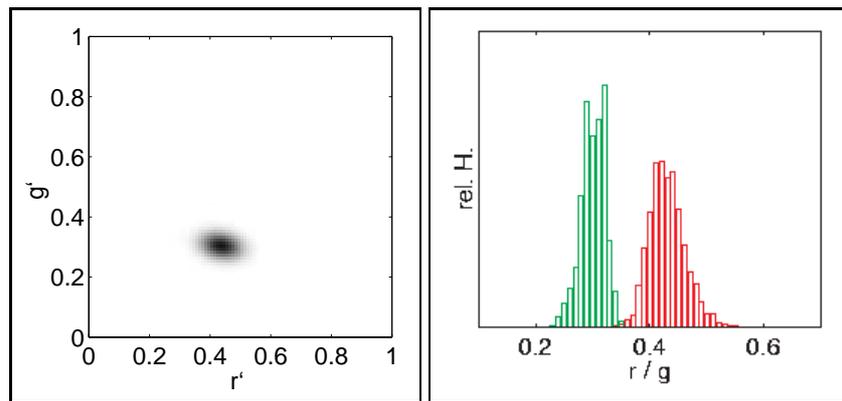


Abbildung 8.5: Verwendetes Hautfarbmodell (links, bivariate Gaußverteilung) als Draufsicht auf die r - g -Ebene, rechts die Häufigkeit der Farbwerte entlang der r - bzw. g -Achse.

Für die weitere Behandlung dieses ersten Zwischenergebnisses wurden folgende Prämissen und heuristische Annahmen definiert:

- Überschreitet der Mahalanobis-Abstand eine vorgegebene Schwelle, so wird die Wahrscheinlichkeit, dass dieses Pixel Hautfarbe darstellt, mit 0 bewertet. Entsprechend gilt umgekehrt, dass im Fall der Unterschreitung dieser Schwelle die Wahrscheinlichkeit, dass dieses Pixel Hautfarbe darstellt, mit 1 bewertet wird. Damit wird eine Binarisierung des hautfarbklassifizierten Bildes erreicht, was die nachfolgenden Verarbeitungsschritte deutlich vereinfacht.
- Zunächst sollten die hautfarbklassifizierten Pixel zu zusammenhängenden Regionen gruppiert werden.
- Das Verhältnis von Höhe zu Breite eines so gewonnen Hautfarbsegments sollte für den Fall, dass es sich dabei um ein Gesicht handelt, innerhalb eines bestimmten Intervalls liegen.
- Ein Hautfarbsegment hat, sofern es sich dabei um ein Gesicht in einer entsprechenden Entfernung handelt, immer eine bestimmte Größe, d.h. der Flächeninhalt des zu lokalisierenden Gesichtes liegt ebenfalls innerhalb eines definierten Intervalls.

Abbildung 8.6 fasst die nun folgenden Verarbeitungsschritte zusammen. Zunächst werden die hautfarbklassifizierten Pixel (Abbildung 8.6(b)) einer Schwellwertoperation unterzogen, in deren Ergebnis ein binarisiertes hautfarbklassifiziertes Bild vorliegt (Abbildung 8.6(c)). Für die Umsetzung der beiden genannten Heuristiken muss dieses Binärbild segmentiert werden. Dies geschieht, indem zunächst ein binärer Medianfilter angewendet wird, um kleine Lücken innerhalb von Hautsegmenten zu schließen und sehr kleine Hautregionen, die nur einzelne oder einige wenige Pixel umfassen, auszublenden (Abbildung 8.6(d)). Die Segmentierung (Abbildung 8.6(e)) erfolgt dann über ein Zeilenkoinzidenzverfahren, ein Standardverfahren der digitalen Bildverarbeitung [Pav90]. Höhe, Breite und Größe eines Hautfarbsegments fallen bei dem Segmentierungsverfahren als Nebenprodukt mit ab. Mit Hilfe der beiden letztgenannten Annahmen lassen sich alle unregelmäßigen oder übergroßen Hautfarbregionen aus dem hautfarbklassifizierten Bild ausblenden. Zu beachten ist, dass die Hautfarbklassifikation und die anschließende

Schwellenoperation auf das Eingangsbild in seiner originalen Auflösung angewendet werden, während die Umsetzung der Heuristiken (Form und Größe der Region) auf allen Ebenen der Auflösungspyramide erfolgt.



Abbildung 8.6: Ablauf der Hautfarbsegmentierung, dargestellt für eine Ebene der Auflösungs-
pyramide (aus [Hem00]).

Weitere Ergebnisse zur Hautfarbdetektion können den Abbildungen 8.14 bis 8.16 in Abschnitt 8.2.6 entnommen werden.

8.2.3 Kopfschulterpartiedetektion

Konturen können für die visuelle Szenenanalyse ein wesentliches Merkmal von Objekten darstellen. Sie bilden Umrisse von Silhouetten, die durch die Projektion dreidimensionaler Objekte auf die zweidimensionale Bildebene entstehen.

So stellt auch der Konturverlauf des Kopf- und Schulterbereiches von Personen ein charakteristisches Merkmal dar. Die relativ hohe Invarianz gegenüber Veränderungen der Körperposition, z.B. im Vergleich zu den Extremitäten, prädestiniert diesen Konturbereich für die Personenverifikation (siehe auch [FZ98]).

8.2.3.1 Gewinnung eines Konturmodells

Um ein geeignetes Konturmodell zu erstellen, wurde zunächst aus einem Satz von Beispielbildern der mittlere Konturverlauf von in etwa frontal zugewandten Personen ermittelt (Abbildung

8.7). Dabei wurde jede der 10 im Datensatz enthaltenen Personen unter einheitlichen Beleuchtungsbedingungen vor homogenem Hintergrund jeweils frontal sowie um $+10^\circ$ bzw. -10° davon abweichend aufgenommen. Um ein symmetrisches Konturmodell zu erhalten, wurde jedes Bild vertikal gespiegelt, so dass insgesamt 60 Beispielbilder in Grauwertform zur Verfügung standen, die dann von der Ursprungsdimension (256×256) tiefpassgefiltert und auf eine Größe von 16×16 unterabgetastet wurden. Anschließend wurden mittels Sobel-Operator die Kanten dieser Bilder hervorgehoben und danach über alle Bilder gemittelt. Mit einer einfachen Schwellenoperation erfolgte eine Binarisierung zur Bestimmung des repräsentativen Konturverlaufs. Damit liegt der interessierende Konturverlauf in Form einer 16×16 Binärmatrix vor. Noch offen ist die lokale Orientierung der einzelnen Konturabschnitte. Diese wird bestimmt mittels orientierungsspezifischer Analyse dieser Binärmatrix, deren Resultat das eigentliche Konturmodell Λ darstellt. Ein Element $\lambda_{i,j}$ der Matrix Λ kodiert jetzt die in der Nachbarschaft dieses Elements vorherrschende Orientierung (siehe auch Abbildung 8.7 rechts).

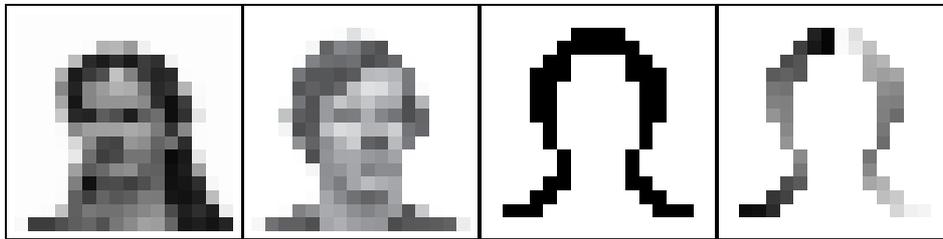


Abbildung 8.7: v.l.n.r.: zwei Beispielbilder (16×16) des Trainingsdatensatzes; daraus ermittelter binärer Konturverlauf; eigentliches Konturmodell (Prototyp) als Arrangement der dominanten Orientierungen entlang des binären Konturverlaufs, wobei die Orientierungswinkel als Grauwerte kodiert wurden (0° : schwarz; 90° : mittelgrau; 180° : weiß); das Konturmodell ist symmetrisch, da der Datensatz entlang der y-Achse gespiegelt wurde (nach [Bra01]).

Die Ermittlung der lokal dominanten Orientierung eines Bildausschnitts erfolgt wiederum durch das bereits zur Odometrikorrektur eingesetzte und sehr effiziente Verfahren nach [Jae97], das beispielsweise auch in [Bra01] sehr ausführlich erörtert wird.

8.2.3.2 Eigentliche Konturdetektion

Nachdem auch für das aktuelle Eingangsbild die lokal dominante Orientierung für jeden Bildpunkt mittels des o.g. Verfahrens berechnet wurde, erfolgt nun auf der Basis des Konturmodells Λ die Definition eines Ähnlichkeitsmaßes $E_{x,y}$, das das Vorhandensein einer Kopf-Schulter-Partie an der entsprechenden Bildposition beschreibt (Gleichung 8.15).

$$E_{x,y} = \frac{1}{2} \sum_{j=-\frac{d_i}{2}}^{\frac{d_i}{2}} \sum_{i=-\frac{d_i}{2}}^{\frac{d_i}{2}} [\cos(2 * \|\lambda_{i+\frac{d_i}{2},j+\frac{d_i}{2}} - OT_{x+i,y+j}\|) + 1] \quad (8.15)$$

Wegen der π -Periodizität des Ergebnisses der orientierungsspezifischen Analyse einerseits, und um andererseits geeignet die Ähnlichkeit zweier Elemente von Λ (Konturmodell) und OT (aus dem Eingangsbild ermittelte lokal dominante Orientierung) zu bestimmen, benötigt die Funktion 8.15 dieselbe Periodizität.

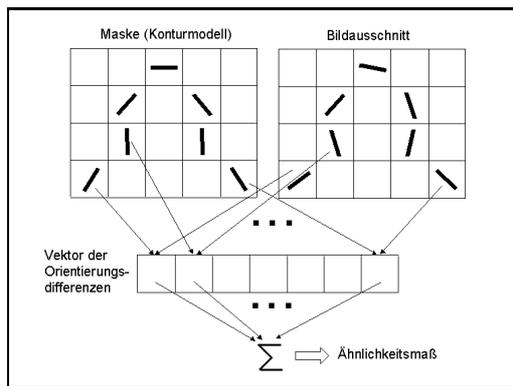


Abbildung 8.8: Grundprinzip der modellbasierten Konturdetektion.

Abbildung 8.8 verdeutlicht das Grundprinzip, das der Detektion einer Kopf-Schulter-Partie zugrunde liegt. Für den gerade betrachteten Bildausschnitt wird entlang der Modellkontur die erwartete Orientierung $\lambda_{i,j}$ mit der im Bild berechneten lokal dominanten Orientierung $OT_{x,y}$ verglichen, was zu einem Vektor der Orientierungsdifferenzen führt, dessen Dimension durch die Anzahl der betrachteten Konturabschnitte bestimmt ist. Überschreitet die Differenz zwischen tatsächlicher und erwarteter Orientierung einen festgelegten Schwellwert, so wird der entsprechende Konturabschnitt von der weiteren Betrachtung ausgeschlossen. Ein Ähnlichkeitsmaß wird nur dann berechnet, wenn eine festgelegte Mindestanzahl an Konturabschnitten in die Berechnung eingehen, wobei eine Normierung auf die Anzahl der beitragenden Konturabschnitte erfolgt.

Das beschriebene Verfahren liefert im Falle eines wenig strukturierten oder homogenen Bildhintergrundes sehr gute Ergebnisse. Bei der Anwendung in der realen Einsatzumgebung zeigte sich jedoch, dass hier vorwiegend mit stark strukturiertem Hintergrund zu rechnen ist. Dies führt in vielen Fällen zu dem Effekt, dass die gesuchte Kontur zwar im Bild vorhanden ist, jedoch sehr stark von Strukturen des umgebenden Hintergrundes überlagert wird. Im Resultat werden die gesuchten Konturen nicht detektiert, und es kommt zu einer Vielzahl von falsch-positiven Detektionsergebnissen. Um diesem Effekt Rechnung zu tragen, wurde der Detektionsmechanismus zu einem zweistufigen Verfahren erweitert. Die erste Stufe umfasst eine Vorselektion, die nach dem oben beschriebenen Verfahren funktioniert. Dabei wird die Schwelle für die maximal zulässige Orientierungsdifferenz pro Konturabschnitt relativ groß gewählt, um sicherzustellen, dass die gesuchte Kontur in der Menge der somit erzeugten Hypothesen enthalten ist (hohe Sensitivität). In der anschließenden zweiten Stufe werden diese Hypothesen einer detaillierteren Analyse unterzogen, deren Grundlage die Berechnung weiterer dominanter Orientierungen pro Bildpunkt bildet. Um weitere dominante Orientierungen pro Bildpunkt zu berechnen, nutzt man die Tatsache aus, dass jeweils vor der eigentlichen Orientierungsrechnung ein Tiefpass eingesetzt wird, der die Dimension der Bildregion bestimmt, die in die Orientierungsberechnung einfließt. Durch Variation dieses Tiefpasses können nun verschiedene dominante Orientierungen pro Bildpunkt in Form eines Merkmalsvektors zur Verfügung gestellt werden, dessen Dimension durch die Zahl der verwendeten Tiefpassfilter determiniert ist. Für die detaillierte Analyse der in der ersten Verarbeitungsstufe generierten Konturhypothesen erfolgt nun ein nochmaliger Vergleich des Konturmodells mit den berechneten verschiedenen dominanten Orientierungen pro Konturabschnitt. Dabei wird jetzt die Schwelle für die maximal zulässige Orientierungsdifferenz pro Konturabschnitt relativ klein gewählt, um möglichst nur die gesuchten Konturen zu detektieren (hohe Spezifität). Eine abschließende Schwellen-

operation, die auf den berechneten Ähnlichkeitsmaßen für jeden Bildpunkt angewendet wird, gibt an, ob die gesuchte Kontur an der entsprechenden Position vorliegt oder nicht.

Die zweistufige Auslegung des Verfahrens erweist sich insbesondere aus Effizienzgründen als sehr sinnvoll, da eine alleinige detaillierte Analyse für jeden Bildausschnitt zu viel Rechenzeit in Anspruch nehmen würde. Abbildung 8.9 verdeutlicht den Effekt des skizzierten zweistufigen Verfahrens zur Konturdetektion.



Abbildung 8.9: Illustration des zweistufigen Konturdetektionsverfahrens anhand einer Laborumgebung (links) und einer Szene aus der realen BauMarkt-Umgebung (rechts). Zu sehen sind wiederum die verwendeten fünf Auflösungsebenen, die hier einen Entfernungsbereich von 0.5 bis etwa 3m, bezogen auf die Bildebene der Frontalkamera, abdecken.

Weitere Ergebnisse zur Kopf-Schulter-Partie-Detektion sind in den Abbildungen 8.14 bis 8.16 in Abschnitt 8.2.6 dargestellt.

8.2.4 Gesichtsdetektion

Für die Gesichtsdetektion wurden in den letzten Jahren verschiedene Verfahren publiziert, die z.B. die Nutzung von so genannten Eigenfaces [TP91], merkmalsbasierte Methoden [YC97] [FZ99b] sowie den Einsatz neuronaler Netzwerke [RBK98] umfassen.

Der Vorteil der neuronalen Verfahren liegt in der Tatsache begründet, dass der das Gesicht enthaltende Bildausschnitt direkt als Verteilung von Pixelintensitäten aufgefasst werden kann, und dass mittels eines Satzes an möglichst repräsentativen Beispielen die Parameter des Klassifika-

tors über einen Trainingsprozess entsprechend adaptiert werden können. Während Beispiele für Gesichter relativ leicht generiert werden können, besteht ein Problem in der Charakterisierung typischer Kein-Gesicht-Bilder, die für das Training des Klassifikators benötigt werden. Man kann dieses Problem umgehen, indem man, wie z.B. in [RBK98] vorgeschlagen, einen Bootstrap-Algorithmus einsetzt, der während des Trainings falsch-positive Detektionsergebnisse automatisch der Menge der Negativbeispiele hinzufügt.

Das Modul zur Gesichtsdetektion wurde im konkret vorliegenden Fall als Cascade-Correlation-Netzwerk (CCNW, [FL90]) implementiert. Der Grund für den Einsatz dieses speziellen Netzwerk-Paradigmas liegt in seiner Fähigkeit, eine optimal an die Komplexität des Abbildungsproblems angepasste Netzwerktopologie zu erzeugen. Im Gegensatz zum Multilayer-Perceptron, bei dem die Topologie vorab festgelegt bzw. experimentell ermittelt werden muss, passt das CCNW während des Trainingsprozesses sowohl seine Parameter als auch seine Topologie an das vorliegende Abbildungsproblem an. Beginnend mit einem minimalen Netzwerk (direkte lineare Eingabe-Ausgabe-Abbildung) werden während des Trainings so lange neue Hidden-Neuronen eingefügt, bis ein vorgegebenes Gütekriterium erfüllt ist.

Für das Training des Netzwerks wurden 174 Bilder eines öffentlich zugänglichen Datensatzes der AT&T Laboratories Cambridge (<http://www.cam-orl.co.uk/facetatabase.html>) verwendet. Aus diesen Bildern wurden Gesichtsregionen der Dimension 15×20 Pixel manuell ausgeschnitten, die dann als Positivbeispiele fungierten. Als Negativbeispiele dienten zunächst zufällig ausgewählte Bildausschnitte, die garantiert kein Gesicht enthalten, ehe während des fortschreitenden Trainingsprozesses dann mittels des Bootstrap-Algorithmus diese Menge sukzessive erweitert wurde. Aus Effizienzgründen wurde eine möglichst minimale Netzwerktopologie angestrebt. Um dies zu erreichen, wurde eine Vielzahl von Netzwerken mit unterschiedlichen Initialisierungsbedingungen trainiert. Abbildung 8.10 verdeutlicht das Grundprinzip der CCNW-basierten Gesichtsdetektion und zeigt die Topologie des aktuell verwendeten Netzwerks.

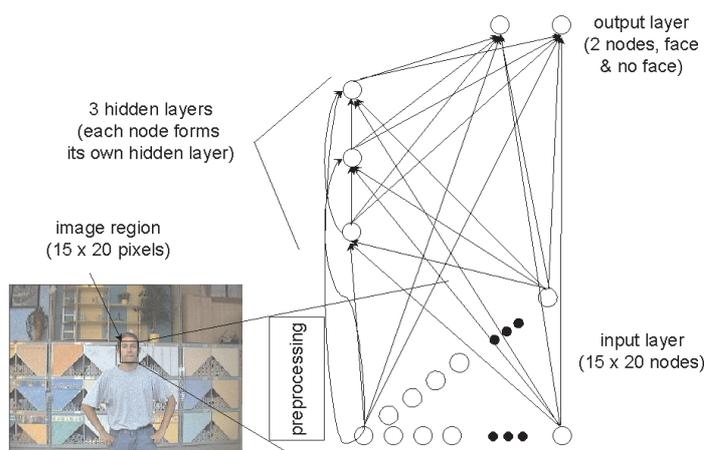


Abbildung 8.10: Neuronal basierte Gesichtsdetektion mittels Cascade-Correlation-Netzwerk (aus [BWK⁺ 01a]).

Einige exemplarische Detektionsergebnisse stellt Abbildung 8.11 dar. Weitere Ergebnisse zur CCNW-basierten Gesichtsdetektion finden sich in den Abbildungen 8.14 bis 8.16 in Ab-

schnitt 8.2.6.



Abbildung 8.11: Exemplarische Ergebnisse des CCNW-basierten Gesichtsdetektors. V.l.n.r.: zwei Bilder, die sowohl korrekte als auch falsch-positive Detektionen enthalten; ein Bild mit ausschließlich korrekter Detektion und ein Beispielbild, das nur falsch-positive Detektionen aufweist.

Im Ergebnis der Gesichtsdetektion liegt ein visueller Cue vor, dessen Vorteil in der hohen Spezifität zu sehen ist. Den entscheidenden Nachteil stellt die hohe Berechnungskomplexität im Vergleich zu den beiden anderen Cues Hautfarbe und Kopf-Schulter-Kontur dar. Um dennoch eine hinreichende Performanz des Gesamtsystems zur Personenverifikation zu erreichen, wurde eine Vorselektion der mit dem Gesichtsdetektor zu untersuchenden Regionen durch die beiden anderen Cue-Module eingeführt. Diese Tatsache wurde bei der Fusion der Einzelcues, die in Abschnitt 8.2.5.2 beschrieben wird, explizit berücksichtigt. Mit der genannten Vorselektion konnte der Berechnungsaufwand für die Gesichtsdetektion etwa um den Faktor 10 gesenkt werden.

8.2.5 Fusion und Selektion

8.2.5.1 Kodierung der Einzelcues

Im Ergebnis der in den Abschnitten 8.2.2 bis 8.2.4 beschriebenen Teilmodule zur Personenverifikation liegen jetzt, wie aus Abbildung 8.3 ersichtlich wird, drei Merkmalspyramiden vor. Um zu erreichen, dass jeder Cue gleichberechtigt zum Verifikationsresultat beitragen kann, erfolgt für alle Cues eine einheitliche Kodierung mittels gaußförmiger Aktivitätsblobs identischer Größe, die an jeder 3D-Position, an der ein Cue ein signifikantes Detektionsergebnis liefert, in die entsprechende Merkmalspyramide eingetragen werden.

8.2.5.2 Fusion der Einzelcues

Bei der Fusion der Einzelcues sollte der Tatsache Rechnung getragen werden, dass durch deren parallele Nutzung möglicherweise auftretende Probleme bei der Detektion eines Cues durch einen anderen Cue kompensiert werden können. Aus diesem Grund wurde von Beginn an keine pipelineartige Struktur gewählt, die die sichere Detektion eines Einzelcues voraussetzen und damit die Funktion des Gesamtsystems Personenverifikation von einem einzigen Cue abhängig machen würde.

