

## 0 Robotik

Kapitelherausgeber: Alois C. Knoll

Autoren: W. Burgard, T. Christaller, A. Knoll

### 0.1 Einführung

Mit dem Begriff des Roboters<sup>1</sup> verbinden sich die unterschiedlichsten Assoziationen: Von Menschmaschinen (Androiden), gegenüber deren Intelligenz die menschliche nur wie ein Durchgangsstadium der Evolution wirkt, über Industrie- und Serviceroboter, die seit den sechziger Jahren in wachsender Zahl die Fabrikhallen erobern, bis hin zu abstrakten Programmen, die mehr oder weniger automatisch das World-Wide-Web nach interessanten Inhalten absuchen. Ihnen allen ist gemein, daß sie versprechen, dem Menschen Teile seiner Arbeit abzunehmen – bislang hauptsächlich körperliche oder gefährliche Tätigkeiten, in Zukunft vermehrt auch solche, die kognitive Fähigkeiten voraussetzen.

#### 0.1.1 Historischer Abriss

Die Robotik gründet sich auf zwei *enabling technologies*: Telemanipulatoren und die Technik der numerischen Steuerung von Werkzeugmaschinen (NC: Numerical Control bzw. CNC: Computer NC). Telemanipulatoren sind von Menschen ferngesteuerte Maschinen, die gewöhnlich aus einem Arm mit einem Greifer bestehen. Ihre Bewegungen folgen direkt den Bewegungen, die der Bediener an einer an seinem Arm und seiner Hand befestigten Steuereinheit vormacht. Die ersten dieser Maschinen wurden Anfang der vierziger Jahre zur Handhabung radioaktiven Materials eingesetzt. Die Technik der numerischen Steuerung erlaubt es, die Achsen einer Werkzeugmaschine präzise in bezug auf ein Koordinatensystem zu bewegen, wobei die anzufahrenden Punkte als Folge von Koordinaten ursprünglich auf einem Lochstreifen kodiert wurden. Diese Technik wurde zuerst 1952 durch eine Arbeitsgruppe am MIT eingeführt, deren weitere Arbeiten zur ersten Programmiersprache für die Programmierung von Werkzeugmaschinen führten (APT: Automatic Programmed Tools). Die Kombination beider Techniken führte zum ersten *programmierbaren Manipulator*, einem Arm, dessen Achsbewegungen numerisch gesteuert werden. Der erste nach diesem Prinzip arbeitende Industrieroboter wurde 1961 installiert, ab Mitte der sechziger Jahre wurden diese Maschinen dann kommerziell vertrieben (*Unimate* der Firma Unimation). Heute gibt es Industrieroboter in den unterschiedlichsten Ausführungen (Achszahl, Größe und Form des Arbeitsraums, Tragkraft, Geschwindigkeit, Genauigkeit) für die unterschiedlichsten Aufgaben (Schweißen, Tragen, Palettieren, Säubern, Verpacken, Montieren, . . . ) in einer Vielzahl von Anwendungsbereichen (Industrie, Logistik, Medizin, Raumfahrt, . . . ). Weltweit werden jährlich etwa 80.000 bis 100.000 dieser Roboter verkauft, der Gesamtbestand beträgt gegenwärtig knapp eine Million [NoR97]. Der Hauptabnehmer ist – je nach Land – die Automobilindustrie mit ihren einfachen Tätigkeiten; es ist jedoch damit zu rechnen, daß ihr Anteil sinken wird, weil sich mit der langsamen Einführung von Sensoren und kognitiven Fähigkeiten die Anwendungsmöglichkeiten und Absatzmärkte deutlich erweitern werden.

Die Entwicklung der *mobilen Roboter* gründet einerseits in dem Wunsch, Transportvorgänge in Produktionsprozessen zu automatisieren, zum anderen in der Absicht, Fahrzeuge im freien Gelände autonom fahren zu lassen. Ersteres führte Ende der siebziger Jahre zur Entwicklung fahrerloser Transportsysteme (FTS), die in Fabrikhallen verschieden große Objekte zu den einzelnen Bearbeitungsstationen bewegen können. Sie orientieren sich mit Hilfe bestimmter ausgezeichneter Punkte im Raum (die z.B. von Laser-Scannern detektiert werden) oder entlang eines in den Boden eingelassenen Metallbandes. Interesse an der autonomen Fahrt im freien Gelände hat vor allem das Militär, in gewissem Umfang (zur Unterstützung des Fahrers in kritischen Situationen) auch die Autoindustrie. Entsprechende Forschungsaufträge wurden in den USA seit Anfang der achtziger Jahre vergeben, in Deutschland startete das zivile *Eureka*-Projekt *Prometheus* im Jahre 1986. In jüngerer Zeit sind noch

---

<sup>1</sup>Er geht auf den tschechischen Schriftsteller Karel Capek zurück, der ihn für sein Bühnenstück „R.U.R. Rossum's universal robots“ (erschienen 1921) vom tschechischen Wort für Zwangsarbeit ableitete.

weitere Formen der Fortbewegung hinzugekommen: von ein- und mehrbeinigen „Hüpfmaschinen“ Anfang der achtziger Jahre bis hin zu den „insektoiden“ Mehrbeinern, die nach biologischem Vorbild modelliert wurden. Praktische Anwendungen finden sich hier möglicherweise in der Fortbewegung im unwegsamen Gelände oder in Kanalröhren; erwähnt seien in diesem Zusammenhang auch die (halb-)autonomen Meeresroboter. Seit wenigen Jahren sind radgetriebene mobile Roboter für die Bewegung in Gebäuden kommerziell erhältlich, neben der Verwendung für Forschungszwecke dienen sie zunehmend als Plattform für Servicedienste (Holen und Bringen), etwa in Krankenhäusern; daneben finden sich eine ganze Reihe weiterer Service-Anwendungen, die gerade das Experimentalstadium verlassen. Bei den humanoiden Robotern schließlich begann die Entwicklung

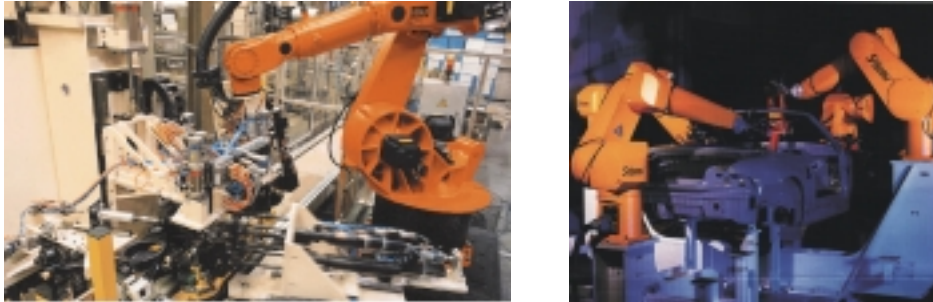


Abbildung 1: Beispiele für den Einsatz von Industrierobotern in der Automobilfertigung. Links: Ein Roboter der Firma Kuka trägt eine Spezialvorrichtung zur Achsmontage. Rechts: Zwei Roboter der Firma Stäubli-Unimation vermessen eine Rohkarosserie.

in Japan mit dem ersten primitiven „Torso“ (Wabot-I) im Jahre 1975 über den mit Händen und Füßen Orgel spielenden Wabot-II (1985) bis zu dem aktuellen Modell Wabot-III, das bereits über einfache kognitive Fähigkeiten verfügt. Gleiches gilt für den Humanoiden „Cog“ des MIT-AI-Lab, der seit 1994 entwickelt wird. Einer breiteren Öffentlichkeit bekannt wurde der Roboter der Firma Honda, der 1999 vorgestellt wurde. Er wird von einem menschlichen Bediener gesteuert, kann aber autonom laufen und Treppen steigen. Die Konstruktion solcher Systeme mit dem Ziel der vollständigen Autonomie wird in weiten Kreisen nach wie vor als „Science Fiction“ betrachtet; dennoch ist es nicht schwierig vorauszusehen, daß komplette Humanoide oder Teilsysteme in nicht allzu ferner Zukunft als *personal robots* Eingang in unser Alltagsleben finden werden – zuerst nur für spezielle Aufgabennischen, dann als immer universeller instruier- und verwendbare „Maschinensklaven“.

### 0.1.2 Einordnung

Die Konstruktion von Robotern als *autonome intelligente Systeme* bedingt ein enges interdisziplinäres Zusammenwirken der Gebiete mechanisches und elektrisches Design der aktorischen Komponenten mit den sensorisch-kognitiv orientierten Bereichen: Signalverarbeitung, Mustererkennung, Wissensrepräsentation und Inferenz, Bilderkennung und -deutung, Planung, Navigation, Lernen, Dialogführung Mensch-Maschine, Architekturplan- und verhaltensbasierter sowie hybrider Systeme, Kooperation und Kommunikation in verteilten (Multi-Agenten-)Systemen. Mit anderen Worten: *Robotik ist die umfassende Disziplin, die wesentliche Teilbereiche der Gebiete KI, Kognitionswissenschaften und Aktor-Design zusammenführt und in einem technischen System operationalisiert, welches komplexe Handlungen in unterschiedlichen Umwelten autonom ausführen kann, um damit zielgerichtet Manipulationsaufgaben zu erledigen.*

Unterteilt nach einzelnen Typen von Robotern ergeben sich aus Sicht der künstlichen Intelligenz insbesondere die folgenden Problembereiche:

- Bei den *stationären Robotern* (Beispiele zeigt Abbildung 1) geht es neben der Planung kollisionsfreier Bewegungen des Arms und des Greifers um die (modellbasierte) Erkennung und Verfolgung von Objekten sowie die exakte Bestimmung ihrer Lage (und der Griffpunkte) für den Greifvorgang. Dazu werden

im allgemeinen optische Sensoren verwendet; zur Erhöhung von Genauigkeit, Robustheit und zur Vergrößerung des Sichtwinkels können auch mehrere solcher Sensoren zum Einsatz kommen. In diesem Fall stellt sich zusätzlich die Aufgabe der (dynamischen) Platzierung dieser (bewegten oder beweglichen) Sensoren und der Sensordatenfusion. Von besonderem Interesse werden zukünftig zwei Felder sein: kognitiv adäquate Steuerung (bzw. Programmierung) durch den Menschen samt Lernvermögen sowie die Kooperation einer ggf. sehr großen Anzahl direkt kooperierender Roboter („Roboterfarm“) mit einem hohen Maß an Adaptationsvermögen, d.h. der möglichst automatischen Anpassung an unterschiedliche Handhabungsobjekte und der autonomen Behandlung von Fehlersituationen. Da solche Systeme in der Produktion eingesetzt werden sollen, ergibt sich neben der bereits erwähnten Robustheit insbesondere die Forderung nach Echtzeitfähigkeit des gesamten Systems, was eine entsprechende Architektur mit einer geeigneten Abstimmung zwischen reaktiven und planbasierten Komponenten erfordert.

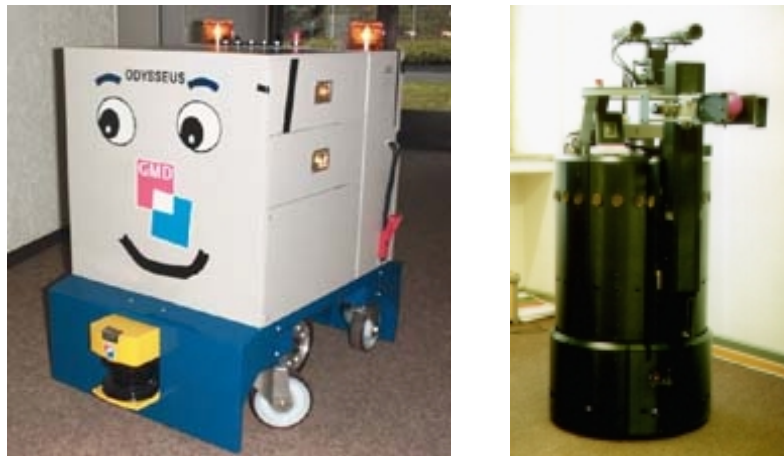


Abbildung 2: Beispiele für mobile Roboter. Links: Ein fahrerloses Transportsystem der Firma Daum im Einsatz bei der GMD. Rechts: Der Roboter „Rhino“ der Universität Bonn, eine mobile Plattform mit Ultraschallsensoren und zwei Kameras zur Navigation in Büroumgebungen.