Die verschiedenen Möglichkeiten zur Fusion der Einzelcues reichen von einer einfachen linearen Superposition über polynomiale Ansätze höherer Ordnung und konnektionistische Verfahren bis hin zu fuzzy-basierten Methoden. Implementiert wurde zunächst ein Verfahren aus der

Fuzzy-Logik, der so genannte Minimum-Maximum-Operator (Min-Max-Operator, siehe auch [Bra01] [Zim91]). Bezeichnen die Terme C_H , C_K und C_G die Beiträge der Hautfarb-, Kontur- und Gesichtsdetektion sowie C_F das Fusionsergebnis, so lautet die entsprechende Berechnungsvorschrift übertragen auf die vorliegende Problemstellung

$$C_F(C_H, C_K, C_G) = \gamma \cdot \min(C_H, C_K, C_G) + (1 - \gamma) \cdot \max(C_H, C_K, C_G) \quad (8.16)$$

mit $\gamma \in [0, 1]$, wobei hier die Indizes zur Bezeichnung der Position innerhalb der entstehenden Fusions- bzw. Selektionspyramide (Abbildung 8.3) weggelassen wurden. Die parametrische Zusammensetzung des Operators aus Operatoren, die die logischen UND- bzw. ODER-Funktionen modellieren, sichert, dass einerseits keine einfache Mittelung vorgenommen wird und andererseits auch Koinzidenzen zwischen den einzelnen Cues in das Fusionsergebnis einbezogen werden. Dies stellt sicher, dass sowohl die Präsenz aller Einzelcues als auch die optimale Präsenz nur eines Cues nicht zwingend erforderlich ist. Somit wird eine gute Artefaktunterdrückung auf einfache Weise möglich: Koinzidenzen sind notwendig, aber nur bis zu einem gewissen Grade - einzelne starke Beiträge sind ebenfalls notwendig, aber nicht notwendigerweise ausreichend für ein starkes Fusionsergebnis (siehe auch [Bra01]).

Das bislang beschriebene Verfahren zur Fusion der Einzelcues setzt voraus, dass alle Cues auf der gesamten Bilddimension berechnet werden. Da jedoch die Gesichtsdetektion aus Performanzgründen erst nach einer Vorselektion der in Frage kommenden Regionen durch die beiden anderen Cue-Module zum Einsatz kommt, wurde eine alternative Methode zur Fusion der Einzelcues in Form einer einfachen Fusionsregel implementiert. Diese besagt, dass eine Bildposition (3D innerhalb der Auflösungs- bzw. Selektionspyramide) nur dann an der finalen Selektion teilnehmen darf, wenn mindestens zwei der drei Einzelcues an dieser Position ein signifikantes Detektionsergebnis liefern. Damit erübrigt sich die Gesichtsdetektion für das gesamte Eingangsbild, da das alleinige Vorhandensein eines Gesichtsdetektionsergebnisses ohne die Unterstützung mindestens eines der beiden anderen Cue-Module nicht ausreichend ist. In Abbildung 8.12 ist das Ergebnis der Anwendung dieser einfachen Fusionsregel, die sich im Vergleich zum ursprünglich eingesetzten Fuzzy-Operator als ähnlich praktikabel erwiesen hat, anhand einer Beispielszene aus der realen Einsatzumgebung dargestellt.

Das Fusionsergebnis wiederum dient als Eingang für die im nachfolgenden Abschnitt beschriebene Auflösungs- bzw. Selektionspyramide aus dynamischen Neuronen, in der eine finale Selektions- bzw. Verifikationsentscheidung vorgenommen wird.

8.2.5.3 Dynamische Selektion

Prämissen Mit dem Entwurf eines Verfahrens zur finalen Selektion wurden die folgenden Ziele verfolgt:

- Die Selektion soll umso sicherer erfolgen, je mehr Beiträge aus dem Fusionsergebnis innerhalb einer Auflösungsebene als auch an korrespondierenden Positionen in unmittelbar benachbarten Auflösungsebenen lokal begrenzt vorliegen.
- Die Selektion soll berücksichtigen, dass die lokalen Analyseergebnisse der einfließenden Einzelcues hinsichtlich ihrer Position nicht notwendigerweise exakt korrespondieren.

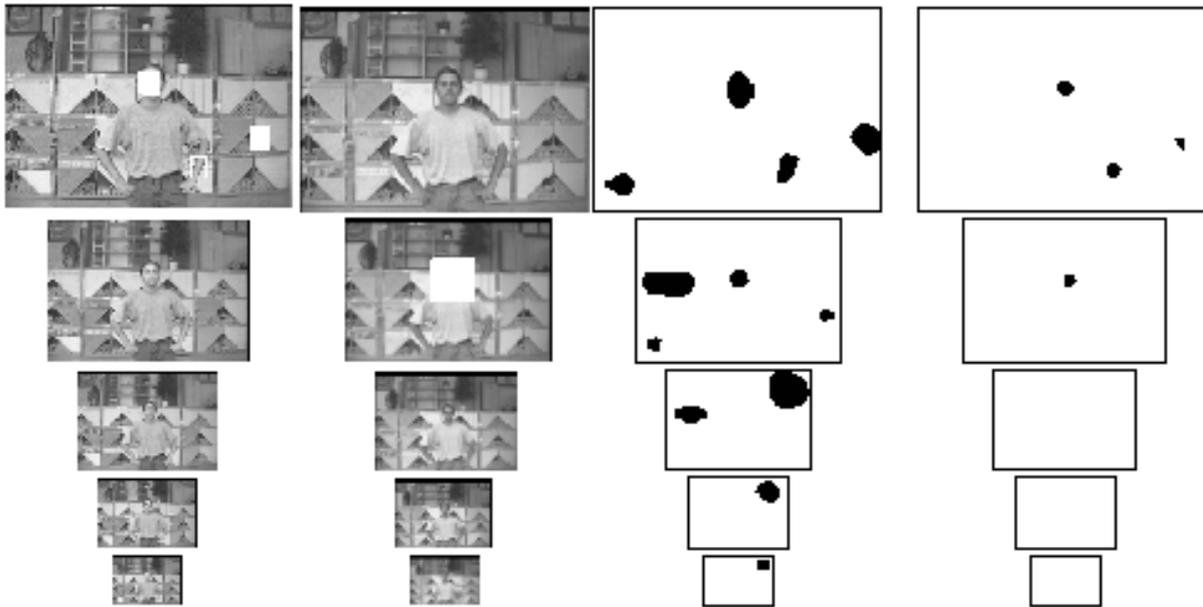


Abbildung 8.12: Beispiel für die Anwendung der einfachen Fusionsregel für die Integration der Einzelcues zur Personenverifikation. Nur die Positionen, an denen mindestens zwei der drei Cue-Module ein signifikantes Ergebnis liefern, dürfen am finalen Selektionsprozess teilnehmen. V.l.n.r.: Gesichtsdetektion, Kopf-Schulter-Partie-Detektion, Hautfarbdetektion und Ergebnis der Fusion.

- Angestrebt wurde weiterhin eine zeitliche Stabilisierung des Selektionsergebnisses, da aufeinander folgende Selektionsergebnisse typischerweise an korrespondierenden Positionen in der Selektionspyramide erscheinen, eine hinreichende Effizienz der vorgelagerten Verarbeitungsstufen vorausgesetzt. Durch einen solchen inherenten Trägheitseffekt können nahezu zwangsläufig auftretende Störungen, die fast ausschließlich kurzzeitige fehlerhafte Detektionen umfassen, die nur singular auf einer Auflösungsstufe auftauchen, wirkungsvoll unterdrückt werden.

Insbesondere der letztgenannte Aspekt ist funktionell sehr wesentlich, da die darin enthaltene Annahme über die Stetigkeit der Umwelt maßgeblich zur Robustheit des Gesamtverfahrens zur Personenverifikation beiträgt. Allerdings muss an dieser Stelle einschränkend bemerkt werden, dass dieser Vorteil erst bei ausreichender Performanz der vorgelagerten Verarbeitungsstufen voll zum Tragen kommt, was zum gegenwärtigen Zeitpunkt noch nicht hinreichend gegeben ist (siehe auch die abschließende Diskussion zu den Verifikationsergebnissen).

Für die Auswahl der Position, an der mit größter Wahrscheinlichkeit eine dem Roboter in etwa frontal zugewandte Person zu finden ist, kommt ein so genannter Winner-Take-All-Prozess (WTA-Prozess) zum Einsatz. Dazu wurde die Auflösungspyramide, in der der Selektionsprozess stattfindet, als dreidimensionales Feld dynamischer Neuronen ausgelegt, wobei der WTA-Prozess über die feldinternen Wechselwirkungen der dynamischen Neuronen realisiert ist.

Konkrete Umsetzung Die erforderliche Topologie des dynamischen Feldes verdient zunächst besondere Beachtung, da mit ihr der Multiskalen-Repräsentation als Basisstruktur zur Personenverifikation Rechnung getragen werden muss. Erst damit ist es möglich, die Verifikation auch hinsichtlich verschiedener Distanzen vorzunehmen.

Für die Lösung der Verifikationsaufgabe erweist es sich jedoch als zweckmäßig, die pyramidale Repräsentation übergangsweise zu verlassen (siehe Abbildung 8.13), um die Korrespondenzen zwischen den Auflösungsebenen der Repräsentation auf ein orthogonales Raster abzubilden. Der Grund ist, dass das wegen der dreidimensionalen Topologie der Daten erforderliche dreidimensionale Feld viel einfacher als kubisches Feld ausgeführt werden kann. Aus der Literatur ist eine solche formale und praktische kubische Erweiterung eines Ansatzes mit dynamischen neuronalen Feldern nicht bekannt. Nach dieser kubischen Umkodierung liegen entsprechend der Pyramidenabstufung fünf Ebenen vor (Abbildung 8.13), mit denen ein dreidimensionales Feld aufgespannt wird, innerhalb dessen die Selektion abläuft. Als Kompromiss für eine einheitliche Auflösung dieser fünf Ebenen würde sich die Auflösung der mittleren Pyramidenebene anbieten, wie dies in Abbildung 8.13 auch zu sehen ist. Jedoch wurde im weiteren zugunsten einer höheren Abarbeitungsgeschwindigkeit die Auflösung der obersten (größten) Pyramidenebene von Hautfarb- und Konturcues übernommen, so dass das kubische Feld eine Dimension von $27 \times 20 \times 5$ dynamischen Neuronen aufweist.

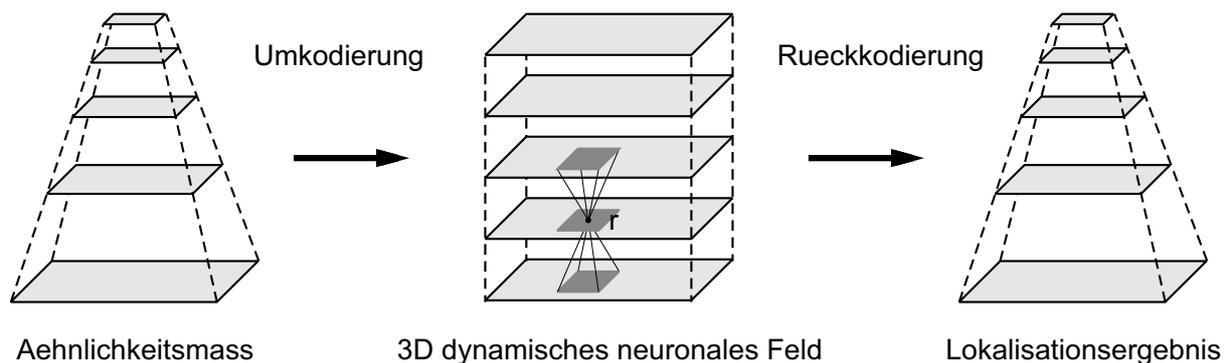


Abbildung 8.13: Temporäre Umkodierung der pyramidalen in eine kubische Repräsentation. Dieser Schritt ist erforderlich, um Korrespondenzen zwischen den Pyramidenebenen für die algorithmische Umsetzung des dreidimensionalen dynamischen neuronalen Feldes zu vereinfachen. Die im Halboktavabstand angeordneten Ebenen erscheinen hier gemäß dem Logarithmus ihrer Kantenlängen (oder auch Abstände) gleichabständig. Alle Ebenen liegen im Kubus mit der Auflösung der mittleren Pyramidenebene vor (der Kubus ist aus Anschauungsgründen vergrößert dargestellt). Nach erfolgter Relaxation des Feldes schließt sich implizit eine Rückkodierung an, so dass der Bezug zwischen Kubusebene und Pyramidenebene hergestellt ist und damit eine Entfernungsschätzung erfolgen kann. Die Skizze zeigt schematisch im Inneren des Kubus' den Umgebungsbereich N , aus dem ein Element an der Stelle $\mathbf{r} = (x_1, x_2, x_3)^T$ direkt aktiviert werden kann (siehe Gleichungen 8.17 ff.).

Das verwendete dynamische neuronale Feld F vom AMARI-Typ besteht aus gegenseitig wechselwirkenden formalen dynamischen Neuronen und kann als rückgekoppeltes nichtlineares dynamisches System beschrieben werden. Ziel des darin ablaufenden WTA-Prozesses ist

die Ausbildung eines *einzelnen* lokalen Blobs aktiver Neuronen, die sich bei Annäherung an einen Gleichgewichtszustand gegenüber den übrigen behaupten können. Die folgende DGL beschreibt die nichtlineare Dynamik des Aktivierungszustandes $z(\mathbf{r}, t)$ eines Neurons (Elements) in kontinuierlicher Notation und konkretisiert die prinzipielle Darstellung des DGL-Systems. Die darin enthaltenen Koeffizienten c_h , c_l und c_i sind Parameter des Systems, anhand derer die externen und internen Erregungen (s. u.), die die Störterme in der DGL bilden, gewichtet werden.

$$\tau \frac{d}{dt} z(\mathbf{r}, t) = -z(\mathbf{r}, t) - c_h h(t) + c_i x(\mathbf{r}, t') + c_l \int_{N \subseteq \mathbb{R}^3} w(\mathbf{r} - \mathbf{r}') y(z(\mathbf{r}', t)) d^3 \mathbf{r}' \quad (8.17)$$

Hierbei steht \mathbf{r} für die dreidimensionale Koordinate einer Neuronenposition im Feld. $z(\mathbf{r}, t)$ ist die Aktivierung des Neurons \mathbf{r} zum diskreten Zeitpunkt t . Korrespondierend dazu symbolisiert $y(z(\mathbf{r}, t))$ die Ausgabeaktivität dieses Neurons (Wertebereich $[0 \dots 1]$). Diese wird mittels der logistischen Sigmoidfunktion (Fermifunktion) bestimmt:

$$y(z(\mathbf{r}, t)) = \frac{1}{1 + \exp(-s \cdot z(\mathbf{r}, t))} \quad (8.18)$$

Hier liegt die Ursache für die Nichtlinearität der DGL in Gl. 8.17. Bezüglich des Neuronenmodells von AMARI stellt $y(z(\mathbf{r}, t))$ ein Maß für die Impulsemissionsrate an Feldkoordinate \mathbf{r} zur Zeit t dar.

$x(\mathbf{r}, t')$ bezeichnet den lokalen externen Input, der den umkodierten Beiträgen der einzelnen Cues (siehe Abschnitt 8.2.5.1) entspricht. Durch die Notation t' wird ausgedrückt, dass dieser Stimulus ab der Startzeit t' für einen Iterationsvorgang bis zum Ende des Relaxationsprozesses konstant bleibt.

$h(t)$ ist eine feldglobale Inhibition zum Zeitpunkt t , die über alle Neuronenaktivitäten $y(t - \Delta T)$ im gesamten Feld $F \subseteq \mathbb{R}^3$ aufsummiert wird:

$$h(t) = \int_{F \subseteq \mathbb{R}^3} y(\mathbf{r}', t - \Delta T) d^3 \mathbf{r}' \quad (8.19)$$

Die laterale Gewichtungsfunktion $w(\mathbf{r} - \mathbf{r}')$ ist vom Mexikanerhut-Typ und wurde folgendermaßen realisiert:

$$w(\mathbf{r} - \mathbf{r}') = 2 \exp(-(\mathbf{r} - \mathbf{r}')^\top \Sigma_1^{-1} (\mathbf{r} - \mathbf{r}')) - \exp(-(\mathbf{r} - \mathbf{r}')^\top \Sigma_2^{-1} (\mathbf{r} - \mathbf{r}')) \quad (8.20)$$

mit

$$\Sigma_2^{-1} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (8.21)$$

$$\Sigma_1^{-1} = \left(\frac{2}{3} \Sigma_2 \right)^{-1} = \begin{pmatrix} \frac{4}{3} & 0 & 0 \\ 0 & \frac{4}{3} & 0 \\ 0 & 0 & \frac{2}{3} \end{pmatrix}^{-1} = \begin{pmatrix} \frac{3}{4} & 0 & 0 \\ 0 & \frac{3}{4} & 0 \\ 0 & 0 & \frac{3}{2} \end{pmatrix} \quad (8.22)$$

Sie gewichtet als Modell eines rezeptiven Feldes die laterale Aktivierung eines Neurons r aus seiner Nachbarschaft $N \subseteq \mathbb{R}^3$, die in die DGL (siehe Gl. 8.17) eingeht. Für ein ausgewähltes r ist N in Abb. 8.13 dunkelgrau markiert. Die gewählte diskrete Ausdehnung von N beträgt $5 \times 5 \times 3$ Matrixelemente, d. h. es werden gerade drei umkodierte Ebenen umfasst. Diese laterale Aktivierung erfolgt, wie die Parametrisierung von Σ_1 und Σ_2 und auch die diskrete Ausdehnung zeigen, nicht isotrop. Das hat seine Ursache darin, dass der dreidimensionale Gitterabstand des Feldes nicht isotrop sein kann, da die Information über räumliche Tiefe, die über die fünf Ebenen kodiert wird, in einem erheblich größeren Raster vorliegt, als in den dazu orthogonal angeordneten und durchweg viel feiner aufgelösten, zu den Bildmatrizen korrespondierenden Matrizen (Ebenen). Dieser Umstand fand in der anisotropen Parametrisierung der entsprechenden Verkopplungsmatrizen eine grobe Berücksichtigung. Qualitativ vorgegeben ist, zentral eine positive Gewichtung zu verwenden, die nach außen hin in eine negative Gewichtung übergeht. Der Mexikanerhut-Typ ist keine ausdrückliche Vorgabe AMARIS. Hervorzuheben ist weiterhin, dass diese Funktion hier nicht von absoluten Feldkoordinaten abhängt (jedes Neuron hat das gleiche rezeptive Feld) und dass sie symmetrisch ist, weswegen F ein homogenes und symmetrisches dynamisches neuronales Feld darstellt. Ferner besteht keine zeitliche Veränderlichkeit der Verkopplungsmatrizen. Da sich diese aus einer Differenz von Funktionen vom Gauß-Typ zusammensetzen, besteht die Gelegenheit, implementationsseitig die Faltung mit den einzelnen Gaußfunktionen zunächst jeweils linear separiert zu berechnen und anschließend die Differenz der Faltungsergebnisse zu bilden. Für eine detailliertere Diskussion sei auf [Bra01] verwiesen.

8.2.6 Experimentelle Ergebnisse

Die Abbildungen 8.14 bis 8.16 sollen anhand einiger typischer Beispielszenen die Gesamtfunktionalität der Personenverifikation illustrieren.

Gegenwärtig verfügt das System über eine Gesamtperformanz von ca. 0.5 Hz auf einem Doppel-Pentium III mit 500 MHz Taktfrequenz. Die Dimension des Eingangsbildes beträgt etwa 200×200 Pixel. Die verwendeten fünf Auflösungsebenen decken einen Entfernungsbereich von etwa 0.5 m bis 2.5 m ab. Verwendet wurde in den dargestellten Beispielen eine der beiden Frontalkameras.

Bei der Parametrisierung des Moduls zur Personenverifikation wurde insbesondere darauf Wert gelegt, dass das System hochgradig spezifisch auf das Vorhandensein von Personen reagiert, was die in Abbildung 8.17 zusammengefassten Ergebnisse auch verdeutlichen.

Dies ist notwendig, weil im Fall eines falsch positiven Verifikationsergebnisses der Shopping-Assistent versuchen würde, auch mit unbelebten Gegenständen Kontakt aufzunehmen, was einem Beobachter nicht unbedingt den Eindruck eines intelligenten Verhaltens vermitteln würde. Diese hochgradige Spezifität bewirkt allerdings auch, dass die Verifikation nicht in jedem Fall erfolgreich ist. Da dem an einer Interaktion mit dem Roboter interessierten BauMarkt-Kunden jedoch immer die Möglichkeit bleibt, sich über den Touchscreen direkt „einzubuchen“, sollte diese Vorgehensweise keine starke Einschränkung des Interaktionsverhaltens des Shopping-Assistenten darstellen (siehe auch Abschnitt 9.3).

Das Modul zur Personenverifikation und insbesondere das dynamische Feld realisieren für den Fall, dass eine hinreichend effiziente Verarbeitung über einem kontinuierlichen Datenstrom erfolgen kann, die Funktionalität eines dynamischen Zustandsschätzers, die dem zum Perso-

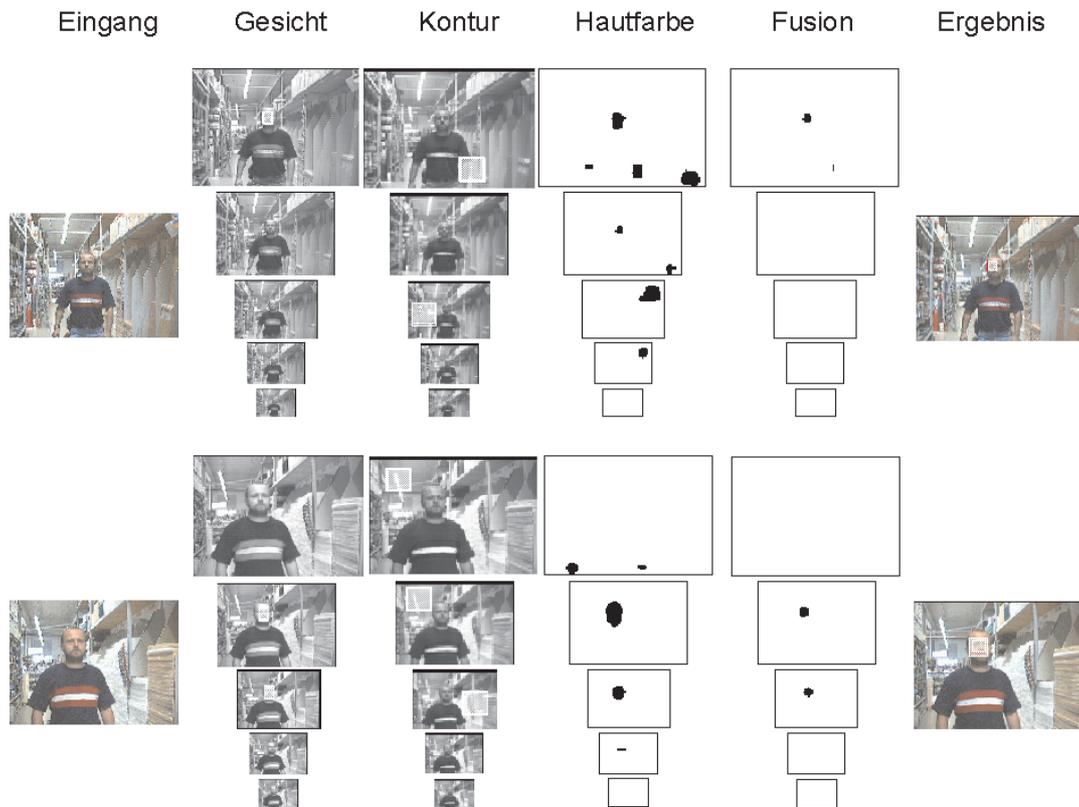


Abbildung 8.14: Verifikationsergebnisse für verschiedene typische Beispielszenen, jeweils in einem Gang zwischen zwei Regalreihen des BauMarktes. Das Eingangsbild ist jeweils links, das Verifikationsergebnis ist rechts dargestellt. Dazwischen (v.l.n.r.) finden sich die Ergebnisse der Cue-Module für Gesicht, Kopf-Schulter-Kontur und Hautfarbe sowie das Ergebnis der Cue-Fusion, jeweils für alle verwendeten 5 Auflösungsebenen. Die korrekte Verifikation basiert in beiden Szenen auf der Korrespondenz zwischen Gesicht- und Hautfarbedetektion.

nentracking bzw. zur visuellen Selbstlokalisierung eingesetzten Condensation-Algorithmus sehr ähnlich ist. Auch hier besteht das Ziel darin, eine a priori unbekannte, möglicherweise multimodale Verteilungsfunktion iterativ zu approximieren, wobei hier zusätzlich die Selektion des Maximums dieser Verteilungsfunktion angestrebt wird. Durch die Definition der Wechselwirkung innerhalb des Feldes (3D-Kubus) wird implizit der stochastische Anteil des Bewegungsmodells implementiert. Die Erregung der Neuronen korrespondiert mit der Bewertung der einzelnen Positionen (Stützstellen im Falle der unvollständigen Abtastung des Zustandsraumes) anhand des Wahrnehmungsmodells. Der Nachteil des neuronalen Verfahrens besteht darin, dass die Verteilungsfunktion über ihrem gesamten Definitionsbereich (alle Ebenen der Auflösungspyramide) kontinuierlich berechnet werden muss, was momentan auch nicht annähernd schritthaltend mit dem einkommenden Datenstrom realisiert werden kann. Im Falle einer schritthaltenden Berechnung könnte man das Modul zur Personenverifikation ebenso gut auch zum Personentracking einsetzen.

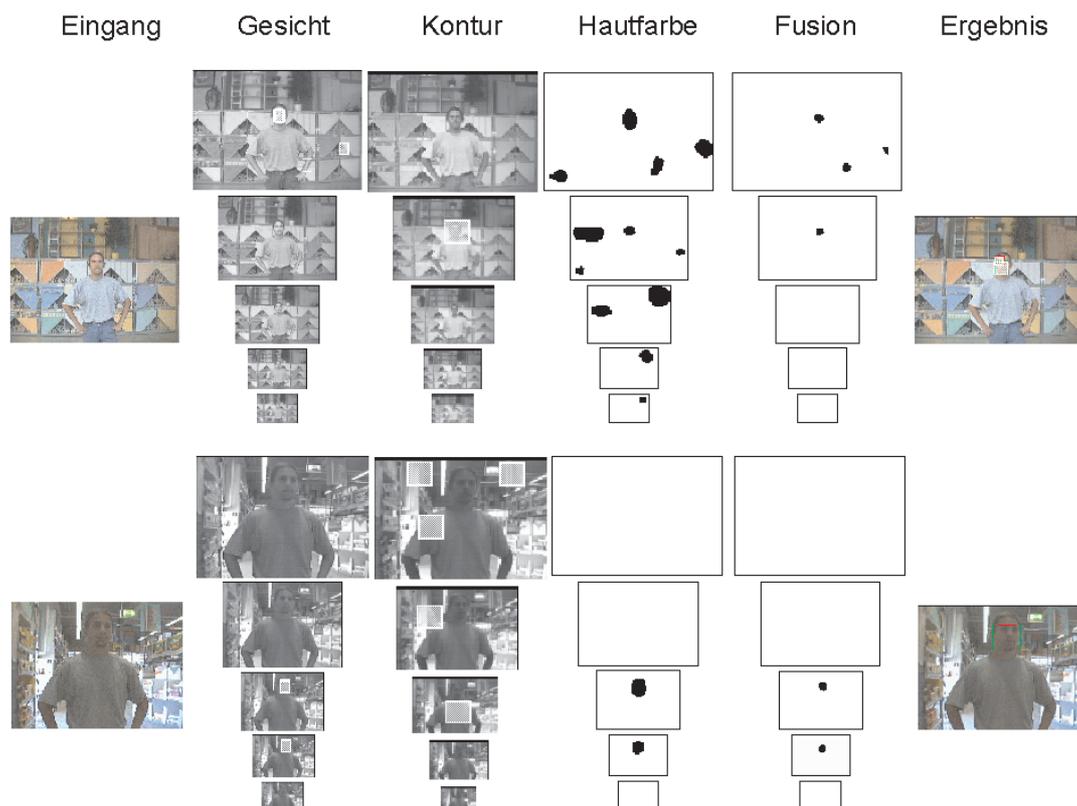


Abbildung 8.15: Verifikationsergebnisse für verschiedene typische Beispielszenen: oben vor einem stark strukturierten Tapetenregal des BauMarktes, unten in einem Freiraumbereich des BauMarktes. Im oberen Beispiel treten sowohl Korrespondenzen zwischen Hautfarbe und Gesicht als auch zwischen Kontur und Gesicht auf, im unteren Beispiel führen die Korrespondenzen zwischen Hautfarbe und Gesichtsstruktur zu einer korrekten Verifikation.

Die Personenverifikation kann sowohl auf den Bildern der Omnikamera als auch der Frontkamera(s) ablaufen. Beim Einsatz der Omnikamera wird der zu analysierende Bildbereich jedoch auf einen Bereich von $\pm 30^\circ$ um den von der Aufmerksamkeitssteuerung gelieferten Winkel eingeschränkt und anschließend von Polarkoordinaten in kartesische Koordinaten transformiert, um auf den Bildern beider Kamerasysteme identische Algorithmen einsetzen zu können.

Das Ergebnis der Personenverifikation wird an zwei nachfolgende Module weitergegeben: zum einen triggert es eine Sprachausgabe, mit dem Ziel, den Kunden gezielt anzusprechen und zur Interaktion zu ermuntern, zum anderen liefert die Personenverifikation eine gute Initialisierung für das im nachfolgenden Abschnitt beschriebene Personentracking.

8.3 Visuelles Personentracking

Für die Interaktion zwischen Shopping-Assistent und Benutzer ist das kontinuierliche Aufrechterhalten des Kontakts von zentraler Bedeutung. Der Roboter sollte also immer wahrnehmen können, ob sich der aktuelle Kunde noch in seinem sensorischen Erfassungsbereich befindet.

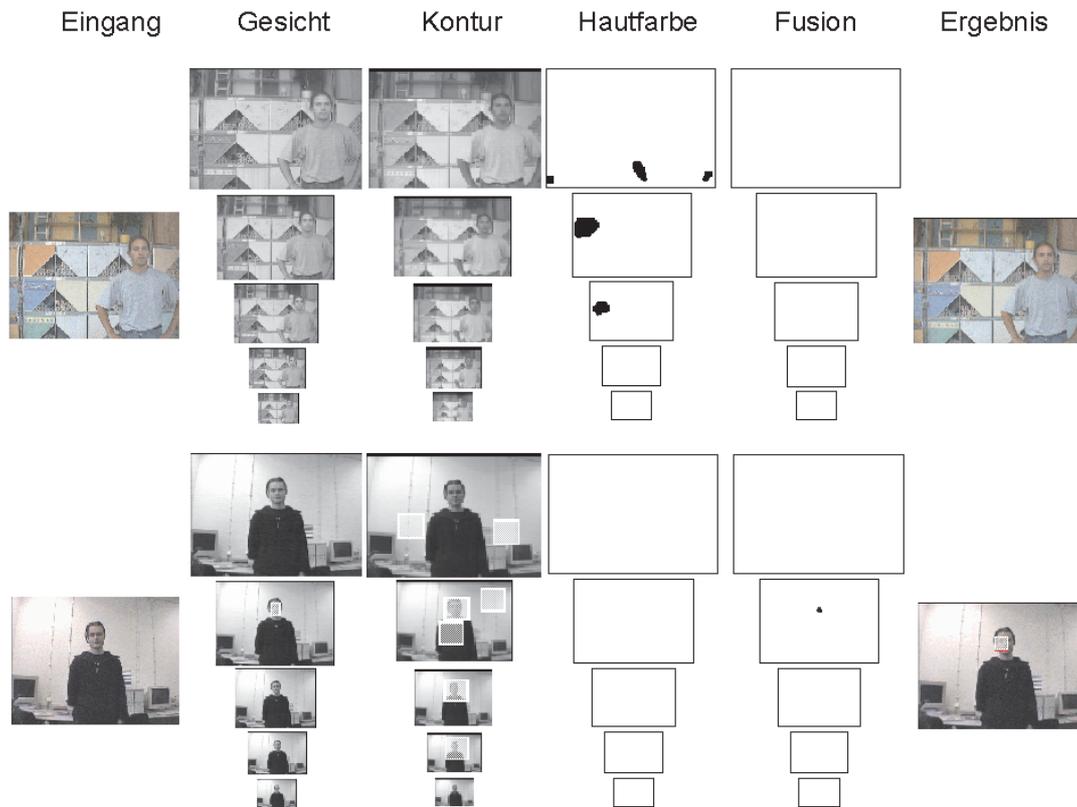


Abbildung 8.16: Verifikationsergebnisse für verschiedene typische Beispielszenen: oben wiederum vor dem Tapetenregal, wobei hier die Personenverifikation kein Ergebnis liefert. Unten dargestellt ist eine Laborszene, in der aufgrund ungünstiger Beleuchtungsbedingungen die Hautfarbanalyse keinen Beitrag zum Verifikationsergebnis liefern kann, aber aufgrund der Kombination der beiden alternativen Cues Kontur und Gesicht trotzdem eine korrekte Verifikation erfolgt.

Andernfalls würde der Shopping-Assistent beispielsweise eine durch den Benutzer einmal gewählte Zielposition stur anfahren, auch wenn der Kunde sich zwischenzeitlich einem anderen Artikel zuwendet und dem Roboter gar nicht mehr folgt. Dies wäre für das angestrebte, möglichst intuitive und natürliche Gesamtverhalten des Shopping-Assistenten wenig wünschenswert. Aus diesem Grund müssen Verfahren implementiert werden, die dem Roboter dieses Kontakthalten ermöglichen. Als Sensormodalitäten kommen prinzipiell sowohl die Abstandssensorik als auch die Omni- bzw. Frontalkamera(s) in Frage, wobei der Schwerpunkt auf visuelle Trackingverfahren gelegt wurde. Zwar stellt dies für die konkreten Umgebungsbedingungen des BauMarktes eine relativ große Herausforderung dar, jedoch können die beispielsweise in [SB01, SBFC01] vorgestellten Verfahren zum Tracking bewegter Objekte mittels Laserscanner nicht unmittelbar auf die auf dem PERSES-Roboter vorhandenen Ultraschallsensoren (Genauigkeit der Entfernungsmessung und Winkelauflösung) übertragen werden. Weiterhin bietet ein visuell-basiertes Verfahren den Vorteil, Informationen über das zu verfolgende Objekt zu gewinnen, die rein entfernungs-basiert nicht extrahierbar sind. So kann beim laser-basierten

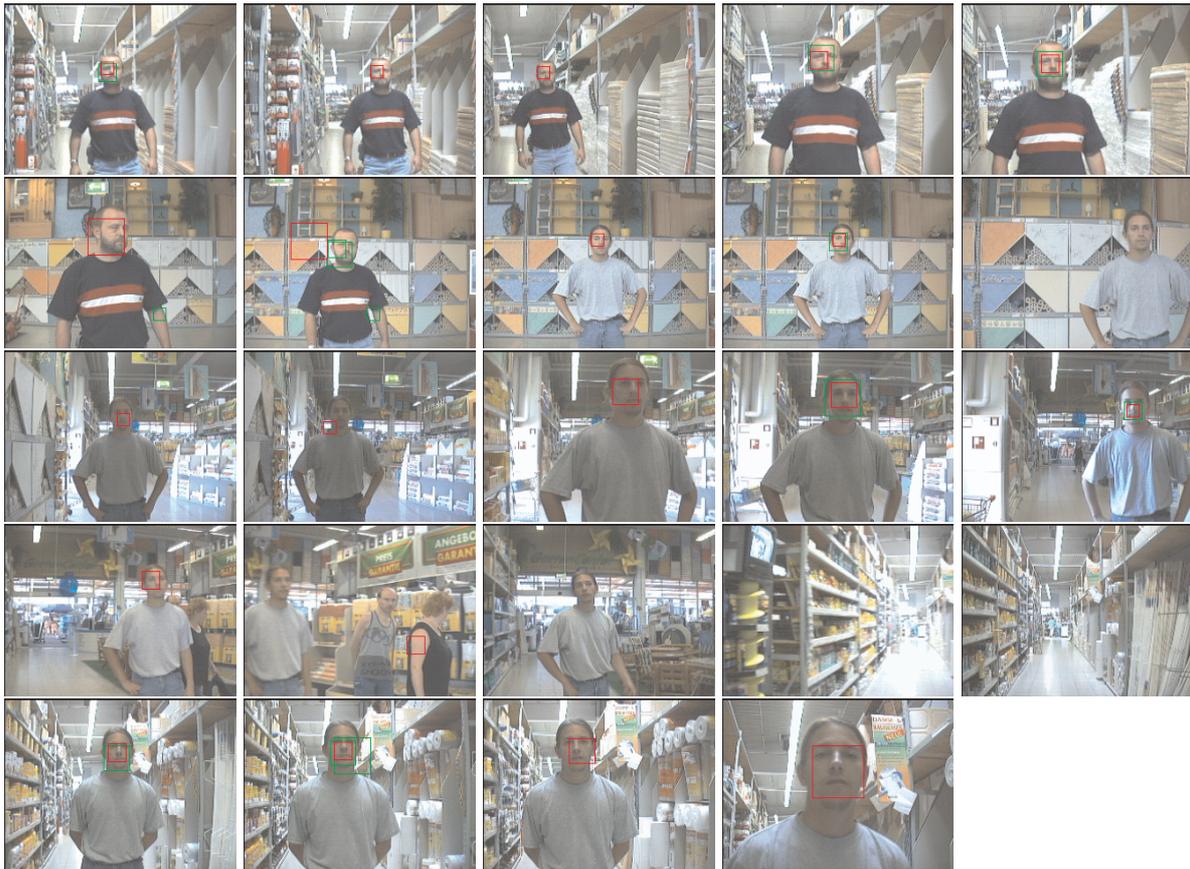


Abbildung 8.17: Ergebnisse der Personenverifikation für verschiedene Situationen im Baumarkt: in Gängen zwischen den Regalreihen sowie vor Regalen, die jeweils schwierige Hintergrundsituationen darstellen. Nochmals hervorgehoben werden soll die hohe Spezifität des Verfahrens, die zwar in einigen Fällen zur Nicht-Detektion einer vorhandenen Person führt, aber auch sichert, dass in Bildern, die keine Person beinhalten, eben auch keine falsch-positiven Detektionen auftreten. Damit soll verhindert werden, dass der Roboter aufgrund fehlerhafter Verifikation auch unbelebte Objekte anspricht. Grüne Rahmen markieren stark ausgeprägte Blobs innerhalb des Selektionsprozesses (siehe Abschnitt 8.2.5.3), rote Rahmen entsprechen schwächer ausgeprägten Blobs, die jedoch im Selektionsprozess nicht unterdrückt werden konnten.

Tracking beispielsweise nicht unterschieden werden, ob sich die Person dem Roboter zu- oder abwendet. Diese Information ist aber möglicherweise entscheidend für ein situationsgerechtes Verhalten des Roboters. Befindet sich die Person an der Grenze des sensorisch erfassbaren Bereiches, so kann für den Fall, dass sich die Person dem Roboter zuwendet, eine Sprachausgabe und ein Warten des Roboters ausreichend sein, um den Kontakt aufrechtzuerhalten. Wendet sich die Person in der gleichen Situation dagegen ab, sollte der Roboter zusätzlich zu einer Sprachausgabe durch aktives Hinterherfahren den drohenden Kontaktverlust vermeiden.

Die nachfolgenden Abschnitte (8.3.1, 8.3.3) beschreiben zunächst den Trackingprozess, der ausschließlich auf der Nutzung visueller Information beruht. Im Abschnitt 8.3.2 werden dann einige Erweiterungen, die das Personentracking robuster gestalten, vorgestellt und diskutiert.

8.3.1 Condensation-Algorithmus

Das eigentliche Tracking-Verfahren basiert wiederum auf dem in [IB98] vorgeschlagenen CONDENSATION-Algorithmus, welcher bereits im Abschnitt zur visuellen Selbstlokalisierung (7.3) erläutert wurde.

Initialer Belief Bei der Berechnung des initialen Belief als Verteilung, die angibt, mit welcher Wahrscheinlichkeit das gesuchte Objekt an der entsprechenden Position vorhanden ist, müssen zwei Fälle unterschieden werden: im ersten Fall konnte die Person erfolgreich verifiziert werden. Dies äußert sich darin, dass die Verteilung mit einer entsprechend großen Zahl an Stützstellen in der Umgebung dieser Position abgetastet wird. Konnte keine Personenverifikation erfolgen, werden die Stützstellen homogen über der gesamten Bilddimension verteilt. Die in diesem Fall typischerweise zu approximierende multimodale Verteilung stellt auch den entscheidenden Grund für die Wahl des eigentlichen Tracking-Verfahrens dar. Für den Fall, dass die initiale Position der Person bekannt ist, könnte ebenso gut ein Kalman-Filter das Tracking übernehmen. Allerdings würde das Kalman-Filter versagen, wenn der zweite genannte Fall eintritt und gleichzeitig mehrere Positionshypothesen über die Zeit verfolgt werden müssen (für eine ausführliche Diskussion dazu siehe [IB98]).

3D-Tracking Auch beim Personentracking liegt wie im Fall der Selbstlokalisierung des Roboters (Abschnitt 7.3) ein dreidimensionaler Zustandsraum vor. Die Koordinaten x und y beschreiben hier die Position innerhalb eines Bildes. Die dritte Dimension z wird hier nicht durch die Orientierung des Objekts aufgespannt, sondern durch dessen Abstand zur Kamera. Dem wird ebenso wie bei der Personenverifikation durch die Verwendung von Auflösungsipyramiden als Basisstruktur, innerhalb der der Trackingprozess abläuft, Rechnung getragen. Entsprechend erfolgt die Platzierung der Stützstellen sowohl unmittelbar auf als auch zwischen den Auflösungsebenen, wobei für den letztgenannten Fall eine einfache lineare Interpolation der sensorischen Beobachtung erfolgt.

Betont werden soll an dieser Stelle, dass sich der Trackingprozess ebenso wie die Personenverifikation über mehrere Auflösungsebenen (z.B. drei beim Tracking mit der omnidirektionalen Kamera) erstreckt. Um die Beschreibung jedoch nicht unnötig zu verkomplizieren, wird hier nur der zweidimensionale Fall betrachtet.

Wahrnehmungsmodell Für die Bewertung der Stützstellen bedarf es eines Wahrnehmungsmodells, das die an einer Stützstellenposition erwartete Beobachtung beschreibt. Dieses Modell setzt sich aus zwei Komponenten zusammen:

- Die erste Komponente wurde unmittelbar aus der Personenverifikation abgeleitet und beinhaltet das dort verwendete Kopf-Schulter-Kontur-Modell (Abschnitt 8.2.3). Im Gegensatz zur Kopf-Schulter-Partie-Detektion im Rahmen der Personenverifikation kommt beim Personentracking jedoch nur ein einstufiges Verfahren zum Konturvergleich zum Einsatz. Weiterhin werden die Parameterintervalle so angepasst, dass eine größere Invarianz gegenüber leichten perspektivischen Änderungen der Kontur vorliegt. Auf die im Rahmen der Personenverifikation eingesetzte abschließende Schwellenoperation wurde

verzichtet, um ein kontinuierliches Ähnlichkeitsmaß berechnen zu können. Dies führt zwar einerseits dazu, dass auch Personen in nahezu seitlicher Ansicht verfolgt werden können, allerdings bedingt dies auch eine enorme Zunahme an falsch positiven Hypothesen, wie dies in Abbildung 8.9 bereits verdeutlicht wurde. Die Berechnung des Ähnlichkeitsmaßes erfolgt analog zu der im Rahmen der KSP-Detektion (Abschnitt 8.2.3.2) angegebenen Gleichung 8.15 und geht als d_{KSP} in Gleichung 8.23 zur Bewertung der entsprechenden Stützstelle ein.