- Bei den *mobilen Robotern* (Abbildung 2) wird weiterhin die Hauptaufgabe in der Navigation bestehen, die in unterschiedlich stark strukturierten und dynamischen, ggf. auch unbekanntem Umgebungen zuverlässig arbeitet. Hier werden ebenso wie im stationären Fall unterschiedliche Sensoren eingesetzt, die häufig nach verschiedenen Wirkungsprinzipien arbeiten (Kameras, Ultraschall-Echolote, Laser-Scanner, Radwinkelgeber). Das erfordert eine Methodik zur Sensordatenfusion, die diese unterschiedliche Domänen verbindet. Da die Umweltsituation sich hier besonders schnell ändern kann (ein Mensch tritt vor den fahrenden Roboter), ist es eine besondere Herausforderung, Architekturen für eine zeitnahe Sensor-Aktor-Kopplung zu implementieren, die noch dazu mit der begrenzten Leistung des mitgeführten Rechners auskommen müssen. Daneben wird aber auch hier die Kooperation/Kommunikation mehrerer solcher Roboter zur Erreichung eines gemeinsamen Ziels wichtig werden<sup>2</sup>, sowie natürlich die direkte Interaktion mit dem Menschen zum Zwecke der Aufgabenerläuterung. Von der Klasse der mobilen Roboter gibt es einen fließenden Übergang zu den *Servicerobotern*, die in ihrer großen Mehrzahl nicht ortsfest sein werden (eine Ausnahme sind z.B. Robotersysteme, die Automobile betanken), deshalb geht es hier um eine gute Abstimmung zwischen dem die Mobilität erzeugenden „Lokomotionsteil“ und dem Manipulationsteil, der die eigentliche Handhabungsaufgabe wahrnimmt (Abbildung 3).
- Die „Krone der Robotik“ stellen zweifellos die *humanoiden Roboter* (Abbildung 4) dar, bei denen neben die Aspekte der Fortbewegung und Manipulation noch die umfassende maschinelle Implementierung

<sup>2</sup>Zu einiger Bekanntheit ist der Wettbewerb „RoboCup“ gelangt, bei dem gerade diese Kooperationsfähigkeit eine wesentliche Voraussetzung zum Spielerfolg ist.



Abbildung 3: Beispiele für Serviceroboter. Links: Universalroboter der Firma Nomad Technologies mit multi-sensorgestützter fahrbarer Plattform und aufgesetztem Puma-Arm der Firma Unimation. Mitte: Prototyp einer hochbeweglichen „Schlange“ der GMD mit der Fähigkeit, in Kanälröhren zu navigieren. Rechts: Beispiel für einen „insektoiden Roboter“ zur Navigation in unwegsamem Gelände (z.B. zur Minensuche).

kognitiver Fähigkeiten und ihre Integration in den „Körper“ tritt. Die Fähigkeiten dieser Roboter sollten weitgehend dem Menschen nachgebildet werden (bzw. dem mit dem Roboter kooperierenden Menschen als solche erscheinen), um eine bruchlose Instruktion zu ermöglichen und – vielleicht – dem Roboter über seine menschenähnliche *Verkörperung* einen Zugang zu menschlichen Erfahrungswelten zu schaffen. Was die Problematik der Nachbildung betrifft, befinden wir uns ganz sicher erst am Anfang der Entwicklung: weder kann von einer adäquaten Emulation der menschlichen sensorisch-kognitiven Fähigkeiten die Rede sein, noch gibt es bislang wirklich überzeugende und praktisch einsetzbare Lösungen für die Nachbildung von Händen, Armen, Beinen – geschweige denn eine ausdauernde Energiequelle, die ganz offensichtlich Voraussetzung für wahre Autonomie ist. Es ist jedoch durchaus klar, wie ein Forschungsprogramm aussehen sollte, welches diese Ziele erreichbar scheinen läßt (siehe dazu etwa [BS94]).

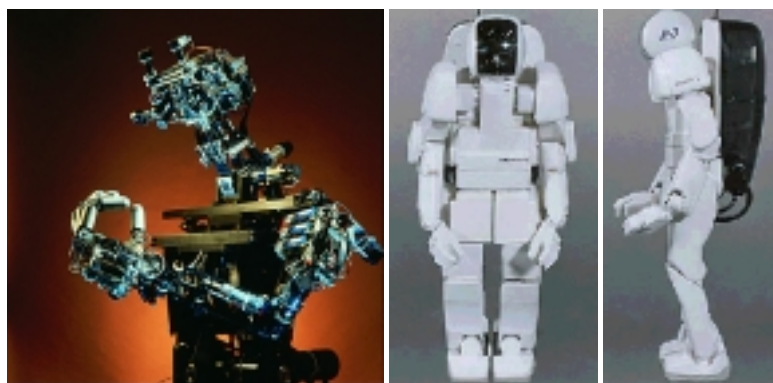


Abbildung 4: Plattformen für humanoide Roboter. Links: „COG“, ein ab der Hüfte aufwärts im AI-Lab des MIT installierter Torso, der etwa den Abmessungen des menschlichen Körpers entspricht. Er verfügt über Arme, ein aktives Stereokamerasystem, Mikrofone und Lautsprecher. Mitte und rechts: Prototyp der Firma Honda. Er ist nicht autonom, besitzt aber die Fähigkeit zum Laufen und Treppensteigen; nur eingeschränkte Sensorik zur Kommunikation und Umweltwahrnehmung.

### 0.1.3 KI und Robotik

Für lange Zeit waren die Arbeiten am Stanford Research Institute (SRI) die sichtbarsten, in denen (mobile) Roboter entwickelt wurden, um ein künstlich intelligentes sensomotorisches Gesamtsystem zu entwickeln. Der erste funktionsfähige Roboter in einer ganzen Serie war Shakey [Rap76]. Hier wurden die ersten Planungsverfahren, z.B. STRIPS und NOAH, Wissensrepräsentationen und Architekturen aus der KI-Perspektive für Roboter entwickelt. Diese Arbeiten haben vielfältigen Einfluss in der KI gehabt. Die Nachfolgesysteme, Flakey I und II dienten dann auch als Vorlage für die sogenannte Saphira-Architektur [KM] und für professionell gefertigte mobile Roboter.

Eine andere frühe Arbeit war das System SHRDLU von Terry Winograd [Win72]. Dies gilt zwar in erster Linie als eines der ersten leistungsfähigen Sprachverarbeitungssysteme doch das Szenario von SHRDLU ist ein Roboterszenario: Ein Greifer und eine Deckenkamera beobachten eine Tischoberfläche, auf der verschieden farbige und geformte Bauklötze (Würfel, Quader, Pyramiden und eine Kugel) liegen. Mithilfe natürlichsprachlicher Anweisungen bewegte der Greifer einzelne Bauklötze, baute Türme auf und ab. Der entscheidende Unterschied zwischen Shakey und SHRDLU ist, daß letzteres den sensomotorischen und physikalischen Teil simuliert.

Erst ab 1985 gab es aus vielerlei Gründen ein wachsendes Interesse an Robotern in der KI. Am besten trifft diesen Zeitgeist der Titel eines programmatischen Artikels von Rodney Brooks „Achieving Artificial Intelligence through building robots“ [Bro86a]. Im Kern begann dies mit einer Kritik an einem ausschliesslich auf Logik basierten Verständnis des Phänomens der Intelligenz und führte zu einem dann neuen Teilgebiet der KI der sogenannten verhaltensbasierten KI. Eine ganz ähnliche Form der Kritik hatte schon vorher zu dem Gebiet des Konnektionismus geführt und etwa zur selben Zeit wie die verhaltensbasierte KI entstand auch das Gebiet *Artificial Life*. Zwischen diesen gibt es einige Überschneidungen aber lediglich bei der verhaltensbasierten KI spielt die Konstruktion physikalischer Agenten, d.h. Roboter eine so überwältigende und methodologisch herausgehobene Rolle.

### 0.1.4 Allgemeine weiterführende Literatur und Konferenzen

Es gibt eine große Zahl von Lehrbüchern über das Gebiet, stellvertretend eine kleine Auswahl für die verschiedenen Aspekte: Allgemeine Einführungen in unterschiedliche Aspekte [Reh99, Ful98, Cra89, Sel96, Wis99, Pan91, SB96, Lee90, Ang97, AB98, Neh99, PS99, Cla97, MB93, CSS94], Industrieroboter [Reh99, Nof99, Mor94, Wis99, Spi90, MTF96, Vol92, LR95], Mobile Roboter [Kni91, Kor98, CW90, CW93, Bor96, Ada98, KBM98], Serviceroboter [Eng89, SS98, vR97, Zel98], Sensordatenfusion [WL90, Hal92, Sad96, BI97, CGS97, GMN97], Planung [Can88, Lat91a, Fuj91, TPJ92]. Eine Übersicht über die Aktivitäten in Deutschland gibt [PDK98].

Die wesentlichen wissenschaftlichen Zeitschriften sind: *Robotica* (Int. Fed. of Robotics, Cambridge Univ. Press), *IEEE Transactions on Robotics and Automation* (IEEE Press), *Robotics and Autonomous Systems* (Elsevier), *The International Journal on Robotics Research* (Sage Press), *Autonomous Robots* (Kluwer), *Adaptive Behaviour* (MIT Press), *Artificial Life*, *Behavioral and Brain Sciences*.

Schließlich seien einige dem Thema gewidmete Konferenzreihen genannt: IEEE International Conference on Robotics and Automation, Intelligent Robot Systems (Robotics Society of Japan und weitere), DARS: Distributed Autonomous Robot Systems (IEEE und weitere), Humanoids (IEEE).

Im folgenden stellen wir drei Teilbereiche der Robotik in ihrem gegenwärtigen Stand dar, welche besonders engen Bezug zur künstlichen Intelligenz in ihrem klassischen Verständnis haben. In Unterkapitel 0.2 reißen wir wesentliche Fragen an, die im Zusammenhang mit der Steuerung kooperierender stationärer Roboter auftreten: Planung, Sensordatenfusion, Lernen von Montagesequenzen, Instruktion im multimodalen Dialog. Unterkapitel 0.3 ist der Frage der Steuerung und des Architekturdesigns verhaltensbasierter Roboter gewidmet. Das dritte Unterkapitel 0.4 befaßt sich mit der Steuerung und der Sensorik von mobilen Robotern, speziell der Navigation.

## 0.2 Künstliche Intelligenz und Stationäre Roboter

Die abwartende Haltung der Industrie gegenüber dem Einsatz adaptiver Systeme mit komplexer Sensorik hat zwar dazu geführt, daß Roboter im industriellen Umfeld im Regelfall noch nach den gleichen Prinzipien programmiert werden wie die ersten Roboter in den sechziger Jahren, d.h. über sehr einfache imperative Sprachen. Dennoch ist absehbar, daß die deutliche Verbesserung der Sensoren bei gleichzeitigem extremem Preisverfall in den nächsten Jahren dazu führen wird, daß der mit optischen Sensoren ausgerüstete Roboter die Regel ist. Gleichzeitig erhalten durch die extrem gestiegene Rechenleistung die Robotersteuerungen eine *Planungsmächtigkeit*, die bislang nur *off-line* vorgeschaltete Simulationssysteme erbringen konnten (und die damit nur auf einen statischen, im voraus bekannten Umweltzustand anwendbar waren).