- Um die Bewertung der Stützstellen spezifischer zu gestalten, als das allein durch das KSP-Modell möglich ist, wird jetzt als zweite Komponente zusätzlich ein Farbmerkmal für die Region bestimmt, die das Modell der Kopf-Schulter-Kontur abdeckt (Abbildung 8.18). Alle in Abbildung 8.18 grau gekennzeichneten Pixel werden zur Berechnung eines Farbmodells (Mittelwertvektor und Kovarianzmatrix) im dichromatischen rg-Farbraum herangezogen. Das mit d_{color} bezeichnete Ähnlichkeitsmaß bezüglich des aktuellen Referenzfarbmodells (auf Basis des Mahalanobis-Abstands) geht ebenfalls in die visuell-basierte Gewichtung $P_{vis}(k)$ der Stützstelle an der 3D-Position k nach Gleichung 8.23 ein. Dabei betragen die Maximalwerte der beiden Abstandsmaße jeweils 1 im Falle der bestmöglichen Übereinstimmung mit dem Referenzmodell.

$$P_{vis}(k) = \gamma \cdot \max(d_{KSP}(k), d_{color}(k)) + (1 - \gamma) \cdot \min(d_{KSP}(k), d_{color}(k)) \quad (8.23)$$

Der damit realisierte Min-Max-Operator erlaubt eine flexible Parametrisierung, wobei sich in den bisherigen Experimenten der Wert $\gamma = 0.9$ als geeignet herausgestellt hat.

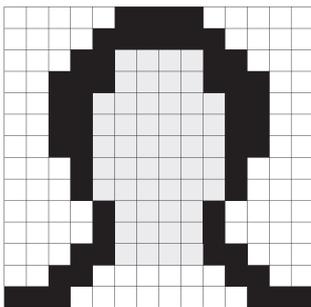


Abbildung 8.18: Darstellung der Region (grau markierter Bereich), in der das Farbmodell extrahiert und neben dem Konturmerkmal zum Personentracking verwendet wird.

Für die Berechnung des initialen Belief wird davon ausgegangen, dass sich die zu verfolgende Person dem Roboter zuwendet, was legitim ist, da der Trackingprozess entweder durch die Personenverifikation oder alternativ während des Einbuchens des Kunden auf dem Touchscreen initialisiert und gestartet wird. Dies impliziert die Nutzung des Hautfarbmodells, welches bereits im Rahmen der Personenverifikation zum Einsatz kam, als initiales Farbmodell.

Um zu berücksichtigen, dass sich das Farbmodell durch variable Beleuchtungsbedingungen und abhängig von der aktuellen Ansicht ändern kann, müssen die Parameter des Farbmodells kontinuierlich adaptiert werden. Dies wird mit einem einfachen rekursiven Tiefpass realisiert.

Dabei ist zu beachten, dass eine Adaption nur dann erfolgt, wenn das berechnete Ähnlichkeitsmaß d_{color} einen vorgegebenen Schwellwert nicht unterschreitet. Experimentelle Untersuchungen ergaben, dass die alleinige Adaption des Mittelwertvektors bei konstanter Kovarianzmatrix im Vergleich zur Adaption der Kovarianzmatrix zu deutlich stabileren Ergebnissen führt.

Bewegungsmodell Um die Bewegung des zu verfolgenden Objekts zu modellieren, kommt ein Bewegungsmodell zum Einsatz. Im Gegensatz zur visuell-basierten Selbstlokalisierung des Roboters, wo anhand der Odometrieinformation zumindest eine Schätzung der real ausgeführten Bewegung vorliegt, kann im hier vorliegenden Fall ein explizites Modell der zu erwartenden Bewegung des Objekts kaum angegeben oder experimentell ermittelt werden. Deshalb enthält das Bewegungsmodell keinen deterministischen Anteil, sondern es erfolgt ausschließlich eine stochastische Verschiebung der Stützstellen, die durch eine geeignet parametrisierte bivariate Gaußfunktion realisiert wird.

Neubewertung der Stützstellen und erneute Abtastung Nach dieser Verschiebung werden die Stützstellen entsprechend Gleichung 8.23 neu bewertet. Für die erneute Abtastung der zu approximierenden Verteilungsfunktion im nächsten Verarbeitungsschritt ist entscheidend, dass die Wahrscheinlichkeit, mit der eine Stützstelle auch im Folgeschritt für die Approximation verwendet wird, proportional zu ihrem Gewichtswert im vorhergehenden Abtastschritt ist, was dazu führt, dass die Verteilungsfunktion vorrangig an den „wichtigen“ Stellen approximiert wird, an denen der Belief bereits einen hohen Wert erreicht hat.

Analog zum in Abschnitt 7.3 dargestellten MCL-Verfahren gilt wiederum, dass die Anzahl der Stützstellen über alle Folgeschritte konstant bleibt, und dass die Bewertungen der Stützstellen nach jedem Aktualisierungsschritt so normiert werden, dass ihre Summe den Wert eins ergibt.

Für das Personentracking können ebenfalls beide zur Verfügung stehenden Kamerasysteme (Omnikamera oder Frontalkamera(s)) eingesetzt werden. Während mit den Bildern der Omnikamera, bedingt durch die geringe Detailauflösung, nur ein Entfernungsbereich bis ca. 1.5 m abgedeckt werden kann, ist der erfassbare Bereich der Frontalkameras durch deren Zoomfunktion deutlich größer, allerdings treten hier Probleme bei sehr nahe am Roboter befindlichen Personen auf. Sinnvoll ist deshalb perspektivisch die kombinierte Nutzung beider Kamerasysteme, wobei ein sich überlappender Entfernungsbereich angestrebt wird, um das zu verfolgende Objekt sicher von einem zum anderen Kamerasystem übergeben zu können.

Wird die Person mittels der omnidirektionalen Kamera verfolgt, kommen drei Auflösungsebenen zum Einsatz, die einen Entfernungsbereich von null bis etwa 1,5 Meter (bezogen auf die Begrenzung des Roboters) abdecken. Analog zur Personenverifikation wird auch hier der zu analysierende Bereich des omnidirektionalen Bildes auf einen Bereich von $\pm 30^\circ$ um den von der Aufmerksamkeitssteuerung bzw. von der Personenverifikation gelieferten Winkel eingeschränkt und anschließend von Polarkoordinaten in kartesische Koordinaten transformiert. Beim Einsatz der Frontalkamera(s) besteht die Auflösungspyramide aus fünf Ebenen, was einen erfassbaren Entfernungsbereich von 0,5 bis etwa 2,5 Meter bedeutet.

8.3.2 Erweiterungen

Da die Personenverifikation kontinuierlich weiter berechnet werden kann, können deren Ausgaben in den Trackingprozess eingekoppelt werden, was zu einer Erhöhung der Robustheit des Trackingprozesses führt.

Unterstützend kommt weiterhin eine einfache Auswertung der Sonarinformation zum Einsatz, die den entsprechenden Bereich, in dem sich die gerade verfolgte Person befindet, überwacht. Aus der so gewonnenen Information kann wie auch aus dem visuellen Trackingprozess auf die Entfernung des Nutzers zum Roboter geschlossen werden. Momentan wird lediglich versucht, die Person so nahe wie möglich am Roboter zu halten, was sich in einer entsprechenden sonar-basierten Bewertung der Stützstellen P_{sonar} mit dem Wert 1 bei minimalem Abstand und einem linearen Abfall bis zum Wert Null bei ca. 1.5m Entfernung zum Roboter niederschlägt. Die Bewertung P_{ges} einer Stützstelle k für das Personentracking setzt sich danach aus der visuell- und der sonar-basierten Bewertung nach Gleichung 8.24 zusammen.

$$P_{ges}(k) = P_{vis} \cdot P_{sonar} \quad (8.24)$$

Da im vorliegenden Einsatzszenario nicht davon ausgegangen werden kann, dass das kontinuierliche Kontakthalten zum aktuellen Nutzer unter allen Umständen sichergestellt werden kann, sind Verfahren nötig, die es ermöglichen, den aktuellen Kunden von weiteren Personen in der Einsatzumgebung sicher zu unterscheiden. Dies kommt dann zum Tragen, wenn der Serviceroboter nach zwischenzeitlichem Kontaktverlust Personen lokalisiert bzw. verifiziert und entscheiden muss, ob es sich bei einem der Kandidaten mit hoher Wahrscheinlichkeit um den aktuellen Kunden handelt. Diese Problemstellung wurde bislang noch nicht berücksichtigt, muss aber für den erfolgreichen Einsatz des Systems unbedingt noch gelöst werden.

Weiterhin angestrebt wird ein einheitliches multimodales Trackingsystem, das perspektivisch auch den Laserscanner und weitere visuelle Verfahren (z.B. das in [RR98] vorgestellte Zero-Disparity-Filtering in Verbindung mit stereobasierten Verfahren) umfassen wird, wie dies beispielsweise auch in [Fey00] vorgeschlagen wird. Ebenso muss betont werden, dass zum aktuellen Stand davon ausgegangen wird, dass der Benutzer dem Shopping-Assistenten zu einer Zielposition im Markt folgt, wobei der Roboter versucht, den Kontakt möglichst kontinuierlich aufrecht zu erhalten. Allerdings lässt sich der Fall, dass der Roboter dem Benutzer durch den BauMarkt folgt, mit dem aktuellen Implementationsstand noch nicht robust realisieren.

Alle hier vorgestellten Erweiterungen zum visuellen Personentracking in Richtung eines multimodalen Trackingsystems sind Gegenstand aktueller Untersuchungen am Fachgebiet und können derzeit noch nicht in angemessenem Umfang mit experimentellen Ergebnissen unteretzt werden. Einige der o.g. Erweiterungen wurden jedoch bereits in den nachfolgend dargestellten Trackingergebnissen berücksichtigt.

8.3.3 Experimentelle Ergebnisse

Die Beispielsequenzen zum Personentracking geben zunächst Ergebnisse von Experimenten zum visuellen Tracking wieder. Die Abbildungen 8.19 und 8.20 verdeutlichen, wie sich der visuelle Trackingprozess bei alleiniger Verwendung der visuell-basierten Bewertung der Stützstellen beim Condensation-Algorithmus verhält.



Abbildung 8.19: Beispielsequenz zum Personentracking mittels Frontalkamera in der Laborumgebung. Durch die kontinuierliche Adaption des Farbmodells kann die Person auch dann verfolgt werden, wenn sie sich vom Roboter abwendet (bei Verwendung eines stationären Hautfarbmodells wäre dies nicht gewährleistet). Die Rahmen markieren die Stützstellen des Condensation-Algorithmus und korrespondieren mit der Größe des Konturmodells. Grün markiert ist die Position mit der höchsten Bewertung, rote Rahmen geben die Positionen weiterer hoch bewerteter Stützstellen an.

Abbildung 8.19 zeigt Ergebnisse eines Trackingexperiments im Labor bei Verwendung einer Frontalkamera, wobei auch hier die Kombination aus Kontur- und Farbmodell zum Einsatz kam. Durch diese Kombination konnte die Person einerseits relativ problemlos verfolgt werden, weiterhin ist durch die Verwendung eines adaptiven Farbmodells für das Personentracking auch die Möglichkeit gegeben, dass sich die Person vom Roboter (von der Kamera) abwenden kann.

Ein exemplarisches Beispiel für ein Experiment zum visuell-basierten Tracking unter Verwendung der omnidirektionalen Kamera in einer Laborumgebung zeigt die erste Sequenz in Abbildung 8.20. Prinzipiell kommt hier der gleiche Tracking-Algorithmus zum Einsatz, allerdings erfolgt hier eine Transformation des omnidirektionalen Bildes in ein kartesisches Bild, wobei auch die parabolische Form des Spiegels korrigiert wird.

Den Einfluss, den die Hinzunahme der Sonarinformation in den Trackingprozess bewirkt, zeigt die zweite Sequenz in Abbildung 8.20. Im Gegensatz zu der in der gleichen Abbildung ersten Sequenz erfolgt das Verfolgen der Person deutlich robuster, und die Person kann kontinuierlich im Erfassungsbereich gehalten werden.

Abschließende Beispiele (Abbildungen 8.21 und 8.22) soll das Personentracking in der realen BauMarkt-Umgebung mit Hilfe der omnidirektionalen Kamera verdeutlichen. Hier wurde wiederum die Kombination aus visueller und Sonarinformation verwendet, um die Verteilung der Stützstellen innerhalb der drei verwendeten Auflösungsebenen zu beeinflussen.



Abbildung 8.20: Oben dargestellt wurde eine Beispielsequenz zum visuell-basierten Personen-tracking mittels Omnikamera in einer Laborumgebung, jeweils auf dem gesamten Panoramabild, wobei sich die zu verfolgende Person um den Roboter herum bewegt. Die Punkte markieren die Stützstellen des Condensation-Algorithmus. Entsprechend ihrer Bewertung wurden diese farbig markiert. Dabei symbolisieren Rot eine hohe, Schwarz eine geringe Bewertung. An den Positionen mit der höchsten Bewertung wurde ein grüner Rahmen eingeblendet, der mit der Größe des Kopf-Schulter-Modells korrespondiert. Deutlich erkennbar ist in der oberen Sequenz, dass das Trackingverfahren im Bereich der Tür das Target verliert, da durch die dort vorhandene Farbinformation die Stützstellen offensichtlich höher bewertet wurden als auf der Person selbst. Die untere Sequenz illustriert, dass das Trackingverfahren durch die Hinzunahme der Sonarinformation deutlich robuster arbeitet und die Person kontinuierlich verfolgt werden kann.



Abbildung 8.21: Beispielsequenz zum Personentracking mittels Omnikamera in der realen BauMarkt-Umgebung, wobei die zu verfolgende Person im Vordergrund zu sehen ist. Dargestellt ist hier nur der entsprechende Ausschnitt des Panoramabildes. Hier markiert der rot markierte Rahmen die am höchsten bewertete Stützstelle.

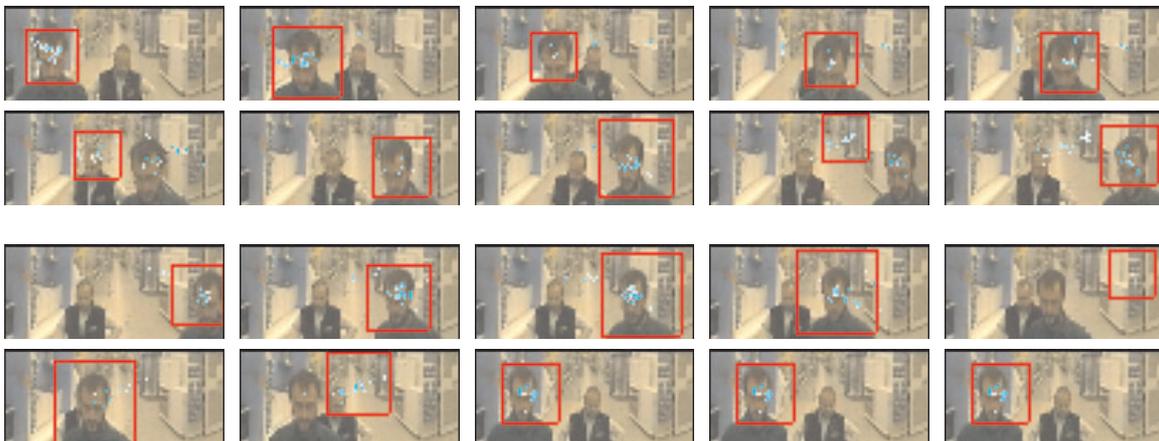


Abbildung 8.22: Beispielsequenzen zum Personentracking (jeweils v.l.n.r.) mittels Omnikamera in der realen BauMarkt-Umgebung, wobei die zu verfolgende Person im Vordergrund zu sehen ist. In beiden Sequenzen wird der Einfluss der Sonarinformation deutlich. Durch die Einbeziehung des Abstandes zum zu verfolgenden Objekt kann verhindert werden, dass das Trackingverfahren von der Person im Vordergrund auf die sich unmittelbar dahinter bewegende Person übergeht. Der rote Rahmen markiert wiederum die Position der am besten bewerteten Stützstelle des Condensation-Algorithmus.

8.4 Weitere Verfahren zur visuell-basierten MRI

Im Kontext des PERSES-Projektes wurden weiterhin Methoden entwickelt, die auf eine gestenbasierte Mensch-Roboter-Interaktion abzielen [BKB⁺98] [BBCG99] [BBB⁺98b].

Hinsichtlich der eigentlichen Gestenerkennung und -interpretation gelten hier jedoch ebenso die in Abschnitt 2.4.2.2 diskutierten Einschränkungen, die einen robusten Einsatz dieser Verfahren unter Realweltbedingungen derzeit noch ausschließen. Allerdings muss festgestellt werden, dass für das konkrete BauMarkt-Szenario neben den auftretenden praktischen Problemen auch hinsichtlich der inhaltlichen Schlüssigkeit momentan kaum Einsatzmöglichkeiten gesehen werden, da nicht davon ausgegangen werden kann, dass die Benutzer des Shopping-Assistenten vorab über Kenntnis bezüglich der vom Roboter wahrnehmbaren und interpretierbaren Gesten haben.

Da die Auswertung von Gesten im Sinne von Körpersprache jedoch perspektivisch auch für den Shopping-Assistenten Relevanz erlangen wird, wurde versucht, hier zumindest methodische Fortschritte zu erzielen. So konnte im Rahmen eines EU-TMR-Projektes eine Promotion abgeschlossen werden, die ein hybrides Gesamtsystem (Kombination von Hidden-Markov-Modellen und neuronalen Netzen) zum Gegenstand hatte [Cor00]. Während für experimentellen Untersuchungen weitgehend ideale Umgebungsbedingungen angenommen wurden, erfolgte parallel dazu die Entwicklung der in den vorangegangenen Abschnitten beschriebenen Verfahren zur robusten Wahrnehmung von Personen unter realen, uneingeschränkten Umgebungsbedingungen mit dem Ziel, beides zu kombinieren [CBG00].

Bezüglich der Interpretation von Körpersprache soll hier zumindest auf die Arbeiten von DAVIS am MIT hingewiesen werden [DB97] [DB98], die einen möglichen Ausgangspunkt darstellen können, insbesondere deshalb, weil sie sehr einfach zu implementieren und damit sehr effizient anzuwenden sind. Allerdings besteht auch hier das Problem, dass nur Ergebnisse für einen sehr stark eingeschränkten Satz von Aktionstemplates existieren und von einer stationären Kamera ausgegangen wird. Es ist daher schwer abschätzbar, wie diese Verfahren bei deutlicher Vergrößerung des Satzes an zu erkennenden Verhaltensweisen und beim Einsatz auf einem mobilen Robotersystem skalieren.

8.5 Grafisches Nutzerinterface

Ein grafisches Nutzerinterface dient der unmittelbaren Interaktion zwischen dem BauMarkt-Kunden und dem Shopping-Assistenten. Dieses Interface wurde nur zu Demonstrationszwecken implementiert und ist dementsprechend einfach gehalten. Es besteht aus einer grafischen Bedienoberfläche und einer einfachen relationalen Datenbank, in der die einzelnen Warengruppen bzw. Artikel, die vom Benutzer ausgewählt werden können, verwaltet werden.

Nach einem Begrüßungsbildschirm stehen die großen Marktbereiche zur Auswahl (Abbildung 8.23), wobei der Malerbereich etwas detaillierter mit Warengruppen und typischen Artikeln untersetzt ist (Abbildung 8.24).

Nach der Auswahl eines bestimmten Artikels und dem Betätigen des Schalters „Losfahren“ wird eine BauMarkt-Karte eingeblendet, in der die aktuelle Position des Roboters und die Zielposition, die jetzt angefahren wird, hervorgehoben sind (Abbildung 8.25).

Jeder Bildschirm wird mit einer korrespondierenden Ansicht aus dem BauMarkt hinterlegt,



Abbildung 8.23: Begrüßungsbildschirm des grafischen Interaktionstools (links) und Überblick über die verschiedenen Hauptabteilungen des BauMarktes (rechts).

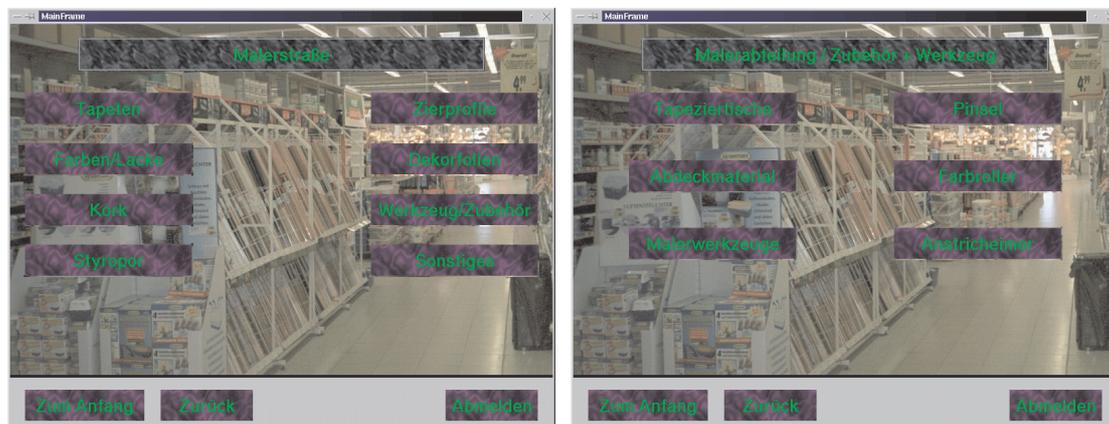


Abbildung 8.24: Untersetzung des Malerbereichs mit Warengruppen und Artikeln.

die entweder den Marktbereich oder die ausgewählte Produktgruppe so zeigt, wie sie real im Markt vorliegen.

Auch wenn das grafische Interaktionstool derzeit nur Demonstrationscharakter hat, kann es dennoch als Prototyp für die unmittelbare Nutzer-Roboter-Interaktion im realen Einsatzfall dienen.

8.6 Sprachausgabe und Robotergesicht

Die Akzeptanz des Serviceroboters wird maßgeblich davon abhängen, wie gut der Roboter in der Lage ist, sich selbst und seine Intentionen intuitiv zu artikulieren. Im Gegensatz zur Spracherkennung, die für den vorliegenden Einsatzfall derzeit noch keine Rolle spielt, kommt der Sprachausgabe hier eine wesentliche Rolle zu. Mittels Sprache kann sich der Roboter vorstellen, seine Dienste erklären und anbieten, oder einfach nur mitteilen, was er als nächstes zu tun gedenkt. Für die konkrete Umsetzung der Sprachausgaben wurden für die entsprechenden Situationen, in denen eine Ausgabe erfolgen soll, Sprachdateien erstellt, die dann entsprechend



Abbildung 8.25: Hätte sich der Kunde für Abdeckmaterial entschieden, würde der links dargestellte Bildschirm erscheinen, der die Ansicht des entsprechenden Standorts wiedergibt. Nach der Auswahl eines Artikels und dem Kommando „Losfahren“ (links) erscheint eine BauMarkt-Karte, in der die aktuelle Position und die Zielposition hervorgehoben sind (rechts).

ausgegeben werden (siehe auch Kapitel 9).

Inspiziert durch das smarte Gesicht des Roboters MINERVA (Abschnitt 3.2), wurde der Shopping-Assistent PERSES ebenfalls mit einem sehr einfachen „technischen“ Gesicht ausgestattet (Abbildung 8.26).

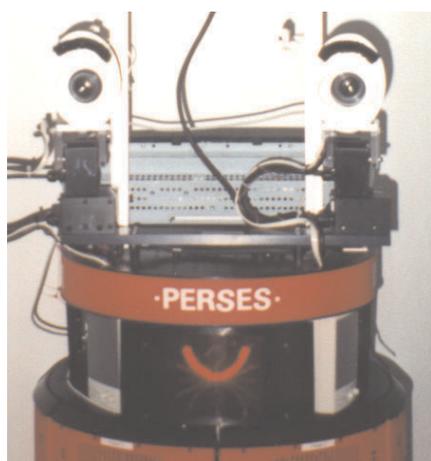


Abbildung 8.26: Robotergesicht, bestehend aus augenähnlichen Kamerafronten und einem ansteuerbaren Mund, der über ein Diodenarray verschiedene Stellungen annehmen kann.

Dabei bilden zwei Kamerafronten die beiden Augen. Der Mund wurde als steuerbare Leuchtdiodenanordnung realisiert. Spricht der Roboter, so bewegt sich für die entsprechende Zeit auch der Mund. Weiterhin können „emotionale“ Zustände des Roboters wie etwa die Freude über einen wahrgenommenen potentiellen Interaktionspartner oder die Verärgerung über die Versperrung des Weges leicht und intuitiv verständlich vermittelt werden.

8.7 Zusammenfassung

Mit den in diesem Kapitel beschriebenen Methoden konnten die informationstechnischen Voraussetzungen für eine möglichst intuitiv und natürlich wirkende Mensch-Roboter-Interaktion geschaffen werden. Mit Ausnahme des grafischen Interaktionstools, das sich verständlicherweise stark an den Anforderungen der vorliegenden Applikation Shopping-Assistent im Bau-Markt orientiert, lassen sich alle dargestellten Verfahren prinzipiell auf eine Vielzahl weiterer Anwendungen von Servicerobotern übertragen. Für die Akzeptanz solcher Systeme, die mit beliebigen Personen interagieren sollen, wird der Aspekt der einfachen Benutzung in Verbindung mit einem möglichst intuitiven und natürlichen Gesamtverhaltens des Serviceroboters zukünftig eine zentrale Rolle spielen. Je besser es gelingt, den Roboter mit Verhaltensweisen auszustatten, die der Benutzer auch von einem menschlichen Interaktionspartner erwarten würde, umso weniger werden Berührungsängste und Zweifel an der Sinnfälligkeit solcher Systeme in den Vordergrund gelangen. Genau diese Zielstellung wurde mit der Entwicklung der in diesem Kapitel vorgestellten Verhaltensmodule verfolgt.

Aufmerksamkeitssteuerung

Die Aufmerksamkeitssteuerung generiert Hypothesen über vorhandene Personen in der unmittelbaren Umgebung des Shopping-Assistenten. In einem multimodalen Ansatz wurden sowohl visuell-basierte als auch akustische Verfahren zu einem Gesamtsystem kombiniert. Diese Multimodalität einerseits und die Auswertung der Bewegungsrichtung bewegungsbasiert detektierter Regionen andererseits sichern ein natürliches und intuitiv verständliches Zuwendungsverhalten des Roboters. Personen in der Umgebung des Roboters können somit beobachten, dass sie vom Roboter aktiv wahrgenommen werden.

Personenverifikation

Um festzustellen, ob sich in der durch die Aufmerksamkeitssteuerung angegebenen Richtung tatsächlich ein potentieller menschlicher Interaktionspartner befindet, wird eine anschließende Personenverifikation durchgeführt. Diese nutzt ausschließlich den visuellen Kanal und basiert auf der Verwendung verschiedener personentypischer Cues (Hautfarbe, Gesichtsstruktur, Kopf-Schulter-Silhouette), die in einem dreidimensionalen Feld dynamischer Neuronen so fusioniert werden können, dass eine sehr robuste Verifikation in einem definierten Entfernungsbereich erfolgen kann. Das Hauptaugenmerk bei der Entwicklung dieses Moduls lag auf der Vermeidung von falsch-positiven Verifikationen, um eine Kontaktaufnahme des Roboters mit unbelebten Objekten möglichst auszuschließen.

Die Personenverifikation stellt ein relativ aufwendiges Verfahren dar, was angesichts der Komplexität der Einsatzumgebung wenig verwundern dürfte, was sich aber auch in einer entsprechenden Rechenzeit niederschlägt, die momentan knapp 2 Sekunden pro Szene beträgt. Da implementationsseitig hier kaum mehr Spielraum für eine effektivere Umsetzung gegeben ist, besteht nur die Möglichkeit, durch den Einsatz leistungsfähigerer Hardware die Performanz dieses Moduls zu erhöhen.

Auf den Umstand der momentan nur bedingt befriedigenden Rechenzeit zurückzuführen ist letztlich auch die Tatsache, dass durch die entstehende Verzögerung durch die Bildauswertung

die mit der Personenverifikation angestrebte Initialisierung des Personentrackings nur dann robust funktioniert, wenn sich die zu verfolgende Person relativ langsam bzw. direkt auf den Roboter zu bewegt.

Personentracking

Bei der Auswahl der Methoden für das Personentracking wurde der o.g. Tatsache dadurch Rechnung getragen, dass das letztlich verwendete MCL-Verfahren (Condensation-Algorithmus) in der Lage ist, auch multiple Hypothesen über die momentane Position des Objekts zu behandeln.

Für das visuelle Personentracking kommen die Merkmale (Haut)farbe und Kopf-Schulter-Silhouette zum Einsatz, die auch für die Personenverifikation verwendet wurden. Durch die Hinzunahme von sonarbasierter Abstandsinformation in Richtung der zu verfolgenden Person konnte eine erhebliche Erhöhung der Robustheit des Trackingverfahrens erreicht werden, was wiederum den Einsatz multimodaler Trackingsysteme für reale Einsatzszenarien im allgemeinen motiviert.

Klassische Interaktionsformen

Alle Bemühungen in Richtung natürlicher und intuitiv verständlicher Mensch-Roboter-Interaktion dürfen nicht darüber hinwegtäuschen, dass an eine vollständige Substitution klassischer Eingabegeräte wie Tastatur oder Touchscreen in absehbarer Zeit noch nicht zu denken ist.

Der in dieser Arbeit gewählte Ansatz versucht vielmehr, nach wie vor notwendige klassische Medien zur unmittelbaren MRI in ein intuitiv und natürlich wirkendes Gesamtverhalten einzubetten.

Das entwickelte grafische Interaktionstool hat derzeit noch eher Demonstrationscharakter, kann aber dennoch als Prototyp für die unmittelbare Nutzer-Roboter-Interaktion im realen Einsatzfall dienen.

Roboterartikulation

Mit dem Einsatz eines Moduls zur Sprachausgabe wird versucht, den aktuellen Interaktionsstatus des Roboters gegenüber seinem menschlichen Interaktionspartner möglichst einfach zu artikulieren. Gegenüber grafischen Ausgaben auf dem Display bietet dies den Vorteil, dass die Äußerungen weitgehend unabhängig von der aktuellen Position des Benutzers wahrgenommen werden können.

Das vorgestellte Robotergesicht soll neben der Vermittlung einer „smarten“ Gesamterscheinung des Roboters auch dazu dienen, einfach verständliche „Emotionen“ des Roboters auf natürliche Weise zu verdeutlichen.

Kapitel 9

Steuerarchitektur und Interaktionszyklus

Im Abschnitt 2.3 wurden einige grundlegende Prinzipien für den Entwurf einer geeigneten Steuerarchitektur diskutiert, die vom Sense-Plan-Act-Paradigma über die Subsumption-Architektur bis zu hybriden Ansätzen reichen.

Im vorliegenden Einsatzfall liegen zunächst verschiedene Module vor, die mehr oder weniger unabhängig voneinander bestimmte Teilverhalten realisieren und in den zurückliegenden Kapiteln ausführlich beschrieben und diskutiert wurden. Abschließend geht es nun darum, diese Teilmodule so zu koordinieren, dass ein sinnfälliges und an die jeweilige Situation angepasstes Gesamtverhalten erzielt werden kann.

Ausgehend von der Aufteilung in Teilverhalten soll eine verhaltensbasierte Steuerarchitektur entwickelt und umgesetzt werden, die neben reaktiven Komponenten auch planende Verfahren integriert und somit letztendlich keinem der klassischen Steuerparadigmen zugeordnet werden kann, sondern selbst wieder eine hybride Architektur darstellt.

Abschnitt 9.1 beschreibt zunächst an einem einfachen Beispiel die Koordination von zwei elementaren Teilverhalten während der Mensch-Roboter-Interaktion. Daran anschließend erfolgt in Abschnitt 9.2 die Darstellung des aktuellen Realisierungsstandes zur Gesamtablaufsteuerung.

Betont werden soll an dieser Stelle, dass der Gesamtkomplex *Steuerarchitektur und Verhaltenskoordination* Gegenstand aktueller Forschungsarbeiten am Fachgebiet ist. Die nachfolgend beschriebenen Verfahren stellen also in erster Linie Konzepte dar, die zunächst anhand der Koordination einfacher Teilverhalten verifiziert werden konnten. Der letztendliche Entwurf und die Realisierung einer umfassenden Steuerarchitektur für den Shopping-Assistenten stehen noch aus und sind nicht mehr Gegenstand dieser Arbeit.

9.1 Koordination von Teilverhalten - Grundidee des Dynamic Approach

Ausgehend vom verhaltensbasierten Ansatz wurde zunächst nach einer Möglichkeit gesucht, eine Steuerarchitektur zu entwerfen, die sowohl einzelne Verhaltensmodule koordinieren kann und in die weitere Verhaltensmodule möglichst einfach integriert werden können. Der am Institut für Neuroinformatik der Ruhr-Universität Bochum entwickelte so genannte „Dynamische Ansatz zur Verhaltenssteuerung“ [Ste97] [SB98] [BBD⁺99], der in ähnlicher Weise auch von

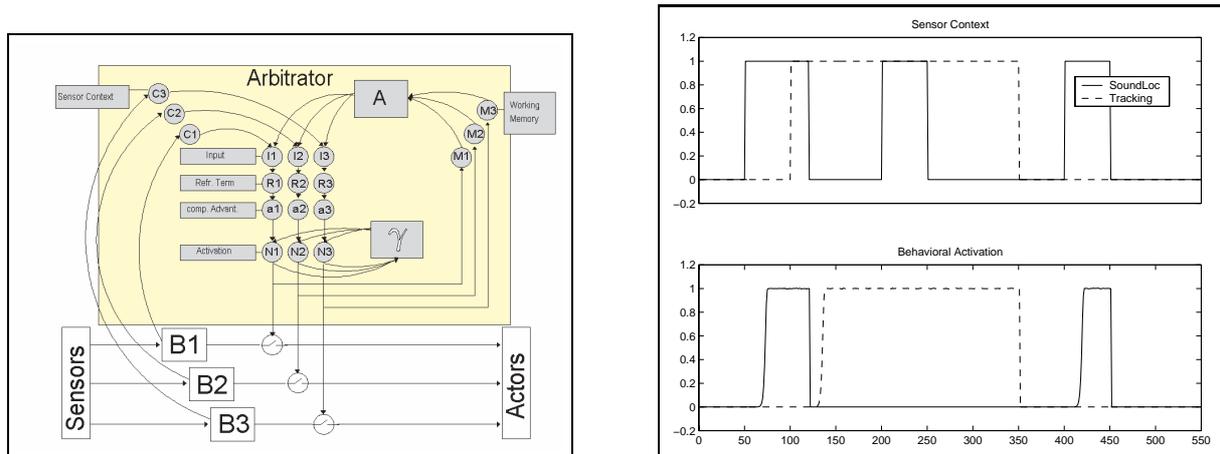


Abbildung 9.1: Links: Schematische Darstellung der Architektur zur Verhaltenskoordination (adaptiert aus [SB98]); Rechts: Zeitverläufe der Variablen *Sensor Context* und Verhaltensaktivierung für das im Text erläuterte Beispiel zur Koordination von visuellem Personentracking und akustischer Personenlokalisierung.

anderen Arbeitsgruppen eingesetzt wird (siehe z.B. [BcJ⁺00] [LC99]), stellt eine solche Möglichkeit dar. Das Kernstück dieser Architektur bildet der Arbitrator, der auf einem nichtlinearen DGL-System basiert, welches die Interaktion der verschiedenen Verhaltensmodule modelliert und dessen Attraktoren gewünschte Teilverhalten darstellen. Anhand der in Abbildung 9.1 rechts dargestellten Zeitverläufe soll die Funktionsweise dieses Systems zumindest grob veranschaulicht werden: Zwei Verhalten, das Tracken einer Person (B1) und das Reagieren auf ein akustisches Ereignis (Geräuschlokalisierung, B2), wurden in dem Beispiel realisiert. Im Initialzustand sind beide Verhaltensmodule nicht aktiviert (*Behavioral Activation* $N1 = 0$ und $N2 = 0$). Beide Module teilen über ihren *Sensor Context* dem Arbitrator mit, inwieweit sie in der aktuellen sensorischen Situation in der Lage sind, ein hinreichend gutes Berechnungsergebnis zu liefern. Nachdem ein akustisches Ereignis aufgetreten ist, wird das Modul zur Geräuschlokalisierung aktiv und inhibiert die Aktivierung des Trackingverhaltens. Nachdem die Schallquelle lokalisiert wurde, erfolgt ein Schwenken der Kameras in die entsprechende Richtung. Nun beginnt der Trackingprozess, der nun nicht mehr durch die Geräuschlokalisierung inhibiert wird, und der seinerseits die Geräuschlokalisierung inhibiert, was eine Inaktivierung der akustischen Aufmerksamkeit nach sich zieht und sich darin äußert, dass $N2$ vom Zustand 1 in den Zustand 0 übergeht, da bei sicherem Tracking keine „Ablenkung“ durch ein akustisches Ereignis erfolgen soll. Damit wirkt sich ein in dieser Phase auftretendes Geräusch nicht auf das Gesamtverhalten der Plattform aus, da das entsprechende Verhaltensmodul nicht auf die Aktorik zugreifen darf.

Schlägt das Tracking hingegen fehl ($N1$ geht aufgrund des fehlenden *Sensor Context* vom Zustand 1 in den Zustand 0 über), wird die akustische Aufmerksamkeit nicht mehr vom Trackingmodul inhibiert und kann zur Unterstützung der Nutzersuche herangezogen werden. Die zur Verhaltenskoordination notwendigen Randbedingungen, die beispielsweise die gleichzeitige Aktivierung verschiedener Verhaltensmodule verbieten oder definierte zeitliche Randbedingungen (Verhalten X muss vor der Aktivierung von Verhalten Y aktiv sein) festlegen, werden über entsprechende Einträge in die Systemmatrizen A bzw. γ realisiert.

9.2 Gesamtablaufsteuerung

Die in Abschnitt 9.1 vorgestellte Methode gestattet eine sehr elegante Verhaltensbeschreibung und lässt sich problemlos um neue Verhaltensmodule erweitern. Weiterhin können für komplexe Teilverhalten mehrere separate Architekturen mit prinzipiell identischem Aufbau in einer Steuerungshierarchie kombiniert werden.

In [KHH⁺01] wird postuliert, dass das Grundprinzip des Dynamic Approach auch für die Koordination von komplexen Gesamtsystemen geeignet ist bzw. entsprechend skaliert werden kann. In aktuellen Forschungsarbeiten am Fachgebiet wird derzeit versucht, dies auch für die vorliegende Problemstellung, der Koordination aller bereits realisierten Teilverhalten für den Shopping-Assistenten, umzusetzen.

Da das Wechselspiel der Teilverhalten jedoch auch hier letztendlich vom Designer vorgegeben werden muss, wurde zunächst diese Hierarchie der Teilverhalten entworfen, die die Voraussetzung für die Realisierung einer alle Teilverhalten umfassenden Steuerarchitektur darstellt. Die obere Hierarchiestufe umfasst entsprechend Abbildung 9.2 eine Kontrollinstanz, die die Teilverhalten *Mensch-Roboter-Interaktion*, *Navigation* und *Serviceleistungen* steuert. Innerhalb dieser Teilverhalten sind wiederum Kontrollstrukturen zu finden, die die Steuerung des internen Ablaufs dieser Teilverhalten übernehmen.

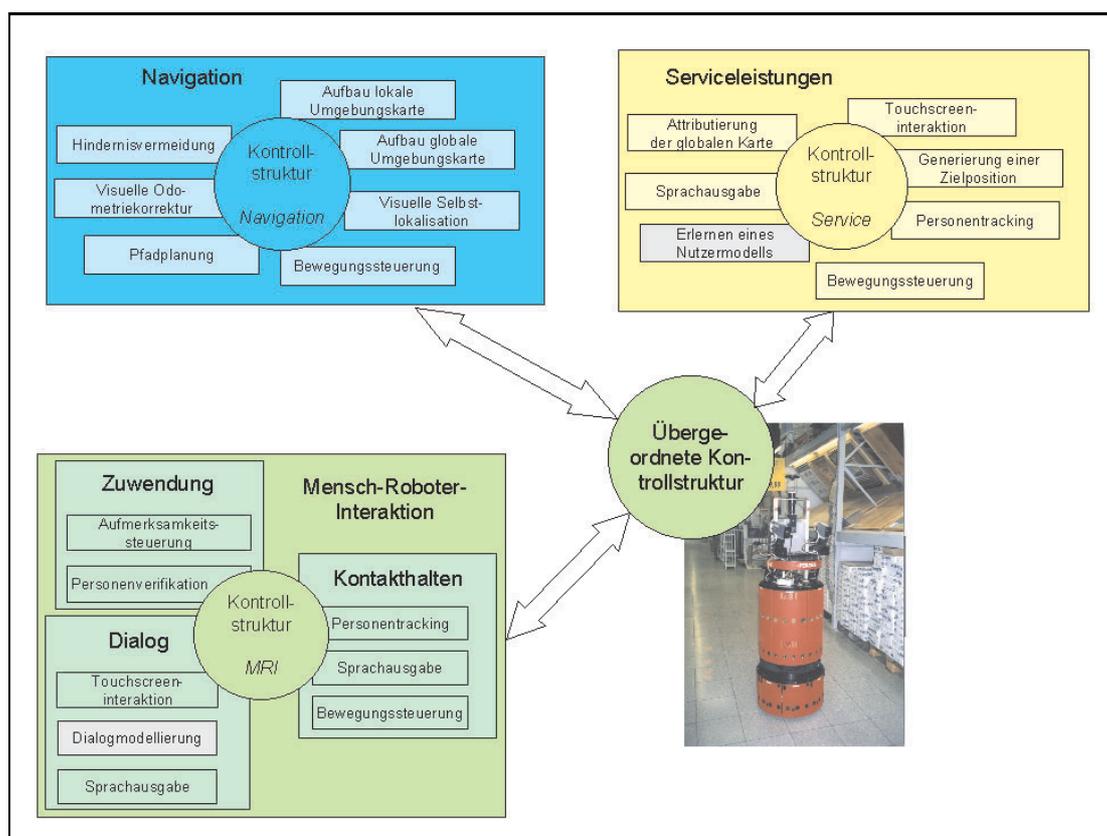


Abbildung 9.2: Überblick über die hierarchische PERSES-Systemarchitektur, die die drei abstrakten Verhaltensmodule Mensch-Roboter-Interaktion, Navigation und Serviceleistungen umfasst. Die grau hinterlegten Module wurden bislang noch nicht realisiert bzw. existieren lediglich auf Konzeptebene.

Die Übertragung des Dynamischen Ansatzes zur Verhaltenskoordination auf das im vorliegenden Abschnitt skizzierte Gesamtsystem stellt, wie oben bereits angedeutet, einen Schwerpunkt der aktuellen Arbeiten am Fachgebiet dar. Aus diesem Grund können hier noch keine experimentellen Ergebnisse präsentiert werden, die über die Koordination einfacher Teilverhalten, wie in Abschnitt 9.1 beschrieben, hinausgehen. Weiterhin muss betont werden, dass die in Abbildung 9.2 dargestellte Strukturierung der Steuerarchitektur lediglich *eine* von mehreren Möglichkeiten darstellt, Teilverhalten zu abstrakten Metaverhalten zu gruppieren.

Angestrebt wird weiterhin, dass das im Rahmen des HERMES-Projektes entwickelte Konzept der Situationsbeschreibung (siehe Abschnitt 4.2.2) aufgegriffen und in das Gesamtkonzept der PERSES-Steuerarchitektur mit einbezogen wird.

9.3 Interaktionszyklus

Aus Abbildung 9.2 wird ersichtlich, dass die Realisierung einer geeigneten und flexiblen Steuerarchitektur angesichts der Komplexität des Gesamtverhaltens eine wissenschaftliche Herausforderung darstellen dürfte.

Für die Umsetzung erster Experimente und die Durchführung von Demonstrationen wurde deshalb zunächst ein relativ starr vorgegebener Ablauf fest programmiert, d.h. das Gesamtsystem Shopping-Assistent dient zum aktuellen Entwicklungsstand in erster Linie dazu, die Funktion der einzelnen Teilverhalten zu beurteilen, was einen relativ starren Ablauf der Interaktion mit sich bringt. Abbildung 9.3 skizziert den Ablauf eines Interaktionszyklus, wie er vom Shopping-Assistenten zum gegenwärtigen Zeitpunkt realisiert und demonstriert werden kann.