Die vor Ort zur Verfügung stehende Rechenleistung ermöglicht noch einen weiteren qualitativen Sprung in der Programmierbarkeit: die Interpretation von natürlichsprachlichen Anweisungen (möglicherweise auch Gesten, Mimik) eines Instrukteurs im Kontext der vorliegenden Umweltsituation sowie die ebenso natürlichsprachlich geäußerte Rückmeldung des Robotersystems über erkannte Problemlagen. Mit anderen Worten: die Abwicklung eines *Handlungsdialogs* – über einer zunächst eingeschränkten Welt – rückt in den Bereich des möglichen; die Roboterinstruktion kann ganz ohne klassische Programmierarbeit vorgenommen werden. Ein reibungsloses und intuitiv vom menschlichen Instrukteur gesteuertes Ineinandergreifen von Anweisungsfolgen, Montagesequenzen und Umweltwahrnehmung erfordert dabei auf der Seite des Robotersystems ein ausgesprochen komplexes Zusammenspiel sensorischer, kognitiver und manipulativer Fähigkeiten – mithin sind also auch hier Kernfelder der Künstlichen Intelligenz betroffen.

Wir beschreiben in 0.2.1 zunächst den Stand der einzelnen Felder, deren Kombination die Realisierung eines solchen Systems zuläßt und stellen dann in 0.2.2 ein funktionsfähiges System vor, das prinzipiell demonstriert, daß der Ansatz eines rein über natürliche Anweisungen gesteuerten Systems tragfähig ist. In einem Ausblick (Abschnitt 0.2.3) zeigen wir Entwicklungslinien auf, die unserer Ansicht nach verfolgt werden sollten, um derartige sensorbasierte adaptive System praktisch einsatzfähig zu machen.

### 0.2.1 Aufgabenstellungen, Forschungsfragen, Stand

#### 0.2.1.1 Planung für die Aktorik

Die Planungsschritte, die dem Zusammenbau<sup>3</sup> eines Aggregats aus elementaren Objekten vorausgehen (oder sie begleiten), sind verschiedenen Abstraktionsebenen zuzuordnen. Wir gehen kurz auf den Stand der Forschung auf diesen Ebenen ein, wobei wir mit der abstraktesten „roboterfernsten“ Ebene, der Montagesequenzplanung, beginnen<sup>4</sup>:

**Montagesequenzplanung.** Aufgabe der Montagesequenzplanung ist es, festzulegen, wann welche Operation mit welchem Objekt durchgeführt wird. Dazu werden Unterziele festgelegt und zur Erfüllung jedes Unterziels bestimmt, ob seine Vorbedingungen erfüllt sind. Solche Vorbedingungen sind beispielsweise die Erreichbarkeit eines zu montierenden Objekts durch den Manipulator, die korrekte Lage des Teilaggregats, an die ein Objekt anmontiert werden soll, die Verfügbarkeit eines geeigneten Manipulators, usw. Die ersten Arbeiten in diesem Bereich stammen aus den frühen siebziger Jahren und hatten typischerweise Aufgaben aus der *blocks-world* zum Gegenstand; STRIPS [FN71] ist sicher der bekannteste dieser Planungsformalisten. Innerhalb dieser Domäne waren Aufgaben zu lösen, die durch eine eindeutig vorgegebene Ausgangssituation und ein ebenso eindeutig spezifiziertes Ziel gekennzeichnet sind (Erzeugung eines Plans zum Aufeinanderstapeln von unterscheidbaren Würfeln in eine gewünschte Konfiguration). Obwohl zunächst von geringer praktischer Relevanz, waren diese Ansätze die Grundlage für verschiedene plangenerierende Systeme für den Fertigungsbereich.<sup>5</sup>

<sup>3</sup>Wir betrachten hier immer den Fall der Montage von komplexen Aggregaten aus einfachen Elementarobjekten. Die anderen Tätigkeiten für Industrieroboter, wie Schweißen, Schleifen, Lackieren sind als Programmier- und Planungsaufgabe und damit aus Sicht der Anwendbarkeit von KI-Techniken weniger interessant.

<sup>4</sup>In der industriellen Anwendung liegt über der Montagesequenzplanung noch die Planung für die Materialverteilung und Einsatzplanung für alle zusammenarbeitenden Aktoren einer größeren Produktionseinheit (also z.B. einer Fabrikhalle), siehe etwa [LB91, Rev99].

<sup>5</sup>Wobei am Rande bemerkt sei, daß hier auch die schon früh entwickelten Techniken zur Planung in hierarchisch untergliederten Suchräumen (beginnend mit ABSTRIPS [Sac74]) zum Planen mit Einschränkungen und *least commitments* ([Wei94], implementiert



In heutigen CAD-Systemen wäre es im Prinzip möglich, die Montagesequenzplanung<sup>6</sup> trotz ihrer hohen algorithmischen Komplexität automatisch vorzunehmen, weil die Geometrie aller Teile samt ihrer Materialbeschaffenheit vollständig modelliert. Mit diesem Objektwissen wäre zudem eine Planung von Annäherungsbewegungen beim Aggregatbau, die Simulation dynamischer Kräfte bei der Ausführung von Bewegungen, Einbezug von Toleranzen und sogar die Erzeugung von Programmen zur Robotersteuerung möglich. Dennoch ist festzuhalten, daß dies alles nicht oder nur in geringem Maße geschieht: auch in Industriebetrieben, die die Konstruktion vollständig über CAD-Systeme abwickeln, wird die Erzeugung von Montagesequenzen, ja sogar die Codierung der entsprechenden Roboterprogramme bis heute vom Menschen vorgenommen.<sup>7</sup> Dies hat im wesentlichen drei Gründe: Erstens bestehen die Steuerungsprogramme für die Roboter nur zum geringeren Teil aus den eigentlichen Bewegungskommandos; den größeren Teil nimmt die Generierung von Prozeß- und Überwachungsdaten ein, und hierfür fehlen den CAD-Systemen die nötigen Erzeugungs- bzw. Integrationskomponenten. Wesentlich wichtiger ist jedoch der zweite Grund, der darin besteht, daß im CAD-System keine *benannte* Zielspezifikation für das endgültige Aggregat möglich ist, d.h. der gewünschte Endzustand kann nicht anders spezifiziert werden, als durch das „manuelle“ Fügen verschiedener Teile zum gewünschten Aggregat<sup>8</sup>. Wenn dieses Zusammenfügen aber die unabdingbare Voraussetzung zur Beschreibung des Aggregats ist, dann entfällt natürlich die Notwendigkeit und die Möglichkeit, die Montagesequenz automatisch zu finden. Drittens gibt es keine Möglichkeit, Konstruktionswissen<sup>9</sup> auf neue Konstellationen zu übertragen und damit in neuen Konstruktionssituationen zumindest Vorschläge für Sequenzen machen zu können.

Bei den reinen *Planungssystemen* sind im Zusammenhang mit der Montageplanung verschiedene Ausprägungen zu unterscheiden:

- Systeme zur Fertigungsplanung für miteinander zu synchronisierende Maschinen auf hoher Abstraktionsebene, also z.B. ganze Fabrikhallen. Zum Einsatz kommt hier die Theorie der *discrete event systems*<sup>10</sup> operationalisiert in Form von Zustandstabellen, endlichen Automaten oder Petri-Netzen samt der dazugehörigen Werkzeuge zur Manipulation (ein typisches Beispiel gibt [Zha99a]). Während diese Ansätze sich hauptsächlich zur deliberativen Planung „im voraus“ eignen, können für dynamische Fälle (sich ändernde Maschinenzustände, Ausfall von Maschinen mit Umleitungsmöglichkeit, Einphasen neuer Bearbeitungszentren im laufenden Betrieb) Multi-Agenten-Systeme mit Kommunikation über das Vertragsnetzprotokoll [SD81, Fox81] vorteilhaft eingesetzt werden [Bak88, Par89, OHB<sup>+</sup>99, BA99]. In beiden Fällen kann aufgrund des hohen Abstraktionsgrads Sensorik nur in der Form des Auftretens eines bestimmten Ereignisses („Sensor hat Teil *X* erkannt“) einbezogen werden.
- Systeme zur Planung von Aktionen einzelner Manipulatoren unter Unsicherheit, d.h. Berücksichtigung des dynamischen Umweltzustandes. Die meisten Planungssysteme gehen davon aus, daß der Ausgangszustand vollständig bekannt ist, daß die umgebende Welt statisch ist und daß nur der Planausführer Zustandsänderungen durchführen kann. Systeme zum Einbezug unsicherer Information, dynamischer Umwelten und nur zeitweise zur Verfügung stehender Betriebsmittel in Planungsprozesse sind erst seit kurzem Gegenstand der Forschung. Grundlagen sind einfache probabilistische Modellierungen von Zuständen

---

etwa in MOLGEN [Ste81]) zum Einsatz kommen.

<sup>6</sup>Dies betrifft mindestens jedoch die Erzeugung sogenannter Montagevorranggraphen, die die Relationen von Objekten repräsentieren und ggf. über unterschiedliche Traversierungen alternative Wege zu einem vollständigen Aggregat aufzeigen. Das bislang leistungsfähigste System, welches auf einfachen Planungstechniken unter Verwendung von Einschränkungen ein Objekt mit 472 Einzelteilen plant, beschreibt [JWC98].

<sup>7</sup>Über erste Ansätze, die Sequenzplanung und die Programmierung zu verknüpfen, wird in [YG99] berichtet; [Zha99b] präsentiert einen Ansatz für ein hybrides Planungs- und Robotersteuerungssystem auf der Basis von Petri-Netzen als *discrete-event-system* zur Spezifikation der Montagesequenz, verknüpft mit einem Steuerungssystem mit kontinuierlichem Zustandsraum.

<sup>8</sup>In gewisser Weise ist das in der Robotik oft umgekehrt: Man weiß, was man montiert haben möchte, aber die Einzelteile sind nur ungenügend beschrieben oder sensorisch erfaßbar. Siehe dazu die programmatischen Ausführungen in [Smi92].

<sup>9</sup>Erfahrungswissen des Menschen oder konzeptuelles/situatives Wissen, welches das System bei vorherigen Konstruktionsvorgängen extrahiert hat.

<sup>10</sup>Einführende Überblicke sind z.B. [RW89, Ho92].

in einer unsicheren Domäne [DHW93] oder aber stochastische Prozesse (partiell beobachtbare Markow-sche Entscheidungsprozesse [Ast65, DL95]). Mit letzteren lassen sich sowohl Unsicherheit bezüglich des Ergebnisses, das eine Handlung hat, als auch die nur unvollständige Beobachtbarkeit des Umweltzustandes modellieren. Das Planungsproblem wird als Kostenfunktion modelliert, wobei jeder Planungsschritt zwar Kosten verursacht, aber gleichzeitig einen Wegfortschritt in Richtung auf einen Zielzustand bringt (eine Übersicht gibt [BDH99]). Hingewiesen sei auf [LL98]; hier wird untersucht, welcher Endzustand bei einer gegebenen Menge minimaler Aktionen, die jeweils ein binäres sensorisches Ergebnis liefern, von einem Startzustand ausgehend erreicht werden kann. Schließlich sind auch für diese Aufgabenstellung Versuche mit dem Vertragsnetzprotokoll gemacht worden [FPWK99].

- Systeme zur Erfüllung von Echtzeitforderungen. Die Einhaltung von Zeitvorgaben kann durch Reduktion des Suchraums über die Abstraktion von Details erfolgen, durch inkrementelle Planung, d.h. Beschränkung auf die Auswahl und direkte Ausführung von Handlung<sup>11</sup> auf Basis einer domänenspezifischen Heuristik [BLG97] oder durch Kopplung von schnellen, reaktiven Planern mit langsameren, deliberativen Subsystemen, wobei das deliberative Subsystem die Parameter des reaktiven im langsamen Zeittakt verändert [GMKB97, BCG<sup>+</sup>99, ZK99].

Planungssysteme, die Pläne für einzelne Manipulatoren unter einem gewissen Maß an Unsicherheit erzeugen können, sind im Prinzip verfügbar. Es bleibt aber festzuhalten, daß bei dem hier interessierenden Thema der Planung unter Unsicherheit und bei beschränkter Zeit für Teams von Agenten (MAP: Multi-Agent-Planning), bei denen ein *Zwang zur Zusammenarbeit* vorliegt, weil sie nur *gemeinsam* Ziele erreichen können, nahezu keine Arbeiten vorliegen<sup>12</sup>. Eine Ausnahme ist [BB97]; hier wird eine STRIPS-artige Repräsentation für nebenläufige, verkettete Aktionen eingeführt und auf ein einfaches Beispiel angewandt. Im übrigen fällt auf, daß die meisten Arbeiten zur dynamischen Planung in der Robotik der Wegplanung für mobile Roboter, speziell unter Vermeidung von Kollisionen, gewidmet sind. Eine Übertragung auf die Problematik der Montage ist nur sehr begrenzt möglich. Schließlich gibt es bislang überhaupt keine implementierten Systeme, die unter den oben genannten Forderungen arbeiten und zusätzlich noch das Eigenverhalten eines Roboters – etwa über ein geometrisches oder gar dynamisches Selbstmodell – mit ins Kalkül ziehen, um Emulation oder Probehandeln wirklichkeitsnah<sup>13</sup> abbilden zu können.

**Bewegungs- und Griffplanung.** Aufgabe der klassischen Bewegungsplanung für stationäre Roboter [FGL87] ist die Erreichung einer hohen *Armgeschwindigkeit* hin auf einen Zielpunkt bei gleichzeitig hoher *Bahntreue* und möglichst geringer mechanischer Belastung der Gelenkmotoren sowie die Vermeidung von Kollisionen mit statischen oder bewegten Objekten<sup>14</sup>. In jüngster Zeit ist mit einer stärkeren Einbeziehung von Sensoren und der daraus folgenden dynamischen Variabilität der Aufgaben darüber hinaus die Problematik der Singularitätenvermeidung immer stärker in den Vordergrund gerückt [Zha95, Llo98]. Deren Ziel ist, daß der Roboter auf seiner Bahn nicht über eine kinematisch unerreichbare (oder in eine nicht eindeutig einer bestimmten Gelenkkonfiguration zuzuordnende) Stellung geführt wird, weil dies unter Umständen unbegrenzte Gelenkgeschwindigkeiten bzw. -beschleunigungen zur Folge hätte. Gegenwärtig konzentriert sich das Interesse auf die Bewegungsplanung mobiler Roboter unter Einbezug von Sensorik, der ein wesentlicher Teil der Beiträge zu den einschlägigen Konferenzen gewidmet ist; zunehmendes Interesse finden kooperierende mobile Roboter [PO98, ŠO98]. Dies erklärt sich damit, daß die Probleme der Bewegungsplanung und Kollisionsvermeidung bei stationären Robotern zwar nicht für alle denkbaren, aber doch für die wichtigsten Einsatzfälle gelöst sind. Es existieren frei verfügbare Software-Bibliotheken, die für die Praxis ausreichend stabil, schnell und anpassungsfähig sind [Mir97].

<sup>11</sup>Siehe zu diesem Problembereich auch [Mae90, EHW<sup>+</sup>92, Nil94, PC96, KQW97, MS97].

<sup>12</sup>Nichtlineare Planung von (nebenläufigen) Aktionen allein reicht dazu nicht aus, etwa wie in [MR91], weil damit das Problem der Aufgabenaufteilung nicht gelöst ist. Ein Beispiel ist das gemeinsame Tragen eines Stabes bei gleichzeitiger Forderung nach permanenter horizontaler Ausrichtung (Standardbeispiel für kooperierende, kraftgeregelter Roboter).

<sup>13</sup>Um dem Instrukteur das „Hineindenken“ in das dynamische Verhalten der Roboterarme zu erleichtern, sollte angestrebt werden, deren dynamisches Verhalten dem menschlichen Vorbild anzunähern. Beobachtungen dieses Verhaltens und Vorschläge zu seiner technischen Umsetzung finden sich in [Hol90, Bur96, HSFS99].

<sup>14</sup>Die Standard-Einführung ist [Lat91a], eine neuere Arbeit über die Bewegungsplanung für Industrieroboter ist [LC97]; in [YX96] wird ein Ansatz auf der Basis genetischer Algorithmen vorgeschlagen und simuliert.



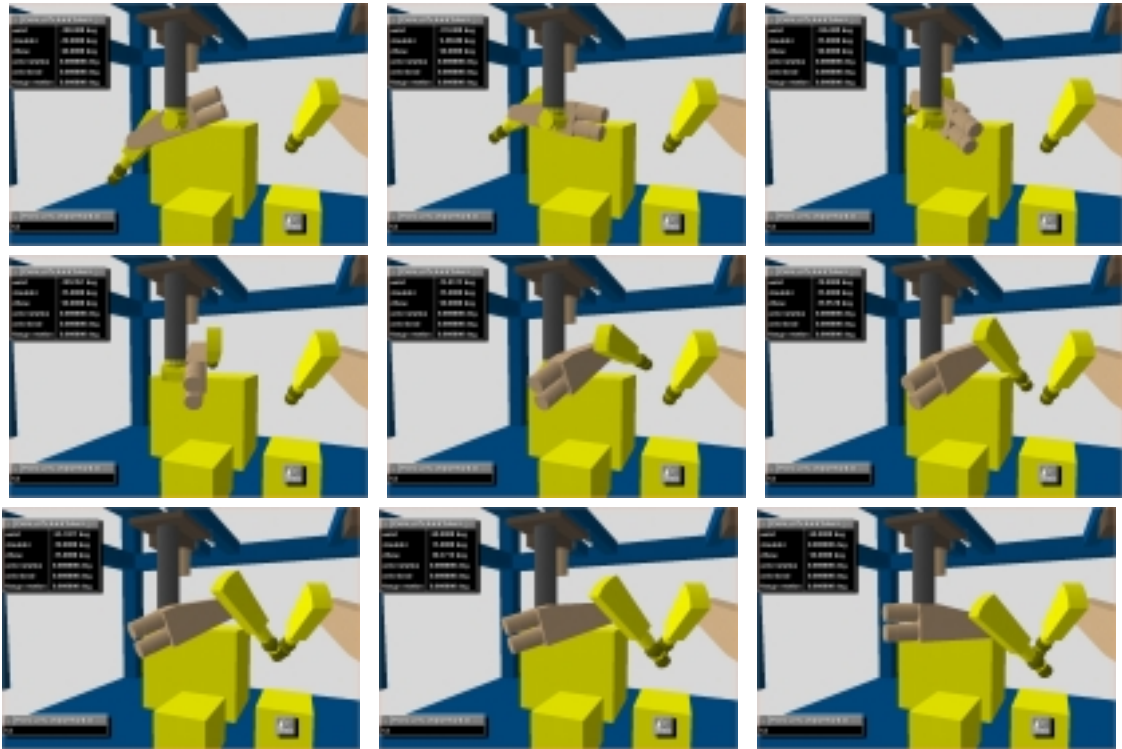


Abbildung 5: Ein kollisionsfreier Pfad führt entlang der Oberseite des großen Quaders zum Ziel unter dem zweiten Roboter (von links nach rechts, von oben nach unten).

Bei der *Griffplanung* ist festzuhalten, daß Planung im Sinne der Lösung eines kombinatorischen Problems nur bei mehrfingerigen Händen erforderlich ist (Bestimmung von Griffposturen und optimalen Griffpunkten, Berechnung von Fingerbewegungen), siehe dazu [MLS94, BFH99, Man99], welche aber bislang keine Verbreitung gefunden haben.

### 0.2.1.2 Sensoreinsatzplanung

Die Aufgabe der Sensoreinsatzplanung besteht in der Aktivierung von Sensoren und in der Einstellung ihrer Parameter mit dem Ziel der Lösung einer Erkennungsaufgabe bei Erzielung vorgeschriebener Erkennungsqualität und ggf. einer Obergrenze für entstehende Kosten. Solche Parameter sind der Sensorstandort bzw. der Sichtwinkel (bei bewegbaren bzw. mitbewegten Sensoren), die Empfindlichkeit und der fokussierte Bereich (Festlegung einer *region-of-interest*). Es geht hier darum, diese Parameter in der Wechselwirkung zwischen Erkennungsaufgabe, Objektstruktur und Sensorcharakteristik möglichst weitgehend automatisch zu bestimmen und dementsprechend einen Plan zu generieren, wann und wo welcher Sensor mit welchen Parametern zu aktivieren ist. Eine solche Planung ist die Voraussetzung für eine sinnvolle Fusion der redundanten Informationen aus den einzelnen Sensoren. Die Anzahl von Arbeiten zu diesem Thema ist noch relativ klein, und sie beschränkt sich auf die Planung von optischen Sensoren, d.h. Laser-Entfernungsmesser und Kameras bzw. die dazugehörigen Lichtquellen. In der Übersicht [TAT95] werden als Eingangsgrößen für generische Sensorplanungssysteme Sensormodelle (einschließlich der Wirkung von Beleuchtung), Objektmodelle und die Beschreibung der Erkennungsaufgabe (Objekterkennung, Szenenrekonstruktion, Merkmalsextraktion) genannt. Ausgangsgrößen sind die Blickrichtung der Kamera, ihre optischen Parameter und die Strahlrichtung der Beleuchtungsquelle. In [AAT99] wird dies wieder aufgegriffen; auch hier geht die Systematik jedoch nicht über die Aufstellung einiger Heuristiken hinaus. In [GI94] wird untersucht, welche Ansichten eines Passagierflugzeugs zu einem guten Erkennungsergebnis beitragen können. In [TTA95] wird ein implementiertes System zur Erkennung eines einfachen mechanischen Aggregats vorgestellt. Dazu werden verschiedene *constraints* definiert, die ein optimaler Satz von Kameraparametern erfüllen muß (Sichtbarkeit vs. Teilverdeckung, Auflösung, Entfernung für den Fo-

kuspunkt, Öffnungswinkel) und danach wird dieser berechnet. Ein Zusammenhang mit einer Montageaufgabe, also die Planung unter der Randbedingung, daß eine Aktion zur sensorischen Erfassung einen möglichst großen Fortschritt im Hinblick auf die Steuerung einer Teiloperation bringt, wird nicht hergestellt. Letzteres ist jedoch das Ziel in [MI98]. Basierend auf dem Wissen um die für die Montage relevanten Kontaktflächen, auf dem Wissen um die Extraktionsfähigkeit des Sensors bezüglich der Objektmerkmale und einer Bewertungsfunktion für den zu erwarteten Erfolg einer Sensoroperation wird ein Plan erzeugt. Dieser Ansatz wird für einen Laser-Entfernungsmesser, der um eine Tischfläche rotiert und in seiner Höhe verändert werden kann, implementiert. In [CA99] wird dargestellt, wie allein basierend auf einer Objektansicht eine Folge von *visual tasks* erzeugt werden kann, die ein Roboter, an dessen Hand eine Kamera befestigt ist, dann ausführt.

Ein weiteres Einsatzgebiet ist die Überprüfung von Teilen auf Einhaltung von Toleranzen [TUWR97, YMC98, GMC99]. Solche Teile können im Prinzip ungeordnet und in unvorhersehbarer Position angeliefert werden. Interessant ist hier die Bewertungsfunktion für die beste Sicht; sie wird üblicherweise probabilistisch oder über eine Fuzzy-Regelbasis [KAF99] modelliert. In [Lee90] wurde vorgeschlagen, sie so anzusetzen, daß diejenige Sicht des Objekts (bzw. derjenige Sensor) ausgewählt wird, die/der die größte Reduktion von Vieldeutigkeiten bei der Interpretation des interessierenden Bildausschnitts erzielt.