Den Ausgangspunkt der Interaktion bildet die Aufmerksamkeitssteuerung, die sich aus der Bewegungsanalyse und der Geräuschlokalisierung zusammensetzt. Nach der anschließenden Drehung in Richtung der vermuteten Person erfolgt eine Personenverifikation, die im positiven Fall eine dem Roboter zugewandte Person als potentiellen Interaktionspartner detektiert. Nach erfolgreicher Verifikation begrüßt der Roboter den potentiellen Interaktionspartner mit einer typischen Sprachsequenz und bietet seine Dienste an. Anschließend hat der Kunde die Möglichkeit, über das grafische Interface beispielsweise den gewünschten Marktbereich oder Artikel auszuwählen, zu dem der Roboter ihn dann lotst. Um sicherzustellen, dass der Kontakt zum aktuellen Kunden möglichst kontinuierlich bestehen bleibt, erfolgt ein visuelles Personen-tracking, welches durch die Personenverifikation initialisiert wird und dessen Robustheit durch die parallel dazu fortgesetzte Verifikation enorm verbessert werden kann. Weiterhin erfolgt eine einfache sonarbasierte Überwachung des Umgebungsbereiches, in dem sich der Nutzer gerade befindet. Wird der Abstand zwischen Nutzer und Shopping-Assistent größer, vermindert letzterer seine Geschwindigkeit bzw. bittet den Kunden, den Erfassungsbereich möglichst nicht zu verlassen.

So lange der Kontakt problemlos aufrecht erhalten werden kann, erfolgt keine Artikulation seitens des Roboters. Detektiert der Roboter einen drohenden Kontaktverlust, bittet er den Kunden via Sprachausgabe, einen kürzeren Abstand zu wahren, oder der Roboter unterbricht alternativ seine Fahrt zur aktuell gültigen Zielregion und versucht seinerseits, dem Kunden hinterherzufahren, um einen Kontaktverlust aktiv zu vermeiden.

Das Lotsen des BauMarkt-Kunden zu einem von diesem ausgewählten Marktbereich oder

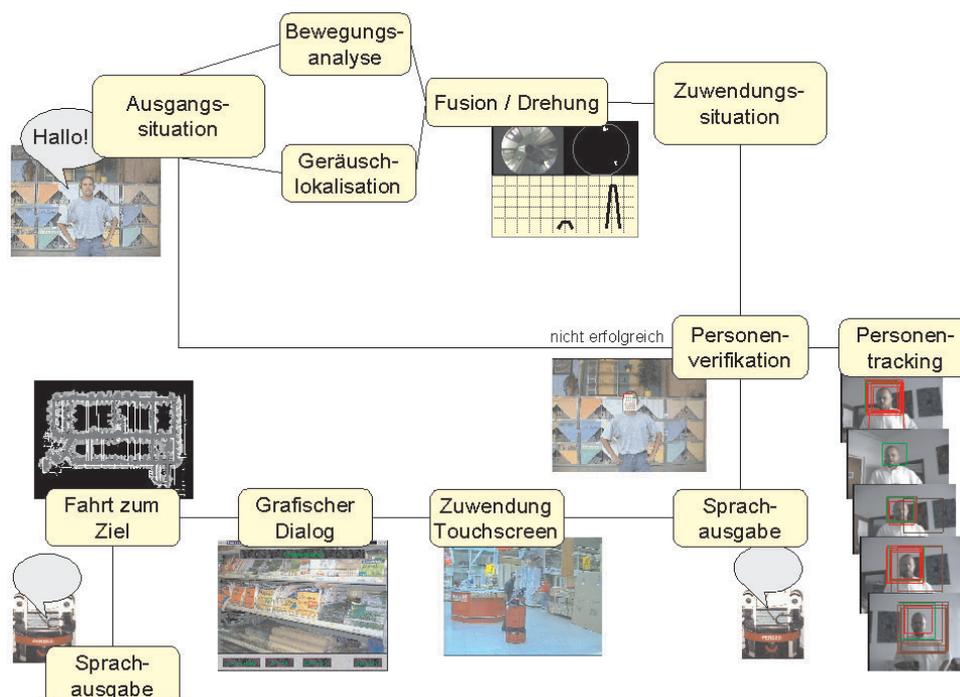


Abbildung 9.3: Überblick über den derzeit realisierten Interaktionszyklus zwischen dem Shopping-Assistenten PERSES und seinem Benutzer.

Artikelstandort stellt dabei die elementare Serviceleistung des Shopping-Assistenten dar und ist gleichzeitig momentan die einzige Serviceleistung, die demonstrabel umgesetzt werden kann. Damit handelt es sich bei dem angestrebten Shopping-Assistenten PERSES ebenso um eine Tourguide-Anwendung wie im Falle der in Kapitel 3 beschriebenen Systeme POLLY, RHINO oder MINERVA, wenn man den derzeitigen Implementationsstand betrachtet.

Um die Lotsenfunktion umzusetzen, muss die vom Shopping-Assistenten gelernte globale Umgebungskarte mit den entsprechenden Informationen attribuiert werden. Für den realen Einsatz des Shopping-Assistenten sollte hier eine unmittelbare Anbindung an das Warenwirtschaftssystem des BauMarktes vorgesehen werden, was eine entsprechende informationstechnische Infrastruktur im BauMarkt selbst voraussetzt, die derzeit noch nicht gegeben ist. Um dennoch das System demonstrieren und testen zu können, wird jeweils ein bestimmtes Marktareal ausgewählt, welches der Roboter zunächst kartiert. Parallel dazu erfolgt der Eintrag der Position definierter Artikel bzw. Warengruppen aus diesem Marktareal in eine vorliegende metrische Bauplan-Karte des Marktes, die die Anordnung der Regalreihen, die Position der Informationsstände etc. enthält. Nachdem die erlernte Karte vorliegt, wird diese manuell mit der Bauplan-Karte abgeglichen. Damit ist die Information über die Artikelstandorte in der vom Roboter erlernten Karte, die für die Navigationsaufgaben genutzt wird, verfügbar. Wählt der Kunde einen bestimmten Artikel im grafischen Interaktionstool aus, wird die entsprechende Position dieses Artikels als Zielposition definiert und an das Modul zur Pfadplanung weitergegeben.

Alle weiteren im Abschnitt 6.3 skizzierten Serviceleistungen, wie die Einblendung eines te-

lepräsenten Fachberaters oder die Präsentation von aktuellen Sonderangeboten, existieren derzeit lediglich auf Konzeptebene und sollen im weiteren Verlauf des Gesamtvorhabens Intelligenter Shopping-Assistent Schritt für Schritt in Angriff genommen werden. Betont werden muss an dieser Stelle, dass diese deutlich über die reine Tourguide-Funktion hinausgehenden Serviceleistungen eine entsprechende technische Infrastruktur im BauMarkt selbst voraussetzen, die momentan noch nicht gegeben ist.

Kapitel 10

Zusammenfassung und Ausblick

Zum Abschluss der Arbeit erfolgt nun in Abschnitt 10.1 eine Zusammenfassung der erreichten Ergebnisse, die wiederum nach den einzelnen Schwerpunkten bzw. Teilkomponenten des Gesamtsystems Shopping-Assistent untergliedert ist. Insbesondere wird der Frage nachgegangen, was die wesentlichen Ergebnisse und Erfahrungen sind, die sich aus den bisherigen Arbeiten ableiten und möglichst auch auf weitere Servicerobotik-Szenarien übertragen lassen. Abschnitt 10.2 diskutiert dann derzeit noch offene Probleme und aktuell laufende Forschungsarbeiten am Fachgebiet.

10.1 Zusammenfassung und Wertung der Ergebnisse

Im gesamten Verlauf der bisherigen Arbeiten erfolgte stets eine konsequente Orientierung an dem konkreten BauMarkt-Szenario. Welche Konsequenzen hatte das?

- Das BauMarkt-Szenario bietet an Schwierigkeiten nahezu alles, was man sich in einer Realwelt-Anwendung vorstellen kann und bietet somit eine sehr gute Testumgebung für die Beurteilung sowohl der Verfahren, die im Rahmen der Arbeit vorgestellt wurden als auch der Verfahren, die noch umgesetzt werden müssen.
- Es bestehen kaum Möglichkeiten, die Komplexität der Umgebung einzugrenzen, was mit dem Zwang einhergeht, Methoden zu entwickeln, die unter diesen Bedingungen robust funktionieren bzw. einsetzbar sind.
- Dennoch wurde versucht, möglichst viele Heuristiken, ohne die man momentan nicht auskommt, in die Entwicklung der verschiedenen Verfahren mit einzubeziehen. Allerdings erschwert gerade die genannte Komplexität der Einsatzumgebung auch oft den Einsatz von Heuristiken.
- Bei der Konzeption der verschiedenen Teilsysteme wurde sich konsequent an den bislang aus der Literatur bekannten Verfahren orientiert, die derzeit aus methodischer Sicht in Realwelt-Anwendungen sinnvoll und robust umsetzbar sind.

10.1.1 Navigation

Im Gesamtkomplex Navigation wurden zunächst aus der Literatur bekannte Verfahren aufgegriffen und reimplementiert. Aus den konkreten Eigenschaften der Einsatzumgebung ergab sich die Notwendigkeit zur Entwicklung und Integration von Methoden zur visuellen Informationsverarbeitung. Folgende Aspekte lassen sich zusammenfassen:

- Die entwickelte visuell-basierte Odometriekorrektur nutzt die absolut konstant bleibende Struktur des Fußbodens, um den Aufbau einer probabilistischen Umgebungskarte zu erleichtern. Dies stellt aber nur insofern ein generisch anwendbares Verfahren dar, als dass hier versucht wurde, sich an möglichst invarianten Gegebenheiten der Einsatzumgebung zu orientieren.
- Eine Verbesserung der Hinderniswahrnehmung bzw. Hindernisvermeidung für die lokale Navigation (egozentrische Umgebungsrepräsentation) kann mit dem vorgestellten visuellen Verfahren erreicht werden. Allerdings besteht hier neben der Kombination mit der sonarbasierten Methode angesichts der komplexen Hinderniskonfigurationen vermutlich noch weiterer Entwicklungsbedarf.
- Die Erweiterung des sonarbasieret erlernten globalen Umgebungsmodells um visuelle Ansichten und deren Nutzung für die robuste kontinuierliche Schätzung des Roboterzustands kann ausgehend von den bislang vorliegenden experimentellen Ergebnissen zur visuellen Selbstlokalisierung als sehr erfolgversprechend eingeschätzt werden. Der Vorteil dieser Vorgehensweise liegt in erster Linie in der gegebenen Stabilität der visuellen Ansichten bzw. der daraus gewonnen merkmalsbasierten Beschreibung begründet, die die im BauMarkt ständig auftretenden lokalen Veränderungen (Personen, unterschiedlicher Füllungsgrad der Regale, lokale Platzierung von Waren) sehr gut tolerieren kann.
- Beide letztgenannten Verfahren sind im Vergleich zur visuellen Odometriekorrektur eher generischer Natur und lassen sich auf andere Szenarien übertragen.
- Ebenfalls gut verallgemeinerbar ist der Einsatz multimodaler Informationsverarbeitung sowie die kombinierte Verwendung alternativer Methoden bei der Auswertung visueller Informationen.

10.1.2 Mensch-Roboter-Interaktion

Das Erreichen einer intuitiven und für den menschlichen Interaktionspartner möglichst natürlich wirkenden MRI stellte ein zentrales Anliegen der Arbeit dar. Auch hier bestand von Beginn an das Ziel darin, robuste Verfahren zu entwickeln, die den Anforderungen des BauMarkt-Szenarios gerecht werden können. In diesem Kontext lassen sich folgende Kernpunkte resümieren:

- Im Gesamtkomplex MRI ergibt sich die Notwendigkeit für den Einsatz visuell-basierter Methoden aus der Tatsache, dass nur über diesen Kanal genügend Informationen über den Interaktionsstatus des menschlichen Interaktionspartners gewonnen werden können.

- Die sichere und gleichzeitig spezifische Wahrnehmung von Personen stellt in Umgebungen mit einer der BauMarkt-Umgebung vergleichbaren Komplexität sicherlich eine Herausforderung dar, weshalb diesem Aspekt besondere Aufmerksamkeit gewidmet wurde. Das vorgestellte visuell-basierte System kann diese Aufgabe robust lösen. Aus der Literatur bekannte laserbasierte Verfahren waren mit Hilfe der vorhandenen Sonarsensoren aufgrund der zu geringen Entfernungs- und Winkelauflösung und des Rauschverhaltens der Sensoren nicht übertragbar. Diese können zwar mit dem demnächst verfügbaren Laserscanner in das Gesamtsystem mit integriert werden und würden so eine Ergänzung, jedoch keinen Ersatz der visuellen Methoden darstellen.
- Das Modul zum Personentracking nutzt einen multimodalen Ansatz und kombiniert visuelle Information mit Sonarinformation. Damit konnte Robustheit des Trackingprozesses erheblich gesteigert werden, was insbesondere für den Fall, dass sich sowohl Roboter als auch Target bewegen, von Bedeutung ist.
- Hervorgehoben werden soll weiterhin die vorgeschlagene Einbettung klassischer Interaktionsformen in ein natürlich und intuitiv verständlich wirkendes Gesamtverhalten des Serviceroboters, da zum aktuellen Stand der Technik klassische Eingabegeräte wie Tastatur oder Touchscreen auch auf absehbare Zeit nicht völlig durch die natürlichen Kommunikationskanäle Bild und Sprache ersetzt werden können.
- Generell muss eingeschätzt werden, dass bezüglich der Mensch-Roboter-Interaktion derzeit noch die größte Herausforderung auf der Methodenseite besteht.

10.1.3 Steuerarchitektur

Die in der Arbeit präsentierten experimentellen Ergebnisse demonstrieren den aktuellen Stand der Arbeiten, beziehen sich jedoch vorrangig auf die verschiedenen Teilmodule und weniger auf das Zusammenspiel aller Komponenten, da die Konzipierung und Umsetzung einer flexiblen Steuerarchitektur erst Gegenstand aktueller Forschungsarbeiten am Fachgebiet Neuroinformatik ist.

Mit der bislang realisierten Gesamtablaufsteuerung konnte jedoch im Rahmen von Vor-Ort-Experimenten zumindest die prinzipielle Funktionsfähigkeit der entwickelten Teilmodule nachgewiesen werden.

10.1.4 Serviceleistungen

Der Shopping-Assistent realisiert, wie das in der Einleitung zu den Kapiteln 3 und 4 und in Abschnitt 9.3 bereits herausgearbeitet wurde, zum aktuellen Entwicklungsstand eine reine Tourguide-Funktion. Die robuste Umsetzung dieser Funktion bildet die Voraussetzung dafür, dass über die Einrichtung einer entsprechenden technischen Infrastruktur im BauMarkt nachgedacht werden kann, um die in Abschnitt 6.3 diskutierten Servicefunktionen Schritt für Schritt umzusetzen. Dieser Stand wurde bislang bei allen Fortschritten in der Entwicklung der erforderlichen Teilmethoden noch nicht erreicht. Gründe aus methodischer Sicht sind:

- Probleme hinsichtlich der robusten Hinderniswahrnehmung, die mit dem in Abschnitt 7.1.3 beschriebenen visuellen Verfahren und dessen gemeinsamer Nutzung mit der sonarbasierten Hindernisdetektion entschärft werden sollen
- die noch fehlende bzw. in der Entwicklung befindliche Steuerarchitektur zur Koordination aller Teilmodule

Als Gründe aus praktischer Sicht sind folgende Aspekte zu nennen:

- Die am Gesamtprojekt Shopping-Assistent beteiligte Arbeitsgruppe mit derzeit 3 wissenschaftlichen Mitarbeitern ist, selbst wenn man die Unterstützung durch in verschiedene Themen einbezogene Studenten berücksichtigt, für die zügige Bewältigung aller anstehenden Aufgaben fast als unterkritisch zu betrachten.
- Dies schlägt sich hinsichtlich der pro Zeiteinheit vorzeigbaren Projektfortschritte, der Durchführung notwendiger Vor-Ort-Experimente etc. entsprechend nieder.

Dennoch wird angestrebt, das Gesamtsystem Shopping-Assistent soweit weiterzuentwickeln, dass mit dem zweiten Quartal des kommenden Jahres mit ersten realen Feldtests in der BauMarkt-Umgebung begonnen werden kann.

10.1.5 Übergreifende Aspekte

Ganz am Ende soll der Frage nachgegangen werden, was die wesentlichen Ergebnisse und Erfahrungen sind, die sich aus den bisherigen Arbeiten ableiten und möglichst auch auf weitere Servicerobotik-Szenarien übertragen lassen:

- Visuelle Verfahren, unabhängig davon, ob für die Navigation oder für die Mensch-Roboter-Interaktion, stellen für eine derartige Einsatzumgebung trotz aller erreichten Ergebnisse nach wie vor eine große Herausforderung dar. Dennoch sind wir (der Autor und die Kollegen der Arbeitsgruppe) überzeugt davon, dass nur durch die Einbeziehung visueller Information beide genannten Problemkreise langfristig robust, was die Navigation anbetrifft, und genügend intuitiv und natürlich, was die Interaktion anbetrifft, umgesetzt werden können. Insbesondere in diesem Kontext konnten mit der vorliegenden Arbeit durch eine Reihe von Ansätzen Beiträge präsentiert werden.
- Aus methodischer Sicht haben sich insbesondere die Module als sehr leistungsfähig erwiesen, die konsequent probabilistisch arbeiten und damit die in realen Szenarien mit realen Robotern zwangsläufig immer auftretenden Unsicherheiten geeignet modellieren. Allerdings gilt gerade für das konkrete BauMarkt-Szenario ebenso wie für eine Vielzahl anderer, ähnlich gelagerter Applikationen, dass auch der probabilistische Modellierungsrahmen dann keine Gewähr für eine robuste Funktionalität bietet, wenn die notwendigen sensorischen Beobachtungen nicht in entsprechend robuster und reproduzierbarer Weise gewonnen werden können. Dies gilt wiederum sowohl für alle Aspekte der Navigation ebenso wie für den Problemkreis MRI.

- Realwelt-Applikationen, wie sie auch das BauMarkt-Szenario darstellt, gebieten den Einsatz multimodaler Verfahren, wie das beispielsweise im Rahmen der lokalen und globalen Navigation oder beim Personentracking vorgestellt wurde. Erreicht die Einsatzumgebung eine bestimmte Komplexität bzw. lässt sich diese Komplexität nicht durch geeignete Maßnahmen einschränken, reicht oft der Einsatz einer Sensormodalität nicht mehr aus, um die notwendige Robustheit der Methoden zu erreichen.
- Die Einbettung klassischer Formen der MRI in ein intuitiv und natürlich wirkendes Gesamtkonzept zur Interaktion zwischen Serviceroboter und menschlichem Benutzer konnte soweit vorangetrieben werden, dass dieses Konzept in dieser Form auch auf eine Vielzahl weiterer Applikationen übertragen werden könnte. Angesichts des aktuellen Entwicklungsstandes kann auch auf absehbare Zeit nicht davon ausgegangen werden, dass klassische Formen der Mensch-Maschine-Interaktion in vollem Umfang durch die ausschließliche Nutzung des visuellen und akustischen Kanals ersetzt werden können.
- Setzt man eine kommerzielle Plattform für ein solches Projekt ein, so ergeben sich zwangsläufig Beschränkungen durch die Art und die Anordnung der vorhandenen Sensorik. Hier lässt sich allerdings nur Abhilfe schaffen, wenn man ein eigenes Robotersystem konzipiert und technisch umsetzt. Einige der beschriebenen Verfahren, wie z.B. die visuelle Hindernisvermeidung, wurden auch deshalb entwickelt, weil die Abstandssensoren eben nur ebenenweise messen können. Wären diese nach oben bzw. nach unten geneigt, wäre eine viel bessere Abdeckung der Umgebung möglich, die das vorgestellte visuelle Verfahren unter Umständen überflüssig machen würde, was jedoch in keiner Weise die Bedeutung der visuellen Informationsverarbeitung in Frage stellt.

10.2 Offene Probleme und Ausblick

Offene Probleme und damit auch Gegenstand der aktuellen Forschungsarbeiten am Fachgebiet stellen die folgenden Punkte dar:

- Ein Defizit besteht sicherlich in den an verschiedenen Stellen noch unzureichenden experimentellen Untersuchungen, was letztlich auch die Wertung der bisher erreichten Ergebnisse etwas erschwert.
- Einen Schwerpunkt der weiteren Arbeit muss deshalb die Durchführung von experimentellen Untersuchungen zur Interaktion von „realen“ BauMarkt-Kunden mit dem Shopping-Assistenten bilden. Dafür fehlen aber hinsichtlich der robusten Navigation, insbesondere was die sichere Hindernisvermeidung anbetrifft, noch einige Voraussetzungen, die mit der Einbeziehung des vorgeschlagenen Verfahrens zur visuell-basierten Hindernisvermeidung zumindest teilweise realisiert werden sollen.
- Gerade diese Experimente unter realen Einsatzbedingungen stellen eine einmalige Chance dar, in Richtung MRI entscheidende Fortschritte zu erzielen. Hier spielen dann wieder die Lernaspekte eine zentrale Rolle, also die Frage, wie der Roboter seine Aktionen aus dem verfügbaren Aktionsraum situationsabhängig so auszuwählen lernt, dass ein wie auch immer definierter „Interaktionserfolg“ erreicht wird. Dabei ergeben sich Bezüge zum

MINERVA-Projekt, wo erste experimentelle Untersuchungen dazu bereits durchgeführt wurden.