In [Erd95] wird argumentiert, daß es gar nicht erforderlich ist, daß ein Sensor den gesamten Zustand der Umgebung oder eines Objekts erfaßt, sondern daß es reicht, wenn er den Aktionsplaner in die Lage versetzt über die jeweils nächste mögliche Aktion zu informieren. Die Sensoren werden als ideal angenommen, ihre Meßunsicherheit wird auf resultierende Ungenauigkeit der durch den Sensor gesteuerten aktorischen Operation abgebildet und deren Abfolge durch *backchaining* geplant.

Es bleibt festzuhalten, daß es auf der Ebene der reinen Ortsplanung und auch für die Aktivierung von Sensoren in Abhängigkeit von einer Montageaufgabe Ansätze gibt. Die Beeinflussung der einzelnen Verarbeitungsschritte über die Wahl der optischen Parameter hinaus, d.h. Vorverarbeitung bis Bilddeutung, unter einer Bewertungsfunktion für die entstehenden Kosten (Bewegungszeit, Verarbeitungszeit, Sensorbelastung) ist mit Blick auf die Datenfusion und die Verkettung mit der Handlungsplanung erforderlich, aber noch nicht untersucht worden. Abgesehen von ersten Ansätzen zum „Nachschauen“ zur Vermeidung von Verdeckungen [MH98] ist auch die Problematik der zielgerichteten Exploration (etwa das Herumgehen um ein Objekt, das kontrollierte Bewegungen eines Aggregats vor einer oder mehreren Kameras) noch weitgehend unbearbeitet.

### 0.2.1.3 Datenfusion

Aufgabe der Sensorsteuerung ist es, für einen bestimmten interessierenden Weltausschnitt die zur Verfügung stehenden Informationen zu sammeln und so aufzubereiten, daß sie in ihrer Gesamtheit für die Lösung einer Erkennungs- oder Steuerungsaufgabe genutzt werden können. Das Ziel der Datenfusion<sup>15</sup> ist, Informationen aus verschiedenen Quellen (Teilansichten), die bezüglich ihres Gehalts redundant sind, zu einem einheitlichen Bild zu verschmelzen, d.h. Widersprüche aufzulösen und Lücken zu füllen.

Gegenstand ist beispielsweise die Zusammenfügung zweier Bilder, die beide einen begrenzten Blickwinkel wiedergeben, zu einem „Panoramablick“ mit dem summierten Blickwinkel. Hier ist für den Ausgleich der Kamera- und Perspektivenverzerrungen und die Anpassung an den beiden zusammenzufügenden Bildrändern zu sorgen. Eine komplexere Aufgabe der Fusion ist die Rekonstruktion von Objekten, die von mehreren Kameras an unterschiedlichen Standorten aufgenommen werden. Dabei sollen die Objektparameter (Position, Orientierung, Form, Farbe) genauer bestimmt werden, als das mit einer Kamera allein möglich wäre.

Das Zusammenfügen von mehreren Bildern unterschiedlicher Weltausschnitte wird schon seit einiger Zeit untersucht, weniger allerdings im Zusammenhang mit der Robotik, vielmehr für Szenen im freien Gelände oder in Büroräumen (neuere Arbeiten sind hier [SK97a, IA97, LWG97]). Die Problematik der Datenfusion ist in der Robotik beginnend mit den frühen Arbeiten [DW87] und [HM91] hauptsächlich unter dem

---

<sup>15</sup>Es herrscht ein gewisses Maß an semantischer Verwirrung um den Begriff der Datenfusion vs. Datenintegration, Datenkombination, Datenaggregation und weitere. Wir verweisen auf [CM99a, CM99b] für eine informelle Klärung der Begriffe und verwenden nur den Begriff Datenfusion.

Aspekt der statistischen Modellierung von Sensoren [BM98] und teilweise auch von Objekten [Por88] untersucht worden, inspiriert auch vom Problem der Verfolgung von beweglichen Objekten mit Sensoren an unterschiedlichen Standorten [BSF88, RDWS93]. Es kommen Kombinationen ganz unterschiedlicher Sensoren zum Einsatz: Kameras, Laser-Entfernungsmesser, Ultraschallsensoren, Formerkennung mit strukturiertem Licht, taktile Sensoren, usw. Methodisch wurde dabei auf Entscheidungstheorie [Ber85, Das93, KMM96], lineare und nichtlineare Kalman-Filter [MDW95, VB97, Mut98], die Kombinationsregel nach Dempster-Shafer [Bog87, HK89, Mur98], Bayes-Netze [CLZ96, Pea97] und Modelloptimierung [JS99] gesetzt. Es gibt aber auch Verfahren, die das Problem auf die Minimierung einer für die unterschiedlichen Sensoren zu definierenden Energiefunktion [CY90] oder auf die Repräsentation der Sensoreigenschaften auf eine Fuzzy-Regelbasis abbilden [AG92, MBF98, ZSK98, KAF99, ZSK99]. Schließlich wurde im Zusammenhang mit der Steuerung mobiler Roboter bereits sehr früh damit begonnen, künstliche neuronale Netze, deren Eingangsschicht direkt mit einfachen eindimensionalen Sensoren verbunden war, auf die direkte Aktorsteuerung in Abhängigkeit von den unterschiedlichen Meßwerten zu trainieren [Pom93, BP97, HLSvS98]. Interessanterweise wurde jedoch das Problem der Verbesserung der Szenenrekonstruktion bei Vorhandensein mehrerer Kamerasichten aus unterschiedlichen Blickwinkeln und der dadurch tatsächlich erreichbare Fusionsgewinn kaum betrachtet (eine Ausnahme ist [LWG97], einen Ansatz für die Fusion von Entfernungsbildern präsentiert [DWJM98]), obwohl spätestens seit den Arbeiten [FH86, Aya88, AF88] das Problem formuliert und ein statistischer Ansatz in Form eines Kalman-Filters vorhanden war. Die algorithmische Seite der Datenfusion ist also – auch durch Anpassung von Arbeiten aus anderen Gebieten, speziell der Statistik – bereits recht ausgereift. Bislang sind allerdings zwei Schlüsselprobleme nur unbefriedigend gelöst: die *Sensormodellierung*, die Voraussetzung für eine genaue Bestimmung der Objektparameter ist, und die die Erkennungsaufgabe bestmöglich lösende automatische Extraktion wichtiger Szeneneigenschaften (*Informationsbewertung*):

- Für die erfolgreiche Anwendung der statistischen Verfahren sind Modelle der Fähigkeiten von Sensoren erforderlich, die über eine einfache quantitative Angabe der vom Sensor zu erwartenden Unsicherheit hinausgehen (siehe [WI95]); es zeigt sich darüber hinaus, daß die übliche Normalverteilungsannahme für das Sensorrauschen *in der Praxis* nur in den seltensten Fällen zu befriedigenden Ergebnissen führt.
- Offensichtlich kann die Fusion nicht generell auf der Ebene von Bildpunkten (auf der Signalebene) stattfinden, sie muß vielmehr auf der Ebene von Bildmerkmalen oder auch Objekteigenschaften durchgeführt werden [BBW<sup>+</sup>97]. Die autonome Auswahl der adäquaten Ebene, die Auswahl der Merkmale, die Einschätzung des Beitrags der einzelnen Sensoren *a priori*, insgesamt also die Extraktion der für die Erkennungsaufgabe wesentlichen Information über eine Bewertungsfunktion sind bislang noch wenig untersuchte Fragestellungen.

Insgesamt besteht bezüglich der beiden letzten Punkte noch erheblicher Forschungsbedarf, bevor das Potential der Datenfusion voll genutzt werden kann.<sup>16</sup>

#### 0.2.1.4 Erwerb und Generalisierung von Fertigkeiten

Die Programmierung von Robotern durch menschliches „Vormachen“ und „Abschauen“, also das *Erlernen von Montagesequenzen*, ist seit langem Ziel der Robotikforschung. Roboter zum Sprühlackieren oder für bestimmte Schweißoperationen, die strukturell einfache Bewegungen machen müssen, bei denen aber Parameter wie Form oder Geschwindigkeitsprofil einzuhalten sind, konnten bereits Ende der siebziger Jahre durch einfache Führung gelehrt werden [Tod86]. Im Prinzip gilt dies auch für die Anwendungen der Telemanipulatoren [She92]. Für die Durchführung von komplexen Montageoperationen hingegen gibt es bis heute kein praktisch einsetzbares System, welches rein oder zum überwiegenden Teil durch Verfolgung der Bewegungen eines menschlichen Instruktors Handhabungsoperationen lernt (oder auch nur parametrisierte instantiiert).

Einer der ersten Versuche, optisch erfaßte menschliche Operationen auf Roboter zu übertragen war [KII92]<sup>17</sup>: Ein Instrukteur machte auf einem Tisch einfache *pick-and-place*-Operationen mit kleinen Klötzchen vor, der

<sup>16</sup>Siehe auch die generellen Bemerkungen zur Frage: „*was ist wann wie zu fusionieren*“ in [DT99].

<sup>17</sup>Eine ähnliche Arbeit ist [IS94, KI95].

Roboter sollte dieselben Manipulationen vornehmen. Weitere Arbeiten zum rein optischen Vormachen sind [PB94]; ein Vorschlag für eine umfassende Architektur findet sich in [SNI<sup>+</sup>94]<sup>18</sup>; eine Möglichkeit, für diesen Lernvorgang Hidden-Markov-Modelle (HMM) zu trainieren, stellt [YXC94, XY95] vor. Ansätze allerdings, die kompliziertere Operationen durch reines Zuschauen erlernen wollten, sind bislang nicht sehr erfolgreich gewesen. Der Grund ist darin zu suchen, daß damit weder die „innere Struktur“ des zu montierenden Aggregats (geeignete Montageflächen, Griffpunkte, Schwerpunkte) noch die nicht-optischen Größen (Kraft, Moment) erkannt werden können. Gleiches gilt für Absichten des Instruktors bzw. sein in möglicherweise nicht sichtbaren Aktionen verstecktes Erfahrungswissen.

Erfolgversprechender sind daher die Versuche, durch Führung des Roboters zu lehren. Dies geschieht z.B. mit Hilfe eines Kraftsensors [Kai97], durch Steuerung über manuelle Gesten/Anweisungen [SWR99, BKM<sup>+</sup>98], durch introspektive Beschreibung der Zieloperation aus Sicht des Instruktors [BU97] oder auch durch Vorlage eines gedruckten Bauplans und Analyse durch das Robotersystem (ein erster Ansatz wird in [ATT99] vorgestellt). [Mye99] berichtet über ein System, das durch die Führung des Roboters im *master-slave*-Modus, durch seine Bewegung im *zero-gravity*-Modus<sup>19</sup> oder durch Abschauen ein Roboterprogramm in einer konventionellen Programmiersprache erzeugt, das schrittweise verfeinert werden kann. In [MTYN99] wird eine Entwicklung vorgestellt, bei der ein Fräs-Zentrum direkt durch die Verfolgung der Fingerbewegungen eines Instruktors gesteuert wird. Damit entfällt der Einsatz von Steuerknüppeln oder speziellen mehrdimensionalen Bewegungsaufnehmern. Komplexe Bewegungszusammenhänge und Aktionen sind hiermit allerdings kaum zu instruieren. Dies ist bei dem Ansatz nach [VK99] eher zu erwarten: hier werden vordefinierte Fertigkeiten durch bestimmte Gesten ausgelöst. Das Hauptproblem wird dabei als das des Findens von Übereinstimmungen identifiziert: Welche vordefinierte (Teil-)Fertigkeit des Roboters erreicht am ehesten das, was der menschliche Instrukteur mit seiner Geste erzielen wollte? Dabei hilft es dem Fluß der Kooperation zwischen Instrukteur und Roboter ganz wesentlich, wenn das Robotersystem über das reine Abschauen hinaus aus der Erfahrung, die es mit einem speziellen Instrukteur gemacht hat, bestimmte Aktionen extrapolieren kann. Eine derartige Vorausschau läßt sich zweckmäßig über HMMs realisieren [PL97, YXC97, YUDS99].

Insgesamt stecken die Arbeiten zum Lernen durch Vormachen noch in der Anfangsphase, und speziell die Übertragung von solcherart gelernten Operationen auf andere Situationen (*skill abstraction*) oder gar die Nutzbarmachung bestimmter Teile von Operationen für andere Fertigkeiten (*skill transformation*) ist bislang kaum angegangen worden.

#### 0.2.1.5 Fazit

Wie in den vorigen Abschnitten dargelegt, gibt es für die verschiedenen Fragestellungen der Robotik unterschiedlich ausführliche Antworten aus dem Bereich der KI. Es ist inzwischen weitgehend anerkannt, daß sich die komplexen Wechselwirkungen in der Relation *Mensch – Robotersystem – Umwelt* nur bedingt simulieren lassen und theoretische Modellbildungen immer nur einen eng umgrenzten Ausschnitt dieses Dreiecks erfassen. Es ist deshalb unumgänglich, zur aufeinander bezogenen Untersuchung der relevanten Fragestellungen ein funktionsfähiges System aufzubauen, anhand dessen sich Modellannahmen verifizieren (und ggf. Simulationen kalibrieren) lassen. Nur über eine solche Verkörperung können die relevanten Phänomene in ihrer Vielfalt untersucht werden; der Aufbau eines arbeitsfähigen Systems erschließt das interdisziplinäre Potential der Robotik, und er geht deshalb weit über die Problematik der reinen Systemintegration hinaus. Im folgenden Abschnitt konzentrieren wir uns auf die Darstellung eines solchen Systems für die Untersuchung zukünftiger Handhabungssysteme mit stationären kooperierenden Robotern.

---

<sup>18</sup>Eine umfangreiche Referenz ist [IV97].

<sup>19</sup>Der Roboter gleicht die Gewichte seiner Armsegmente durch Gegensteuern der Gelenkmotoren gerade in dem Maße aus, wie es erforderlich ist, ihn mit (fast) verschwindender Kraft durch eine Menschen zu führen. Für eine geschmeidige Bewegung ist dazu ein komplexes internes Modell der Roboterstatik und -dynamik erforderlich, darüber hinaus müssen die vom Menschen ausgeübten Führungskräfte durch einen empfindlichen Kraft-Momentensensor an der Roboterhand gemessen werden können.



Abbildung 6: Unterschiedliche Ausgangssituation am Beginn der Montage (aus denen unterschiedliche Montagefolgen resultieren). Die Abbildungen zeigen jeweils dieselbe Objektmenge.

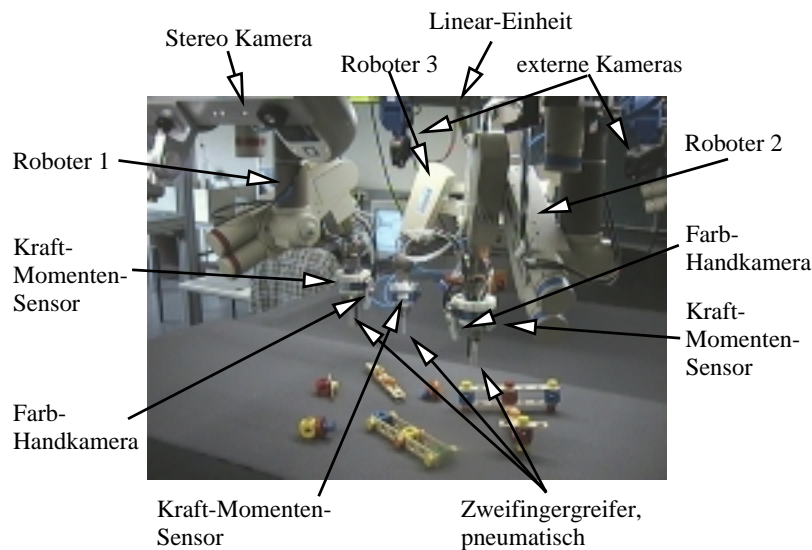


Abbildung 7: Laboraufbau eines Systems kooperierender Roboter mit umfangreicher Sensorausstattung. Die hängend montierten Roboter sind Industriearbeitungen mit stark modifizierter Steuerung.

## 0.2.2 Ein praktisch realisiertes System

Wie so oft in der Beschreibung der Struktur komplexer Systeme (etwa zur Bild- oder Sprachverarbeitung) erscheint es uns auch hier besonders zeckmäßig die Aufgabenstellungen und die zum Tragen kommende Methodik an einem konkret implementierten System zu erläutern. Wir umreißen deshalb in diesem Abschnitt ein Robotersystem<sup>20</sup> für Montageaufgaben, das im Labormaßstab robust arbeitet und das wesentliche Fertigkeiten integriert, die zukünftige stationäre Robotersysteme aller Voraussicht nach aufweisen werden (Abbildung 7). Ausgegangen wird dabei von einem Spielzeug-Szenario: die Aufgabe besteht darin, aus einer mehr oder weniger geordneten Menge von „Baufix“-Elementen, die beliebig auf einem Tisch liegen, zunächst einzelne Aggregate und danach aus diesen fertige Modellobjekte (Flugzeug, Motorroller, etc.) zu bauen.<sup>21</sup> Mit anderen Worten:

<sup>20</sup>Dieses System wurde an der Universität Bielefeld entwickelt, teilweise im Rahmen des DFG-Sonderforschungsbereichs 360. Es ist uns kein anderes System bekannt, das einen ähnlichen hohen Grad der Integration von Instruktion, Kognition und Aktion aufweist.

<sup>21</sup>Baufix-Holzspielkästen sind nach Aussage des Herstellers für Kinder ab drei Jahren geeignet – was nach unserem Eindruck eine eher optimistische Einschätzung ist. Immerhin: es erscheint uns bemerkenswert, daß hier vom Robotersystem Leistungen erbracht werden müssen, für deren Entwicklung der Mensch als Individuum mindestens drei Jahre benötigt. Aus Sicht der Robotik hat die Wahl dieser Domäne darüber hinaus den wesentlichen Vorteil, daß sie von der Beschaffenheit der Teile her mit modifizierten Industrierobotern noch handhabbar ist und dennoch alle wesentlichen Operationen beinhaltet, die Montageszenarien beinhalten können – ein ganz wesentlicher Fortschritt gegenüber klassischen Szenarien, die sich mit *pick-and-place*-Operation, im besten Fall erweitert um *peg-in-hole* begnügen.

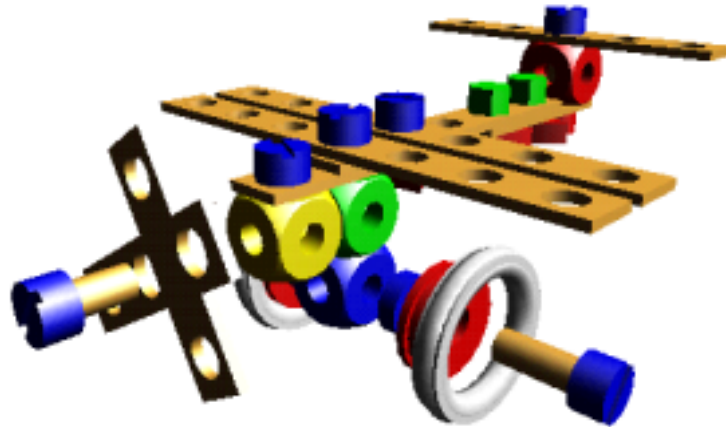


Abbildung 8: Das fertige Modellflugzeug, wie es aus den Bauteilen nach Abbildung 6 zu konstruieren ist.

Aus einem Angebot von Bauelementen nach Abbildung 6 soll beispielsweise ein Modell entstehen, wie es Abbildung 8 zeigt. Der gesamte Konstruktionsprozeß wird dabei ausschließlich durch einen multimodalen Dialog gesteuert, und die Lage der Teile wird rein optisch erfaßt. Besonderer Wert wird dabei auf Unabhängigkeit der Konzepte und soweit wie möglich auch der Implementierung von der konkreten Ausformung der Domäne gelegt: es werden keine speziellen Werkstückaufnahmen oder Greifer eingesetzt, alle Montageoperationen werden stattdessen von zwei kooperierenden Armen mit umfangreicher Kraft- und Sichtsensorik ausgeführt. Der Ablauf ist dabei wie folgt: Dem menschlichen Instrukteur steht ein Konstruktionsplan zur Verfügung, den der Konstrukteur (das Robotersystem) nicht einsehen kann. Der Instrukteur erteilt dem Konstrukteur anhand dieses Konstruktionsplans Anweisungen. Die von den Perzeptions- und Kognitionsschichten gelieferten Ergebnisse werden dann in Aktionen umgesetzt, in denen sich die Intentionen des menschlichen Instrukteurs spiegeln. Drei Systemkomponenten kommt dabei wesentliche Bedeutung zu:

1. **Dialogkomponente zur Instruktion.** Instrukteur  $I$  und Konstrukteur  $K$  nehmen denselben Umwelt-ausschnitt wahr und beziehen auf ihn ihre Handlungen und sprachlichen Äußerungen. Deshalb ist die integrative und kohärente Repräsentation von Objekten, Ereignissen und Sachverhalten sowie den darauf aufbauenden Verstehensprozessen wichtig. Basis der Dialogführung ist die sprachliche Äußerung; weitere Modalitäten (speziell Gestik) werden hinzugezogen, falls dazu die Notwendigkeit besteht (bei Zweifeln, Mehrdeutigkeiten, etc.).
2. **Robuste Sensorbasis und Verteilte Steuerung.** Um der Forderung nach robuster Perzeption bei allen vorkommenden Objektanordnungen und in möglichst allen Umweltsituationen nachkommen zu können, verfügt das System über eine Vielfalt von z.T. redundanten stationären, beweglichen und mitbewegten Sensoren. Damit können eine ganze Reihe von inhärenten Ungenauigkeiten berücksichtigt werden: ungenaue Spezifikation des einzelnen Montageschritts; geringe Reproduzierbarkeit der Ausgangssituation; Begrenzungen in der genauen sensorischen Erfassbarkeit des augenblicklichen Zustands; begrenzte Präzision des Roboters. Sensoren und Aktoren werden als Agenten in einem Multi-Agenten-System modelliert; die Auswahl der in einer jeweiligen Situation erforderlichen sensorischen, aktorischen oder kombinierten Aktion erfolgt über das Vertragsnetzprotokoll. Damit wird ein hohes Maß an Fehlertoleranz erreicht; Sensoren und Aktoren können zur Laufzeit des Gesamtsystems hinzugefügt und außer Betrieb genommen werden.
3. **Flexible Montageoperationen.** Während die berührungsfrei ablaufenden Transportphasen der einzelnen Montageschritte einfach realisiert werden können (es sind lediglich Kollisionen zu vermeiden), verlangen die Kontakt- und Fügephasen ein komplexes Zusammenspiel von Kraft-, Tast- und visueller Sensorik mit der Steuerung bzw. Regelung der Bewegungsfreiheitsgrade des Roboters. Eine wichtige Voraussetzung



zur Nachbildung menschlicher Manipulationsfähigkeiten ist die Konstruktion eines Rahmens, innerhalb dessen ein Grundrepertoire an elementaren Bewegungsfähigkeiten flexibel kombinierbar ist. Die elementaren Montageprimitiva müssen dabei ihrerseits durch eine direkte Sensorkopplung so robust implementiert werden, daß eine sichere Regelung der einzelnen Parameter auch bei sehr langen Folgen (also bei der Konstruktion komplexer Aggregate) gewährleistet bleibt.