- Um in der BauMarkt-Umgebung robust und effizient navigieren zu können, sind Methoden zur kontinuierlichen Adaption der globalen Umgebungskarte Gegenstand aktueller Forschungsarbeiten, die über das momentan verwendete probabilistische Konzept zum Kartenaufbau hinausgehen bzw. dieses erweitern. So muss der Shopping-Assistent unterscheiden können, ob es sich bei einem Hindernis, das in der erlernten globalen Karte noch nicht verzeichnet ist, um Personen, um temporär platzierte Waren oder um eine dauerhafte Veränderung der globalen Karte handelt. In die Modellierung der Einsatzumgebung sollten weiterhin Informationen über die Belebtheit bestimmter Passagen oder typische Routen von Kunden während des Einkaufs einbezogen werden.
- Das in dieser Arbeit nicht behandelte Konzept der Web-Interfaces für Serviceroboter wird perspektivisch in zweierlei Hinsicht enorm an Bedeutung gewinnen: zum einen, um als eine mögliche Serviceleistung auch den Einkauf außerhalb der regulären Öffnungszeiten des Marktes zu ermöglichen, zum anderen für eine möglichst effiziente Fernwartung des Serviceroboters, um zu vermeiden, dass der Entwickler bzw. Systemingenieur ständig vor Ort zur Verfügung stehen muss.
- Lernen, also die Anpassung an wechselnde Umgebungsbedingungen und Situationen, wird bei zukünftigen Applikationen von Servicerobotern neben der intuitiv verständlichen Mensch-Roboter-Interaktion maßgeblich den Gesamterfolg determinieren. Je komplexer die Gesamtaufgabe des Roboters und je dynamischer sein Einsatzfeld, umso mehr wird die Lernfähigkeit in den Mittelpunkt des Interesses rücken. Reichen bei einfachen Aufgaben in einfachen Umgebungen noch per Hand vollständig vorprogrammierte Abläufe aus, so steigt mit zunehmender Komplexität die Gefahr, dass nicht alle Abhängigkeiten und möglicherweise auftretenden Situationen vorab vollständig erfasst und bei der Programmierung entsprechend berücksichtigt wurden. Verfügt der Roboter allerdings über Mechanismen, die ihm eine flexible Adaption an veränderliche Verhältnisse ermöglichen, so sollte dies kein prinzipielles Problem mehr darstellen. Dieser Aspekt wird sowohl beim Entwurf der Steuerarchitektur als auch insbesondere im Kontext einer anzustrebenden Dialogmodellierung mit berücksichtigt.

Abschließend noch ein Gedanke, der mit dem Slogan „Fluch und Segen des BauMarkt-Szenarios“ charakterisiert werden soll: Die bisherige Erfahrung bei der Präsentation bzw. Publikation unserer aktuellen Ergebnisse zeigte stets, dass es hinsichtlich der Sinnfälligkeit der angestrebten Serviceroboter-Anwendung kaum Probleme gibt. Das Messen an diesem Szenario wird bei allen Misserfolgen, die nicht zu vermeiden sind, auch weiterhin Ansporn und Triebfeder sein.

Literaturverzeichnis

- [AFD⁺99] Ahlrichs, U., Fischer, J., Denzler, J., Drexler, C., Niemann, H., Nöth, E., and Paulus, D. (1999). *Knowledge Based Image and Speech Analysis for Service Robots*. In *Proceedings Integration of Speech and Image Understanding*, 21–47. IEEE Computer Society.
- [AMH⁺96] Asoh, H., Motomura, Y., Hara, I., Akaho, S., Hayamizu, S., and Matsui, T. (1996). *Combining probabilistic map and dialog for robust life-long office navigation*. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, 807–812.
- [AN98] Aust, H. and Ney, H. (1998). *Evaluating Dialog Systems used in the Real World*. In *IEEE International Conference on Acoustics, Signal and Speech Processing (ICASSP)*, vol. 2, 1053–1056.
- [Ark98] Arkin, R.C. (1998). *Behavior-Based Robotics*. MIT Press.
- [AWB87] Aloimonos, J.Y., Weiss, I., and Bandopadhyay, A. (1987). *Active Vision*. *International Journal on Computer Vision*, 333–356.
- [Baj88] Bajcsy, R. (1988). *Active Perception*. *IEEE Proceedings*, 76(8):996–1006.
- [Bal91] Ballard, D. (1991). *Animate Vision*. *Artificial Intelligence*, 48(1):1–27.
- [Bar92] Barto, A. (1992). *Reinforcement learning and adaptive critic methods*. In D. White and S. Sofge (eds.), *Handbook of intelligent control: Neural, fuzzy and adaptive approaches*, 469–491. New York: Van Nostrand Reinhold.
- [BBB⁺97] Brakensiek, A., Braumann, U.-D., Böhme, H.-J., Rieck, C., and Gross, H.-M. (1997). *Farb- und strukturbasierte, neuronale Verfahren zur Lokalisation von Gesichtern in Real-World-Szenen*. In *Mustererkennung 1997, 19. DAGM-Symposium Braunschweig*, Informatik aktuell, 113–120. Springer Verlag.
- [BBB⁺98a] Boehme, H.-J., Braumann, U.-D., Brakensiek, A., Krabbes, M., Corradini, A., and Gross, H.-M. (1998). *User Localisation for Visually-based Human-Machine-Interaction*. In *International Conference on Automatic Face- and Gesture Recognition (FG'98)*, 486–491. IEEE Computer Society Press.
- [BBB⁺98b] Boehme, H.-J., Braumann, U.-D., Brakensiek, A., Krabbes, M., Corradini, A., and Gross, H.-M. (1998). *Neural Networks for Gesture-based Remote Control of*

- a Mobile Robot*. In *International Joint Conference on Neural Networks*, vol. 1, 372–377. IEEE Computer Society Press.
- [BBC⁺95] Buhmann, J., Burgard, W., Cremers, A.B., Fox, D., Hofmann, T., Schneider, F., Strikos, J., and Thrun, S. (1995). *The mobile robot Rhino*. *AI Magazine*, 267(2).
- [BBCG99] Boehme, H.-J., Braumann, U.-D., Corradini, A., and Gross, H.-M. (1999). *Person Localization & Posture Recognition for Human-Robot Interaction*. In *GW'99 - The 3rd Gesture Workshop, Gif-sur-Yvette, France*, 105–116. Springer, Lecture Notes in Artificial Intelligence 1739.
- [BBD⁺97] Bergener, T., Bruckhoff, C., Dahm, P., Janßen, H., Joublin, F., and Menzner, R. (1997). *Arnold: An Anthropomorphic Autonomous Robot for Human Environments*. In *SOAVE'97 – Selbstorganisation von adaptivem Verhalten*, 25–34. VDI Verlag.
- [BBD⁺99] Bergener, T., Bruckhoff, C., Dahm, P., Janßen, H., Joublin, F., Menzner, R., Steinhage, A., and von Seelen, W. (1999). *Complex behavior by means of dynamical systems for an anthropomorphic robot*. *Neural Networks, 1999 Special Issue "Organisation of Computation in Brain-like Systems"*, 12:1087–1099.
- [BBM⁺98] Brooks, R.A., Breazeal, C., Marjanovic, M., Scassellati, B., and Williamson, M. (1998). *The Cog Project: Building a Humanoid Robot*. In C. Nehaniv (ed.), *Computation for Metaphors, Analogy and Agents*, vol. 1562 of *Springer Lecture Notes in Artificial Intelligence*, 291–304. Springer-Verlag.
- [BCF⁺98] Burgard, W., Cremers, A., Fox, D., Lakemeyer, G., Hähnel, D., Schulz, D., Steiner, W., and Thrun, S. (1998). *The interactive museum tour-guide robot*. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*.
- [BcJ⁺00] Bredenfeld, A., christaller, T., Jaeger, H., Kobialka, H.-U., and Schöll, P. (2000). *Robot Behavior Design Using Dual Dynamics*. Tech. rep., GMD - Forschungszentrum Informationstechnik GmbH. GMD Report 117.
- [BEF96] Borenstein, J., Everett, B., and Feng, L. (1996). *Navigating Mobile Robots: Systems and Techniques*. A. K. Peters, Ltd., Wellesley, MA.
- [BFG⁺00] Baltus, G., Fox, D., Gemperle, F., Goetz, J., Hirsch, T., Magaritis, D., Montemerlo, M., Pineau, J., Roy, N., Schulte, J., and Thrun, S. (2000). *Towards personal service robots for the elderly*. In Horswill, I. and Balch, T. (eds.), *Proceedings of the Workshop on Interactive Robotics and Entertainment, CMU*.
- [BFHS96] Burgard, W., Fox, D., Hennig, D., and Schmidt, T. (1996). *Estimating the absolute position of a mobile robot using position probability grids*. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, 896–901.

- [BFJ⁺99] Burgard, W., Fox, D., Jans, H., Matenar, C., and Thrun, S. (1999). *Sonar-based mapping of large-scale mobile robot environments using em*. In *Proceedings of the International Conference on Machine Learning (ICML)*.
- [BG99a] Bischoff, R. and Graefe, V. (1999). *Integrating Vision, Touch and Natural Language in the Control of a Situation-Oriented Behavior-Based Humanoid Robot*. In *IEEE International Conference on Systems, Man, and Cybernetics*, vol. II, 999–1004.
- [BG99b] Böhme, H.-J. and Gross, H.-M. (1999). *Ein Interaktives Mobiles Service-System für den Baumarkt*. In *14. Fachgespräch Autonome Mobile Systeme (AMS'99), München*, 344–353. Springer.
- [BGG98] Bischoff, R., Graefe, V., and Graßnick, R. (1998). *Serviceroboter der Zukunft. Spektrum der Wissenschaft. Dossier 4/98*.
- [BGG⁺99] Brafford, A., Gherbi, R., Gibet, S., Richardson, J., and Teil, D. (eds.) (1999). *Gesture-Based Communication in Human-Computer Interaction*. Lecture Notes in Computer Science. Springer.
- [BGKW00] Böhme, H.-J., Gross, H.-M., Key, J., and Wilhelm, T. (2000). *PERSES - a Personal SEvice System*. In *Proceedings SOAVE'2000 - Selbstorganisation von adaptivem Verhalten*, vol. 643 of *Fortschrittsberichte, Reihe 10*, 57–66. VDI Verlag.
- [BH00] Briechle, K. and Hanebeck, U. (2000). *Ein Zwei-Karten-Verfahren zur gleichzeitigen Kartographierung und Lokalisierung*. In *16. Fachgespräch Autonome Mobile Systeme (AMS 2000)*, 165–174. Springer.
- [Bis98] Bischoff, R. (1998). *Neueste Entwicklungen auf dem Gebiet der Servicerobotik. Künstliche Intelligenz*, 3:67–71.
- [Bis00a] Bischoff, R. (2000). *Recent Advances in the Development of the Humanoid Service Robot HERMES*. In *3rd EUREL Workshop and Masterclass - European Advanced Robotics Systems Development, Salford, U.K.*, vol. I, 125–134.
- [Bis00b] Bischoff, R. (2000). *Towards the Development of "Plug-and-Play" Personal Robots*. In *IEEE-RAS International Conference on Humanoid Robots*.
- [BK89] Borenstein, J. and Koren, Y. (1989). *Real-time Obstacle Avoidance for Fast Mobile Robots*. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5):1179–1187.
- [BK91] Borenstein, J. and Koren, Y. (1991). *The Vector Field Histogram – Fast Obstacle-Avoidance for Mobile Robots*. *IEEE Journal of Robotics and Automation*, 7(3):278–288.
- [BKB⁺98] Böhme, H.-J., Krabbes, M., Brakensiek, A., Braumann, U.-D., Corradini, A., and Gross, H.-M. (1998). *Gestenbasierte Interaktion und Kommunikation mit*

- dem mobilen Robotersystem MILVA*. Tech. rep., Technische Universität Ilmenau, Fachgebiet Neuroinformatik. Abschlussbericht TMWFK-Projekt GESTIK, Schriftenreihe des Fachgebiets Neuroinformatik, Report 1/98, ISSN 0945-7518.
- [BKOS97] Berg, M. de, Kreveld, M. van, Overmars, M., and Schwarzkopf, O. (1997). *Computational Geometry: Algorithms and Applications*. Springer.
- [BKW97] Bonasso, R.P., Kortenkamp, D., and Whitney, T. (1997). *Using a Robot Control Architecture to Automate Space Shuttle Operations*. In *9th Conference on Innovative Applications IAAI'97*.
- [Ble01] Bleul, A. (2001). *Vorsicht vase! was haushaltsroboter so können*. *c't*, 2001(17):74–77.
- [BM00] Bitterling, J. and Mertsching, B. (2000). *Sichtgestützte navigation für einen dominospielenden autonomen mobilen roboter*. In *Proceedings SOAVE'2000 - Selbstorganisation von adaptivem Verhalten*, vol. 643 of *Fortschrittsberichte, Reihe 10*, 22–30. VDI Verlag.
- [BM01a] Billard, A. and Mataric, M.J. (2001). *Learning human arm movements by imitation: Evaluation of a biologically inspired connectionist architecture*. *Robotics and Autonomous Systems*. To appear in 2001.
- [BM01b] Bitterling, J. and Mertsching, B. (2001). *A vision-guided robot for manipulation of domino tokens*. In *17. Fachgespräch Autonome Mobile Systeme (AMS 2001)*, vol. 1, 99–103.
- [BMB01] BMBF (ed.) (2001). *Lead Projects Human-Computer-Interaction*. DLR Deutsches Zentrum für Luft- und Raumfahrt e.V. Projektträger des BMBF für Informationstechnik, International Status Conference, Saarbrücken.
- [Boh94] Bohrer, S. (1994). *Visual obstacle recognition by analysis of optical flow in invers-perspective scenes*. *Fortschrittsberichte 10*. VDI Verlag.
- [Bra01] Braumann, U.-D. (2001). *Multi-Cue-Ansatz für ein dynamisches Auffälligkeitssystem zur visuellen Personenlokalisierung*. Ph.D. thesis, Technische Universität Ilmenau.
- [Bro86] Brooks, R.A. (1986). *A robust layered control system for a mobile robot*. *Journal of Robotics and Automation*, 2.
- [Bro99] Brooks, R.A. (1999). *Cambrian Intelligence - the early history of new AI*. The MIT Press.
- [BS98] Bergener, T. and Steinhage, A. (1998). *An Architecture for Behavioral Organization using Dynamical Systems*. In *3rd German Workshop on Artificial Life, GWAL '98*.

- [BTN00] Baratoff, G., Toepfer, C., and Neumann, H. (2000). *Combined space-variant maps for optical flow based navigation. Biological Cybernetics, special issue on Navigation in Biological and Artificial Systems*, 199–209.
- [BWK⁺01a] Boehme, H.-J., Wilhelm, T., Key, J., Schroeter, Ch., Hempel, T., and Gross, H.-M. (2001). *An Approach to Multimodal Human-Machine Interaktion for Intelligent Service Robots*. In *EUROBOT'01 - the fourth Euromicro Workshop on Advanced Mobile Robots*, vol. 86 of *Lund University Cognitive Studies*, 17–24. Lund University.
- [BWK⁺01b] Böhme, H.-J., Wilhelm, T., Key, J., Schröter, Ch., Hempel, T., and Gross, H.-M. (2001). *Multimodale Mensch-Maschine-Interaktion für Servicerobotik-Systeme*. In *16. Fachgespräch Autonome Mobile Systeme (AMS'01)*, Stuttgart, 113–119. Springer. To appear.
- [CBG00] Corradini, A., Boehme, H.-J., and Gross, H.-M. (2000). *A Hybrid Stochastic-Connectionist Approach to Gesture Recognition. International Journal on Artificial Intelligence Tools*, 2000(9):177–204.
- [Chr96] Christaller, Th. (1996). *Von Artificial Intelligence zu Artificial Life und wieder zurück. Künstliche Intelligenz*, 4:41–44.
- [CHS00] Christensen, H.I., Hüttenrauch, H., and Severinson-Eklundh, K. (2000). *Human-Robot Interaction in Service Robots*. In *Robotik 2000, Berlin*, VDI Berichte Nr. 1552, 315–323. VDI Verlag.
- [Cor00] Corradini, A. (2000). *A Hybrid Stochastic-Connectionist Approach to Gesture Recognition*. Ph.D. thesis, Technische Universität Ilmenau.
- [Dav96] Davis, J.W. (1996). *Appearance-Based Motion Recognition of Human Actions*. Tech. rep., M.I.T. Media Lab Perceptual Computation Section. TR 387.
- [DB97] Davis, J.W. and Bobick, A.F. (1997). *The Representation and Recognition of Action Using Temporal Templates*. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. M.I.T. Media Lab Techreport TR 402.
- [DB98] Davis, J.W. and Bobick, A.F. (1998). *SIDEshow: A Silhouette-based Interactive Dual-screen Environment*. Tech. rep., M.I.T. Media Lab Perceptual Computation Section. TR 457.
- [DLR77] Dempster, A., Laird, A., and Rubin, D. (1977). *Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society*, 39:1–38.
- [Edl97] Edlinger, T. (1997). *Hierarchische Steuerung für einen mobilen Roboter zur autonomen Erkundung seiner Einsatzumgebung*. VDI Verlag GmbH.
- [Elf89] Elfes, A. (1989). *Using occupancy grids for mobile robot perception and navigation*. IEEE Computer.

- [EP95] Essa, I.A. and Pentland, A.P. (1995). *Facial Expression Recognition using a Dynamic Model and Motion Energy*. In *International Conference on Computer Vision*. M.I.T. Media Lab Techreport TR 307.
- [FBR98] Fjeld, M., Bichsel, M., and Rauterberg, M. (1998). *BUILD-IT: An Intuitive Design Tool Based on Direct Object Manipulation*. In *Gesture and Sign-Language in Human-Computer Interaction*, 297–308.
- [FBT98] Fox, D., Burgard, W., and Thrun, S. (1998). *A Hybrid Collision Avoidance Method For Mobile Robots*. In *Proceedings of the IEEE International Conference on Robotics and Automation*.
- [FBT99] Fox, D., Burgard, W., and Thrun, S. (1999). *Markov Localization for Mobile Robots in Dynamic Environments*. *Journal of Artificial Intelligence Research*, 11:391–427.
- [FDBT99] Fox, D., Delleart, F., Burgard, W., and Thrun, S. (1999). *Monte Carlo Localization: Efficient Position Estimation for Mobile Robots*. In *Proceedings 16th National Conference on Artificial Intelligence (AAAI-99)*.
- [Fey00] Feyrer, S. (2000). *Detektion, Lokalisierung und Verfolgung von Personen mit einem mobilen Serviceroboter*. Ph.D. thesis, Eberhard-Karls-Universität Tübingen.
- [Fis99] Fischer, C. (1999). *Eine fortgeschrittene Bedienschnittstelle für einen semiautonomem mobilen Serviceroboter*. VDI Verlag. Dissertationsschrift, Technische Universität München.
- [FL90] Fahlman, S.E. and Lebiere, Ch. (1990). *The cascade-correlation learning architecture*. In *Advances in Neural Information Processing Systems 2*, 524–532. Morgan Kaufmann Publishers, Inc.
- [FM00] Franz, M.O. and Mallot, H.A. (2000). *Biomimetic robot navigation*. *Robotics and Autonomous Systems*, 30(1-2):133–153.
- [Fox98] Fox, D. (1998). *Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Navigation*. Ph.D. thesis, Institute of Computer Science III, University of Bonn.
- [Fra94] Fraunhofer-Institut für Produktionstechnik und Automatisierung (1994). *Service-Roboter - ein Beitrag zur Innovation im Dienstleistungswesen*. Studie im Auftrag des BMFT.
- [Fri98] Fritzsche, B. (1998). *Vektorbasierte Neuronale Netze*. Shaker Verlag.
- [FZ98] Feyrer, S. and Zell, A. (1998). *Ein integrierter Ansatz zur Lokalisierung von Personen in Bildfolgen*. In *Dynamische Perzeption*, vol. 8 of *Proceedings in Artificial Intelligence*, 183–190. Infix Verlag.
- [FZ99a] Feyrer, S. and Zell, A. (1999). *Personentracking mit einer mobilen Roboterplattform unter Verwendung eines multimodalen Detektionsansatzes*. *KI*, 1:7–12.

- [FZ99b] Feyrer, S. and Zell, A. (1999). *Tracking and Pursuing Persons with a Mobile Robot*. In *International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems (RATFG-RTS '99)*, 83–88.
- [FZ00] Feyrer, S. and Zell, A. (2000). *Robust Real-Time Pursuit of Persons with a Mobile Robot Using Multisensor Fusion*. In *6th International Conference on Intelligent Autonomous Systems (IAS-6)*, 710–715.
- [GB00] Gross, H.-M. and Boehme, H.-J. (2000). *PERSES - a Vision-based Interactive Mobile Shopping Assistant*. In *IEEE Internat. Conference on Systems, Man and Cybernetics (IEEE-SMC 2000)*, 80–85. IEEE/Omnipress.
- [GB01] Gross, H.-M. and Boehme, H.-J. (2001). *A contribution to vision-based localization, tracking and navigation methods for an interactive mobile service-robot*. In *IEEE Intern. Conference on System, Man and Cybernetics (IEEE-SMC 2001), Tucson*. To appear.
- [GBTS00] Graf, B., Baum, W., Traub, A., and Schraft, R.D. (2000). *Konzeption dreier Roboter zur Unterhaltung der Besucher eines Museums*. In *VDI-Berichte 1552*, 529–536.
- [GH01] Graf, B. and Hägele, M. (2001). *Dependable Interaction with an Intelligent Home Care Robot*. In *Proceedings of the IEEE International Conference on Robotics and Automation*.
- [GHN⁺00] Green, A., Hüttenrauch, H., Norman, M., Oestreicher, L., and Severinson-Eklundh, K. (2000). *User Centered Design for Intelligent Service Robots*. In *IEEE International Workshop on Robot and Human Interactive Communication (ROMAN2000)*.
- [GHSS99] Gross, H.-M., Heinze, A., Seiler, T., and Stephan, V. (1999). *Generative Character of Perception: A Neural Architecture for Sensorimotor Anticipation*. *Neural Networks, 1999 Special Issue "Organisation of Computation in Brain-like Systems"*, 12:1101–1129.
- [GS01] Green, A. and Severinson-Eklundh, K. (2001). *Task-oriented Dialogue for CERO: a User-centered Approach*. In *IEEE International Workshop on Robot and Human Interactive Communication (ROMAN2001)*.
- [GSS98] Gross, H.-M., Stephan, V., and Seiler, T. (1998). *Neural Architecture for Sensorimotor Anticipation*. In *CYBERNETICS AND SYSTEMS'98, Proceedings of the 14. European Meeting on Cybernetics and Systems Research*, vol. 2, 593–98.
- [HE97] Harling, P.A. and Edwards, A.D.N. (eds.) (1997). *Progress in Gestural Interaction*. Springer.
- [Hem00] Hempel, T. (2000). *Koordination und Implementation von Personenlokalisierung, -verifikation und -tracking für mobile Service-Systeme*. Master's thesis, Technische Universität Ilmenau.