Wir betrachten diese Komponenten im folgenden noch etwas eingehender. Zunächst in Abschnitt 0.2.2.1 die Komponente zur Handlungssteuerung im Dialog mit dem Menschen, danach in Abschnitt 0.2.2.2 die sensorische Basis, in 0.2.2.3 die motorischen und sensomotorischen Fähigkeiten und schließlich in 0.2.2.4 die Steuerung des Gesamtsystems durch ein Multi-Agenten-System.

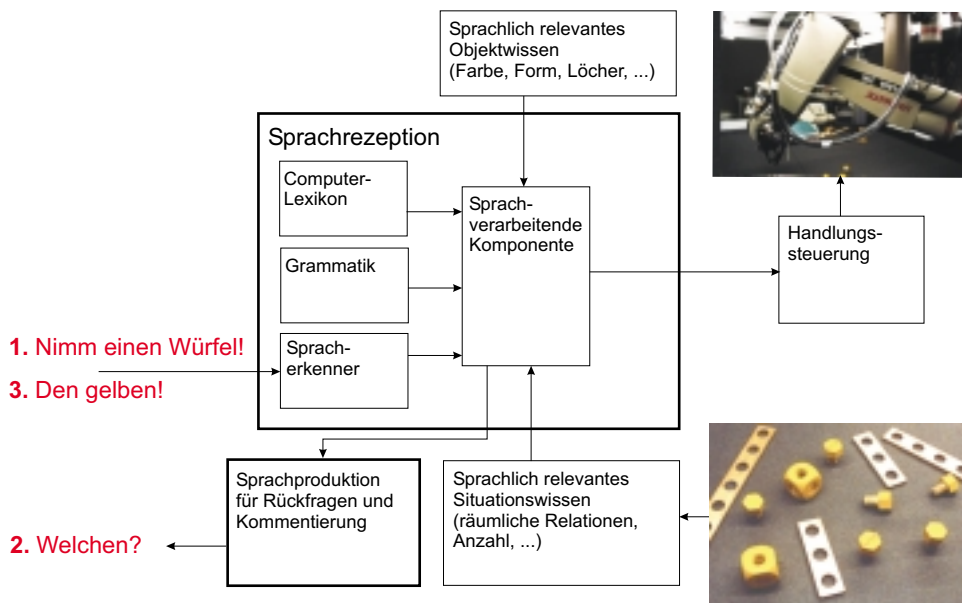


Abbildung 9: Grobstruktur der Dialogkomponente (nur direkt sprachgebundener Teil)

### 0.2.2.1 Dialogkomponente

Der Dialogkomponente kommt insofern zentrale Bedeutung zu, als sie den gesamten Ablauf des Montageprozesses in Abhängigkeit von sprachlichen Äußerungen steuert: sie interpretiert die geäußerten Intentionen des Instruktors  $I$  in Abhängigkeit von Umwelt- und innerem Systemzustand und aktiviert damit sowohl die Aktorik als auch die Sprachproduktion (für Handlungskommentierungen, Rückfragen, Fehlermeldungen). Um ein flüssiges und seitens des Instruktors  $I$  nachvollziehbares Arbeiten mit dem System zu ermöglichen, sollte sie mindestens zwei Verarbeitungsprinzipien berücksichtigen: Inkrementalität und Interaktivität.

- Bei der *inkrementellen* Verarbeitung wird Information möglichst umgehend verarbeitet, ohne diese zuvor zu sammeln oder auf bestimmte Schlüsselinformationen zu warten. Bezüglich der Sprachverarbeitung heißt dies, daß Wörter in der Regel unmittelbar und interaktiv verarbeitet werden, ohne zuvor auf das Satzende zu warten (z.B. [RH97]). Verarbeitungsrelevante Informationseinheiten, die Inkremente, können neben Wörtern auch Konstituenten im syntaktischen Bereich oder Konzepte bzw. Referenten im mentalen Modell im semantischen Bereich sein. Inkremente im visuellen Bereich können objektbezogene Regionen, die durch Farbe oder Form eine Einheit bilden, oder Aufmerksamkeitsbereiche sein.
- Eng verknüpft mit der inkrementellen Verarbeitung ist das Prinzip der *interaktiven* Verarbeitung, bei der unterschiedliche Informationsbereiche frühzeitig berücksichtigt werden. Dies können neben Interaktionen zwischen syntaktischer und semantischer Information auch Interaktionen zwischen sprachlicher und



visueller Information sein. In welchem Maße und unter welchen Bedingungen sprachliche Verstehensprozesse mit den verschiedenen Informationsbereichen interagieren, ist größtenteils noch unerforscht. Dennoch muß ein kognitiv motiviertes Robotersystem Möglichkeiten zur Interaktion und Integration vorsehen; technisch bedeutet dies unter anderem die Fähigkeit zur Datenfusion.

**Technische Implementierung.** Die Hauptkomponenten der Dialogsteuerung sind ein Spracherkenner und ein Sprachverstehensmodul (siehe Abbildung 9). Um die Erkennungsleistung des Spracherkenners zu erhöhen, wurde ein vorhandenes Vollformen-Lexikon dem Vokabular des Szenarios angepaßt und das System um grammatische Bedingungen und einfache Wortkompositionsregeln erweitert [RHZ99]. Der Spracherkenner ist als „Frontend“ der Sprachverstehenskomponente angelegt. Falls eine Äußerung nur partiell korrekt erkannt wurde, ist in der Regel dennoch eine zumindest teilweise Interpretation möglich.

Die Sprachverstehenskomponente basiert auf einer Variante der Kategorialgrammatik, der *Combinatory Categorical Grammar* ([Ste87, Ste93a, Ste96]). Um kognitiven Verarbeitungsprinzipien der Sprachrezeption, wie z.B. der inkrementellen Verarbeitung und der flexiblen Konstituentenstellung im Deutschen zu entsprechen, wurde die Grammatik modifiziert und an das Deutsche angepaßt [HR97, HE99]). Dadurch können selbst komplexe und (lokal) ambige Anweisungen (z.B. *steck die lange Schraube in das zweite Loch von links der siebenlöchrigen Leiste*) effizient interpretiert werden. Aus den sprachlichen Anweisungen leitet die Sprachverstehenskomponente semantische Strukturen ab, aus denen wiederum elementare Handlungsanweisungen für die Akteure generiert werden. Die Interpretation einer Anweisung basiert zusätzlich zum linguistischen Wissen auch auf konzeptuellem Wissen über Objekte und Handlungen.

Tabelle 1 zeigt einen *Beispieldialog* (aus Platzgründen geben wir hier nur den Beginn wieder), der sich mit dem System abwickeln läßt. Ziel ist die Konstruktion des Höhenleitwerks des Modellflugzeugs. Man erkennt, daß bereits diese relativ einfache Montageabfolge eine Vielzahl von sensorisch-kognitiven Fähigkeiten erfordert, die sich bei Zulassen der angedeuteten Alternativen noch vervielfacht. Während der Handlungsinterpretation und -ausführung kommt es zu einem komplexen Interaktionsgeflecht zwischen den Modulen, wobei die zugrundeliegenden Verarbeitungsprinzipien die Basis für ein robustes System bilden, das unerwartete, unvollständige und auch widersprüchliche Information verarbeiten kann, ohne handlungsunfähig zu werden.

Gegenwärtig kann kaum eine konstante Wortakkuratheit von mehr als 85% von einem Spracherkenner erwartet werden. Daraus folgt, daß das Sprachverstehenssystem lexikalische Einfügungen, Ersetzungen und Auslassungen bewältigen muß. Außerdem bildet eine Äußerung nicht immer einen Satz, zumindest nicht gemäß einer üblichen Grammatik.

Äußerung	Alternativen	Bemerkungen
Instrukteur: Bist Du bereit?  Initialisierung		
Konstrukteur: <b>Ja, los geht's!</b>  Bestätigung der Initialisierung	(i) Nein, heute nicht! (ii) Einen Moment bitte! [meine Hände müssen erst frei sein] . . .	
Heute wollen wir ein [Baufix-] Flugzeug [zusammen] bauen [, wir beginnen mit dem Leitwerk]!  Problemspezifikation		– Nur sinnvoll, wenn Wissen über Objekt-domäne bereits vorhanden – “Baufix” spezifiziert im Gegensatz zu “Lockheed” spezielle Domäneneigenschaften – “Bauen . . . ” fokussiert auf Objekt, “Zusammen bauen . . . ” fokussiert auf Kooperation
<i>Weiter im nächsten Tabellenabschnitt</i>		

Äußerung	Alternativen	Bemerkungen
<b>Gut!</b> Bestätigung Problemspezifikation	(i) Ich weiß nichts über diese Flugzeuge! (ii) Schon wieder!	– Alternative Lehren: “Ein Flugzeug ist ...” oder: Diskussion des Bauplans
<b>Ich überprüfe, ob alle Bauteile da sind.</b> Vorbedingung für Handlung	Ich brauche noch eine weitere Schraube [ , Würfel ,...].	Vorbedingungen: – Gemeinsamer Sichtbereich $I/K$ – $K$ verfügt über Objektwissen
<b>Wie würdest Du das [recht-eckige] Objekt [in der oberen linken Ecke, zu deiner linken, zu meiner linken ...] bezeichnen?</b> Verhandlung Objektbenennung		Vorbedingungen: – Gemeinsamer Sichtbereich $I/K$ – Verhandlung nur sinnvoll, wenn $K$ mindestens über ein abstraktes Objektmodell verfügt
Dies ist ein Würfel! <zeigt auf ihn> Objektbenennung		
<b>Gut, dann nennen wir es so!</b> Akzeptiere Objektbenennung		Geschlecht und weitere Eigenschaften nur bei Vorhandensein im Lexikon ableitbar
<b>Nimm eine Schraube!</b> Instruktion	Du brauchst [erst] eine Schraube!	– Objekt unbestimmt, $K$ sucht eines aus, das er erkannt hat – Alternative ist eine unbestimmte Instruktion, die nicht sofort ausgeführt werden muß (aber vor allen anderen)
<b>Ich nehme sie!</b> Handlungskommentar	(i) ... + mit meiner linken Hand (ii) Ich sehe keine Schraube	(i) $K$ liefert die räumlich Relation seiner Hand mit $K$ (ii) Fehlermeldung
<b>[Jetzt] nimm die Lochleiste!</b> Instruktion	... + mit drei Löchern!	Bestimmte Objektauswahl nur möglich, wenn nur eines dieser Objekte in der Szene.
<b>Ich sehe mehr als eine [dieser] Schraube[n].</b> Identifikation von Widersprüchen oder Vieldeutigkeiten durch $K$	(i) ... + Ich würde [lieber] die oberste nehmen. (ii) ... + welche von denen [die ich sehe] soll ich nehmen?	(i) $K$ realisiert seine volle Autonomie (ii) $K$ produziert zwei Äußerungen: Problemspezifikation und Informationsnachfrage (nach Objektspezifikationen).
<b>Nimm diese da!</b> <zeigt auf eine> $I$ löst Widerspruch auf	(i) Nimm die, auf die ich zeige! (ii) Nimm die links von {mir dir}! (iii) Nimm die, die du willst! <und/oder passende Geste> (iv) Nimm die rechte!	(i) Stellt sicher, daß $I$ und $K$ sich auf dasselbe Objekt beziehen (ii) Benötigt Bezugsrahmen (und Info über Position von $I$ ) (iii) z.B. Nicken (iv) Position statt Form und Farbe
<b>Ich habe sie.</b> Bestätigung der Handlung	Soll ich jetzt schrauben?	Antizipation der wahrscheinlichsten Folgeaktion
<b>Steck die Schraube auf die Leiste!</b> (Fehlerhafte) Instruktion	(i) Steck die Schraube durch das Loch! (ii) Steck die Leiste über die Schraube!	Rolle und Objektfunktion stimmen nicht überein
+ Weitere Äußerungen bis zur vollständigen Konstruktion		

Tabelle 1: Beginn eines Beispieldialogs zum Aufbau eines einfachen Aggregats (Leitwerk des Baufix-Flugzeugs). Die Äußerungen des Menschen (Instruktors  $I$ ) sind serifenlos gesetzt, die des Robotersystems (Konstruktors  $K$ ) in **nicht-proportionaler** Schrift.