- [HG01] Heinze, A. and Gross, H.-M. (2001). *Anticipation-Based Control Architecture for a Mobile Robot*. In *Intern. Conference on Artificial Neural Networks (ICANN 2001)*. To appear.
- [HH01] Hanebeck, U. and Horn, J. (2001). *An Efficient Method for Simultaneous Map Building and Localization*. In *Proceedings of SPIE*, vol. 4385, Sensor Fusion: Architecture, Algorithms and Applications V, 233–243.
- [HNS01] Haegele, A., Neugebauer, J., and Schraft, R. D. (2001). *From Robots to Robots Assistants*. In *Proceedings of the 32th International Symposium on Robotics (ISR), Seoul*.
- [Hor93] Horswill, I. (1993). *Polly: A vision-based artificial agent*. In *Eleventh National Conference on Artificial Intelligence (AAAI-93)*. MIT Press.
- [Hor94a] Horswill, I. (1994). *Specialization of perceptual processes*. Tech. rep., MIT AI Lab, Cambridge, MA. AI TR-1511.
- [Hor94b] Horswill, I. (1994). *Visual Collision Avoidance by Segmentation*. In *IROS '94*, 902–909.
- [Hor95] Horswill, I. (1995). *Integrating Vision and Natural Language without Central Models*. In *AAAI Fall Symposium on Embodied Language and Action*.
- [Hor98] Horswill, I. (1998). *The Polly System*. In David Kortenkamp, R. Peter Bonasso, and Robin Murphy (eds.), *AI-based Mobile Robots: Case Studies of Successful Robot Systems*, 125–139. AAAI Press / The MIT Press.
- [htt00] <http://www.gruetter.de/zeitschriften/aq/ausgabe/automation-0300-s30.htm> (2000). *Von der Werkshalle ins Wohnzimmer - das Leben mit Servicerobotern*. *Automation & Qualität*.
- [htt01] <http://www.ipa.fhg.de/srdatabase/> (2001). *Database Servicerobots*.
- [IAY89] Iijima, J., Asaka, S., and Yuta, S. (1989). *Searching Unknown Environment by a Mobile Robot Using Range Sensor - An Algorithm and Experiment*. In *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, 46–53.
- [IB98] Isard, M. and Blake, A. (1998). *CONDENSATION – conditional density propagation for visual tracking*. *International Journal on Computer Vision*, 29(1):5–28.
- [IEE98] IEEE Computer Society (ed.) (1998). *3rd IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE Computer Society.
- [IEE00] IEEE Computer Society (ed.) (2000). *4th IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE Computer Society.
- [Jae97] Jaehne, B. (1997). *Practical Handbook on Image Processing for Scientific Applications*. CRC Press LLC.

- [JR94] Jolion, J.-M. and Rosenfeld, A. (1994). *A Pyramid Framework for Early Vision: Multiresolutional Computer Vision*. Kluwer Academic Publishers.
- [Kah96] Kahn, R. (1996). *Perseus: An Extensible Vision System for Human-Machine Interaction*. Ph.D. thesis, University of Chicago.
- [Kar00] Karlsson, J. (2000). *World Robotics 2000 - Statistics, Market Analysis, Forecasts, Case Studies and Profitability of Robot Investment*. United Nations Economic Commission for Europe (UN/ECE) and the International Federation of Robotics (IFR).
- [KB88] Kuipers, B.J. and Byun, Y.T. (1988). *A Robust, Qualitative Method for Robot Spatial Learning*. In *AAAI*, 774–779.
- [KBM98] Kortenkamp, D., Bonasso, R.P., and Murphy, R. (eds.) (1998). *Artificial Intelligence and Mobile Robots*. AAAI Press / MIT Press.
- [KBSG97a] Krabbes, M., Boehme, H.-J., Stephan, V., and Gross, H.-M. (1997). *Extension of the ALVINN-Architecture for Robust Visual Guidance of a Miniature Robot*. In *EUROBOT'97 - 2nd Euromicro Workshop on Advanced Mobile Robots*, 8–14. IEEE Computer Society Press.
- [KBSG97b] Krabbes, M., Boehme, H.-J., Stephan, V., and Gross, H.-M. (1997). *Handlungsorganisation intentionaler neuronaler Agenten*. In *SOAVE'97 - Selbstorganisation von adaptivem Verhalten*, vol. 663 of *Fortschrittsberichte, Reihe 8*, 35–44. VDI-Verlag.
- [KBSG97c] Krabbes, M., Boehme, H.-J., Stephan, V., and Gross, H.-M. (1997). *Handlungsorganisation mittels intentionaler neuronaler Agenten*. In *AMS'97 - 13. Fachgespräch Autonome Mobile Systeme*, 154–65. Springer-Verlag.
- [KGB01] König, A., Gross, H.-M., and Böhme, H.-J. (2001). *Visuell basierte Monte-Carlo Lokalisation für mobile Roboter mit omnidirektionalen Kameras*. Tech. rep., Technische Universität Ilmenau, Schriftenreihe des Fachgebiets Neuroinformatik. Report 2/01, ISSN 0945-7518.
- [KHH⁺01] Kristensen, S., Hansen, V., Horstmann, S., Klandt, J., Kondak, K., Lohnert, F., and Stopp, A. (2001). *Interaktives Lernen von Weltmodellen für einen Service-Roboter*. *Künstliche Intelligenz*, 1:52–59.
- [KKG00] König, A., Key, J., and Gross, H.-M. (2000). *Visuell basierte Monte-Carlo Lokalisation für mobile Roboter mit omnidirektionalen Kameras*. In *Proceedings SOAVE'2000*. VDI Verlag.
- [KLM96] Kaelbling, L.P., Littman, M.L., and Moore, A.W. (1996). *Reinforcement learning: A survey*. *Journal of AI Research*, 4:237–285.
- [Kni91] Knieriemer, T. (1991). *Autonome Mobile Roboter*. BI Wissenschaftsverlag.

- [KPBI96] Kawamura, K., Pack, T., Bishay, M., and Iskarous, M. (1996). *Design Philosophy for Service Robots*. *Journal of Robotics and Autonomous Systems*, 18:109–116.
- [KSE⁺00] Kraetzschmar, G., Sablatnög, S., Enderle, S., Utz, H., Simon, S., and Palm, G. (2000). *Integration of Multiple Representations and Navigation Concepts on Autonomous Mobile Robots*. In *SOAVE'2000 – Selbstorganisation von Adaptivem Verhalten*, 1–13. VDI-Verlag.
- [KWBG98] Krabbes, M., Weber, S., Boehme, H.-J., and Gross, H.-M. (1998). *Monokulare visuelle Hindernisdetektion auf Basis merkmalsbasierter Bildsegmentierung*. In *AMS'98 - 14. Fachgespräch Autonome Mobile Systeme*, Informatik aktuell, 85–92. Springer-Verlag.
- [Lay00] Lay, K. (2000). *MORPHA - Intelligente anthropomorphe Assistenzsysteme*. *it+ti - Informationstechnik und Technische Informatik*, 42(1):38–43.
- [LC99] Large, E. W. and Christensen, H. I. (1999). *Scaling the dynamic approach to path planning and control: Competition among behavioural constraints*. *International Journal of Robotics Research*, 18(1):37–58.
- [Lie00] Liebetrau, A. (2000). *Probabilistische Verfahren für die Roboternavigation*. Master's thesis, Technische Universität Ilmenau, Fachgebiet Neuroinformatik.
- [LL90] Levitt, T.S. and Lawton, D.T. (1990). *Qualitative navigation for mobile robots*. *Artificial Intelligence*, 44:305–360.
- [LLH⁺94] Lueth, T., Laengle, T., Herzog, G., Stopp, E., and Rembold, U. (1994). *KAN-TRA: Human-Machine Interaction for Intelligent Robots Using Natural Language*. Tech. rep., SFB 314: Künstliche Intelligenz – Wissensbasierte Systeme.
- [LS98] Liu, L. and Sclaroff, S. (1998). *Deformable Shape Detection and Description via Model-Based Region Grouping*. Tech. rep., Boston University. Computer Science Technical Report 98-017.
- [Mar82] Marr, D. (1982). *Vision*. Freeman, San Francisco.
- [May79] Maybeck, P. (1979). *Stochastic Models, Estimation, and Control*, vol. 1. Academic Press.
- [MB01] Martin, C. and Böhme, H.-J. (2001). *Monokulare Roboternavigation zur visuellen Hindernisvermeidung*. Tech. rep., Technische Universität Ilmenau, Schriftenreihe des Fachgebiets Neuroinformatik. Report 1/01, ISSN 0945-7518.
- [MLB91] Mallot, H.P., Bühlhoff, H.H., Little, J.J., and Bohrer, S. (1991). *Inverse Perspective Mapping Simplifies Optical Flow Computation and Obstacle Detection*. *Biological Cybernetics*, 64:177–185.
- [ME85] Moravec, H.P. and Elfes, A. (1985). *High resolution maps from wide angle sonar*. In *International Conference on Robotics and Automation (ICRA)*, 116–121.

- [Mit97] Mitchell, T.M. (1997). *Machine Learning*. McGraw-Hill.
- [MMH01] Maistros, G., Marom, Y., and Hayes, G. (2001). *Perception-Action Coupling via Imitation and Attention*. In *Proceedings of the AAAI Fall Symposium on Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems*. To appear.
- [Moe96] Moeller, R. (1996). *Wahrnehmung durch Vorhersage - Eine Konzeption der handlungsorientierten Wahrnehmung*. Ph.D. thesis, Technische Universität Ilmenau.
- [Mor88] Moravec, H.P. (1988). *Sensor fusion in certainty grids for mobile robots*. *AI Magazine*, 61–74.
- [MS99] Mertsching, B. and Schmalt, S. (1999). *Active Vision Systems*. In Jähne, B., Häußeker, H., and Geißler, P. (eds.), *Handbook of Computer Vision and Applications*, vol. 3, 197–219. Academic Press.
- [Nil69] Nilsson, N.J. (1969). *A Mobile Automaton: An Application of Artificial Intelligence Techniques*. In *First International Joint Conference on Artificial Intelligence (IJCAI), Washington D.C.*, 509–520.
- [NM01] Nicolescu, M. and Mataric, M.J. (2001). *Learning and Interacting in Human-Robot Domains*. *Special Issue of IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*. To appear.
- [NS90] Nehmzow, U. and Smithers, T. (1990). *Mapbuilding using Self-Organising Networks in Really Useful Robots*. In *International Conference on Simulation of Adaptive Behaviours, Paris*, 509–520.
- [Pav90] Pavlidis, T. (1990). *Algorithmen zur Grafik und Bildverarbeitung*. Heise Verlag.
- [Pom93] Pomerleau, D. (1993). *Neural Network Perception for Mobile Robot Guidance*. Kluwer Academic Publishers.
- [RBK98] Rowley, H. A., Baluja, S., and Kanade, T. (1998). *Neural Network-Based Face Detection*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38.
- [RKE98] Rigoll, G., Kosmala, A., and Eickeler, S. (1998). *High Performance Real-Time Gesture Recognition Using Hidden Markov Models*. In *Gesture and Sign-Language in Human-Computer Interaction*, 69–80.
- [Ros94] Rosenfeld, A. (1994). *Multiresolution Image Processing and Analysis*, vol. 12 of *Springer Series in Information Science*. Springer-Verlag.
- [RP98] Roy, D. and Pentland, A. (1998). *Multimodal Adaptive Interfaces*. In *AAAI Spring Symposium on Intelligent Environments*.
- [RPT00] Roy, N., Pineau, J., and Thrun, S. (2000). *Spoken Dialog Management for Robots*. In *AAAI 2000*.

- [RR98] Rae, R. and Ritter, H. (1998). *3D Real-time Tracking of Points of Interest based on Zero-disparity Filtering*. In *Dynamische Perzeption*, vol. 8 of *Proceedings in Artificial Intelligence*, 105–111. Infix Verlag.
- [RS98] Rosales, R. and Sclaroff, S. (1998). *Improved Tracking of Multiple Humans with Trajectory Prediction and Occlusion Modeling*. In *IEEE Conference on Computer Vision and Pattern Recognition, Workshop on the Interpretation of Visual Motion*.
- [SB98] Steinhage, A. and Bergener, T. (1998). *Dynamical Systems for the Behavioral Organization of an Anthropomorphic Mobile Robot*. In *From Animals to Animats 5: Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior (SAB 98)*. M.I.T. Press.
- [SB01] Schulz, D. and Burgard, W. (2001). *Probabilistic state estimation of dynamic objects with a moving mobile robot*. *Robotics and Autonomous Systems*, 34:107–115.
- [SBC00] Schulz, D., Burgard, W., and Cremers, A.B. (2000). *State Estimation Techniques for 3D Visualizations of Web-based Tele-operated Mobile Robots*. *Künstliche Intelligenz*, 4:16–22.
- [SBFC01] Schulz, D., Burgard, W., Fox, D., and Cremers, A.B. (2001). *Tracking Multiple Moving Targets with a Mobile Robot using Particle Filters and Statistical Data Association*. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.
- [SCW⁺99] Srinivasan, M.V., Chahl, J.S., Weber, K., Venkatesh, S., Nagle, M.G., and Zhang, S.W. (1999). *Robot navigation inspired by principles of insect vision*. *Robotics and Autonomous Systems*, 26:203–216.
- [SD91] Shufeldt, H.H. and Dunlap, G.D. (1991). *Piloting and Dead Reckoning*. Naval Institute Press, Annapolis, MD.
- [SDG⁺00] Stephan, V., Debes, K., Gross, H.-M., Wintrich, F., and Wintrich, H. (2000). *A Reinforcement Learning based Neural Multi-Agent-System for Control of a Combustion Process*. In *IEEE-INNS-ENNS Internat. Joint Conference on Neural Networks (IJCNN 2000)*, vol. VI, 217–222. IEEE Computer Society Press.
- [SEN98] Steffens, J., Elagin, E., and Neven, H. (1998). *PersonSpotter - Fast and Robust System for Human Detection, Tracking and Recognition*. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG'98)*, 516–521. IEEE Computer Society.
- [SHF99] Schmidt, G., Hanebeck, U., and Freyberger, F. (eds.) (1999). *Autonome Mobile Systeme 1999*. Informatik aktuell. Springer Verlag.
- [SM99] Schaeffer, C. and May, T. (1999). *Care-O-bot: A System for Assisting Elderly or Disabled Persons in Home Environments*. In *Proceedings of AAATE-99, Düsseldorf*.

- [SP95] Starner, T. and Pentland, A. (1995). *Real-time american sign language recognition from video using hidden markov models*. In *IEEE International Symposium on Computer Vision*. IEEE Computer Society.
- [SPG01] Schauer, C., Paschke, P., and Gross, H.-M. (2001). *Model and Application of a Binaural 360 Degree Sound Localization System*. In *IEEE-INNS Intern. Joint Conference on Neural Networks (IJCNN 2001)*.
- [Sri92] Srinivasan, M.V. (1992). *How bees exploit optical flow: Behavioral experiments and neural models*. *Philosophic Transactions of the Royal Society of London B*, 337:253–259.
- [SS98] Schraft, R.D. and Schmierer, G. (1998). *Serviceroboter - Produkte, Szenarien, Visionen*. Springer-Verlag.
- [SSG98] Stephan, V., Seiler, T., and Gross, H.-M. (1998). *Vorhersage von optischen Flußfeldern zur antizipativen Navigation*. In *Proceedings Workshop "Dynamische Perception"*, Bielefeld, Proceedings in Artificial Intelligence, 151–158. infix-Verlag.
- [SSM98] Schraft, R.D., Schaeffer, C., and May, T. (1998). *The Concept of a System for Assisting Elderly or Disabled Persons in Home Environments*. In *Proceedings of the 24th IEEE IECON, Aachen*, vol. 4.
- [Ste97] Steinhage, A. (1997). *Dynamical Systems for the Generation of Navigation Behavior*. Ph.D. thesis, Ruhr-Universität Bochum.
- [SZPG00] Schauer, C., Zahn, T., Paschke, P., and Gross, H.-M. (2000). *Binaural sound localization in an Artificial Neural Network*. In *Proceedings IEEE-ICASSP'2000*, vol. II, 865–868. IEEE Press.
- [TBB⁺98] Thrun, S., Bücken, A., Burgard, W., Fox, D., Fröhlinghaus, T., Henning, D., Hofmann, T., Krell, M., and Schmidt, T. (1998). *Map Learning and High-Speed Navigation in RHINO*. In David Kortenkamp, R. Peter Bonasso, and Robin Murphy (eds.), *AI-based Mobile Robots: Case Studies of Successful Robot Systems*, 1–25. AAAI Press / The MIT Press.
- [TBB⁺99a] Thrun, S., Bennewitz, M., Burgard, W., Cremers, A.B., Dellaert, F., Fox, D., Haehnel, D., Rosenberg, C., Roy, N., Schulte, J., and Schulz, D. (1999). *MI-NERVA: A tour-guide robot that learns*. In *German Conference on Artificial Intelligence (KI-99)*. Springer.
- [TBB⁺99b] Thrun, S., Bennewitz, M., Burgard, W., Cremers, A.B., Dellaert, F., Fox, D., Haehnel, D., Rosenberg, C., Roy, N., Schulte, J., and Schulz, D. (1999). *MI-NERVA: A second generation mobile tour-guide robot*. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

- [TBB⁺00] Thrun, S., Beetz, M., Bennewitz, M., Burgard, W., Cremers, A.B., Dallaert, F., Fox, D., Hähnel, D., Rosenberg, C., Roy, N., Schulte, J., and Schulz, D. (2000). *Probabilistic algorithms and the interactive museum tour-guide robot minerva*. *International Journal of Robotics Research*, 19(11):972–999.
- [TBF98] Thrun, S., Burgard, W., and Fox, D. (1998). *A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots*. *Machine Learning and Autonomous Robots (joint issue)*, 31(5):1–25.
- [TBF00] Thrun, S., Burgard, W., and Fox, D. (2000). *A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping*. In *Proc. of the IEEE International Conference on Robotics & Automation*.
- [TBF01] Thrun, S., Burgard, W., and Fox, D. (2001). *Probabilistic algorithms for mobile robot mapping*. Tutorial IJCAI-2001.
- [TDA98] Terrillon, J.-C., David, M., and Akamatsu, S. (1998). *Automatic Detection of Human Faces in Natural Scene Images by Use of a Skin Color Model and of Invariant Moments*. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG'98)*, 112–117. IEEE Computer Society.
- [Thr93] Thrun, S. (1993). *Exploration and Model Building in Mobile Robot Domains*. In *Proceedings of the IEEE International Conference on Neural Networks*, 175–180.
- [Thr99] Thrun, S. (1999). *Learning Maps for Indoor Mobile Robot Navigation*. *Artificial Intelligence*, 99(1):21–71.
- [Thr00a] Thrun, S. (2000). *Probabilistic algorithms in robotics*. *AI Magazine*, 21(4):93–109.
- [Thr00b] Thrun, S. (2000). *Probabilistic robotics*. Tutorial AAAI-2000.
- [TP91] Turk, M. and Pentland, A. (1991). *Face recognition using eigenfaces*. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*.
- [Tri99] Triesch, J. (1999). *Vision-based Robotic Gesture Recognition*. Ph.D. thesis, Ruhr-Universität Bochum.
- [VDI00] VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik (2000). *Robotik 2000 : Leistungsstand - Anwendungen - Visionen - Trends*. VDI-Berichte 1552. VDI-Verlag.
- [VF92] Viennet, E. and Fogelman Soulie, F. (1992). *Scene Segmentation using Multiresolution Analysis and MLP*. In *ICANN '92*, 1599–1602.
- [WADP97] Wren, C., Azarbayejani, A. and Darrell, T., and Pentland, A. (1997). *Pfinder: Real-Time Tracking of the Human Body*. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):780–785. M.I.T. Media Lab Techreport TR 353.

- [Web98] Weber, S. (1998). *Monocular Visual Robot Navigation in Indoor Environments*. Tech. rep., TU Ilmenau, Schriftenreihe des Fachgebiets Neuroinformatik. ISSN 0945-7518.
- [WF98] Wachsmuth, I. and Fröhlich, M. (eds.) (1998). *Gesture and Sign Language in Human-Computer Interaction*. Lecture Notes in Artificial Intelligence. Springer.
- [WFJv00] Weber, J., Franken, L., Jörg, K.-W., and von Puttkammer, E. (2000). *Motion Estimation and Segmentation Using a Recurrent Mixture of Experts Architecture*. In *16. Fachgespräch Autonome Mobile Systeme (AMS 2000)*, 111–120. Springer.
- [Wre00] Wren, C. (2000). *Understanding Expressive Action*. Ph.D. thesis, Massachusetts Institute of Technology.
- [YC97] Yow, K.C. and Cipolla, R. (1997). *Feature-based Human Face Detection*. *Image and Vision Computing*, 15:713–735.
- [YLW97] Yang, J., Lu, W., and Waibel, A. (1997). *Skin-Color Modeling and Adaptation*. Tech. rep., Carnegie Mellon University. CMU-CS-97-146.
- [ZDH⁺01] Zobel, M., Denzler, J., Heigl, B., Nöth, E., Paulus, D., Schmidt, J., and Stemmer, G. (2001). *Demonstration von Bildverarbeitung und Sprachverstehen in der Dienstleistungsrobotik*. In *17. Fachgespräch Autonome Mobile Systeme (AMS'01)*, 141–147.
- [Zim91] Zimmermann, H.-J. (1991). *Fuzzy Set Theory – and its Applications*. Kluwer Academic Publishers.