Ein Sprachverstehenssystem muß also selbst bei einem perfekten Spracherkenner robust bezüglich jeglicher sprachlicher Eingabe sein. Insgesamt muß das System eine Reihe qualitativ unterschiedlicher Schwierigkeiten überwinden, die im folgenden ausgeführt werden.

- **Wortersetzungen:** Der Spracherkenner ist oft unsicher bezüglich der Flexionsendungen *-em* und *-en*, so daß beide leicht vertauscht werden. Da durch die Flexion u.a. Kasus und Genus markiert sind, kann bei einer Vertauschung die vorliegende Nominalphrase nicht als syntaktisch kongruent analysiert werden. Als Fehlerbehandlungsroutine liegt es daher nahe, auf Kasus- und Genuskongruenz innerhalb einer Nominalkonstituente ganz zu verzichten, wenn eine konsistente alternative Konstituente fehlt. Um Übergeneralisierungen zu vermeiden, kann eine solche Strategie allerdings nur vereinzelt eingesetzt werden.
- **Unbekannte Wörter:** Falls das erkannte Wort nicht im linguistischen Lexikon vorkommt, wird es von der Sprachverstehenskomponente übergangen. Solange das unbekannte Wort für die semantische Interpretation unerheblich ist, wie bei Höflichkeitsfloskeln, ist dies Verfahren erfolgreich. Langfristig sollte die Sprachverstehenskomponente jedoch unbekannte Wörter lernen können, um inhaltlich relevante Information nicht zu verlieren. Auch wenn das System manchmal auf die syntaktische Funktion eines Wortes aufgrund seines Umfelds schließen kann (z.B. *und die Leiste steckst du auf die rote* <Unbekannt>), wird ein Klärungsdialog mit dem Instrukteur im Regelfall notwendig sein.
- **Wortauslassungen:** Ein Klärungsdialog mit dem Instrukteur sollte außerdem geführt werden, wenn bei einer elliptischen Äußerung oder einer Wortauslassung semantische Information vom System nicht inferiert werden kann. Selbst wenn eine Äußerung nicht vollständig interpretiert werden kann, generiert das Sprachverstehenssystem zumindest partielle Interpretationen, so daß Nachfragen möglich sind, wie z.B. *Was soll ich mit 'die Leiste' und 'auf die rote Schraube' tun? bzw. die rote Was?*.
- **Semantische Inkonsistenz:** Eine Anweisung kann semantisch inkonsistent sein oder zumindest dem System so erscheinen. Im Zweifelsfall muß das System beim Instrukteur um eine Korrektur oder eine detaillierte Spezifikation nachfragen. Manchmal kann es eine vollständige Handlungsanweisung ableiten, indem weitere Informationen aggregiert und fehlende Informationen inferiert werden. Wörtlich genommen kann beispielsweise die Anweisung *schraub die Leiste auf den Würfel* nicht ausgeführt werden, da weder die Leiste noch der Würfel die Funktion einer Schraube ausübt. Wird allerdings das benötigte Instrument vom System inferiert (*schraub die Leiste mit der roten Schraube auf den Würfel*), ist die Handlung möglich. Bei einer unterspezifizierten Anweisung wird also aufgrund semantischen Wissens das benötigte Instrument inferiert.
- **Sprachliche Ambiguität:** Das Sprachverstehenssystem muß darüber hinaus verschiedene Formen der Ambiguität bewältigen. Auf der lexikalischen Verarbeitungsebene kann ein Wort unterschiedliche syntaktische Kategorien haben, z.B. kann *schrauben* sowohl Nomen wie Verb sein (die entsprechende Orthographie liefert der Spracherkenner nicht, da dazu eine syntaktische Voranalyse nötig wäre). In den meisten Fällen wird eine solche lexikalische Ambiguität durch den syntaktischen Kontext aufgelöst; d.h. es wird jene Kategorie gewählt, mit der eine syntaktische Analyse möglich ist. Außerdem kann auf der syntaktischen Verarbeitungsebene die Anbindung der Konstituenten ambig sein. Ein klassisches Beispiel hierfür ist die Anbindung der Präpositionalphrase [HR97]. Hierbei nutzt das Sprachverstehenssystem eine Vielzahl von Informationsquellen. Unter anderem wird geprüft, ob für das polyseme Verb eine präferierte Lesart vorliegt, ob das Objekt semantisch mit der Funktionalität des Verbarguments übereinstimmt und für welche Interpretation es ein Denotat im situativen Kontext gibt. Kann das Sprachverstehenssystem die Ambiguität nicht selbst auflösen, sollte es beim Instrukteur nachfragen.
- **Situative Ambiguität:** Eine Anweisung, die von der Sprachverstehenskomponente semantisch vollständig interpretiert wurde, führt nicht unbedingt zu einer gleichfalls erfolgreichen Handlung. Beispielsweise können bei einer Anweisung wie *nimm eine Schraube* mehrere Schrauben in der Szene als

Denotat zur Verfügung stehen. Im Regelfall folgt die Handlungskomponente dann einem Ökonomieprinzip: Es wird jenes Objekt gegriffen, das am leichtesten zu erreichen ist und dessen Positionierung am wenigsten Greifprobleme bereitet. Je nach Verteilung der Objekte in der Szene kann das System nachfragen, ob beispielsweise eine lange oder eine kurze Schraube benötigt wird.

- **Sprachliche Intervention:** Aus einer Vielzahl von Ursachen kann es dazu kommen, daß die Handlungskomponente eine Handlung ausführen will, die bezüglich der Intention des Instrukteurs falsch ist. Dies ist beispielsweise immer dann der Fall, wenn das System eigenständig entscheidet und ein Objekt greifen will, das der Instrukteur nicht gemeint hat. Während der Greifarm sich auf das Objekt zubewegt, hat der Instrukteur allerdings die Möglichkeit, verbal zu intervenieren. Gegenwärtig kann die Handlung nur durch den Befehl *stop* unterbrochen werden. Ergänzend sollen künftig auch korrigierende Information wie *eine andere Schraube* bzw. *besser die vordere Schraube* einfließen können. Die Handlung wird hierdurch lediglich modifiziert, womit vermieden werden soll, daß sie völlig neu aufsetzen muß.

Die Anforderung an das Gesamtsystem, echtzeitfähig zu sein, hat für die Sprachkomponente zur Folge, daß sie nicht beliebig lange Zeit für die Interpretation benutzen darf. Sie muß vielmehr jederzeit in der Lage sein, zumindest Teilinterpretation an andere Komponenten liefern zu können (*any-time capability*). Hierzu benötigt sie Kriterien, um eine laufende Analyse und Interpretation abbrechen zu können. Ein quantitatives Abbruchkriterium ist die bisher verstrichene Zeit und die Anzahl konkurrierender Lösungswege. Ein qualitatives Abbruchkriterium ist erreicht, wenn die zur Handlung benötigte Information vorliegt, also beispielsweise der Typ der Handlung und alle obligatorischen Argumente. Zugrundegelegt wird hierbei eine inkrementelle und interaktive Verarbeitung sowie das Prinzip, nicht allen möglichen Interpretationen zu folgen, sondern nur der stabilsten und informativsten (*intelligent pruning*). Die Operationalisierung dieses Prinzips ist allerdings nicht unproblematisch und hängt oft von spezifischen Konstellationen ab.

### 0.2.2.2 Sensorische Basis und Fähigkeiten

Die *sensorischen* Fertigkeiten des Systems dienen als Grundlage der Umwelterkennung und der sensomotorischen Manipulationsfähigkeiten. Als physikalische Sensorbasis kommen zum Einsatz: Kraft-Momenten-Sensoren, mitbewegte Miniatur-Farbkameras an den Robotergriffen sowie externe statische und bewegliche Kameras für unterschiedliche Erkennungs- und visuelle Regelungsaufgaben. Der direkte Zugriff auf die Sensoren erfolgt über *Sensoragenten*, realisiert als *MagiC-Agenten* (siehe unten). Mit den von diesen Agenten gelieferten Rohdaten bzw. vorverarbeiteten Daten realisieren andere Agenten (ohne physikalischen Sensor) die eigentliche Verarbeitung.

Wir stellen im folgenden eine Auswahl von implementierten Diensten vor, die jeweils in Form autonom arbeitender Agenten mit einheitlicher Schnittstelle von einer zentralen Instanz direkt angesprochen werden können. Alternativ können diese Agenten sich über das *Vertragsnetzprotokoll* ihrer Dienste auch gegenseitig bedienen (siehe unten).

- **Kamera-Agenten:** Aufgabe der Kamera-Agenten ist es, die ihnen zugeordnete physikalische Ressource Kamera auf Anfrage transparent jedem anderen Agenten im Netzwerk zur Verfügung zu stellen. Transparent bedeutet in diesem Zusammenhang den scheinbar exklusiven Zugriff auf Bilder der angeschlossenen Kamera. Der Kamera-Agent ist dabei als *concurrent server* [Ste93b] ausgelegt, so daß prinzipiell beliebig viele Erkennungsaufgaben über das Netzwerk nebenläufig bearbeitet werden können. Neben der Fähigkeit, Bilder der Typen *Grauwert-Bild*, *RGB-Farbbild*, *YUV-Farbbild* und *HSV-Farbbild* zu liefern, haben Kamera-Agenten noch die zusätzliche Funktion, aus Rohdaten pixel-klassifizierte Bilder zu erzeugen, in denen Pixel eines Farbbildes einer bestimmten Farbklasse zugeordnet werden. Diese Bilder werden nach den in [SK97b] und [Sch97] beschriebenen Verfahren erzeugt und dienen als Grundlage eines lokalen Objekterkennungs-Agenten. Darüber hinaus bieten die Kamera-Agenten die Möglichkeit, Grauwert-Bilder auf niedrigdimensionale Subräume zu projizieren und damit extrem zu komprimieren [Oja83]. So wird erreicht, daß statt des kompletten Bildes lediglich eine niedrigdimensionale Projektion über das Netzwerk versandt wird.

- **Kraft-Momenten-Agenten:** Der Zugriff auf die Kraft-Momenten-Sensoren (KMS) erfordert das Einhalten harter Echtzeit-Bedingungen (garantierte Verweildauer in einem Regelzyklus  $< 10\text{ms}$ ), da sonst eine Echtzeitregelung der kraftempfindlichen sensomotorischen Fertigkeiten (Schrauben, Stecken und Ablegen) nicht möglich wäre. Dies wird dadurch gewährleistet, daß zum einen durch einen Reservierungsmechanismus lediglich ein Prozeß den KMS benutzen kann und zum anderen der Zugriff auf die KMS-Agenten über das Netzwerk verboten ist.

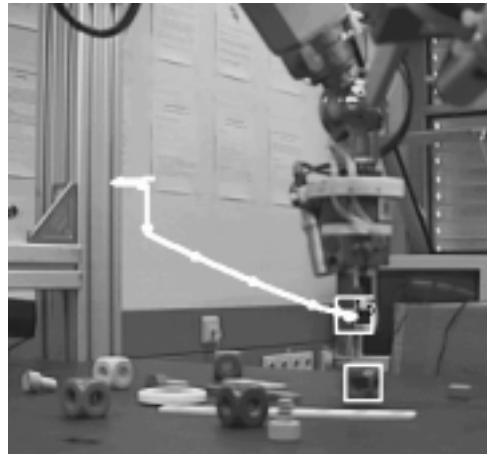
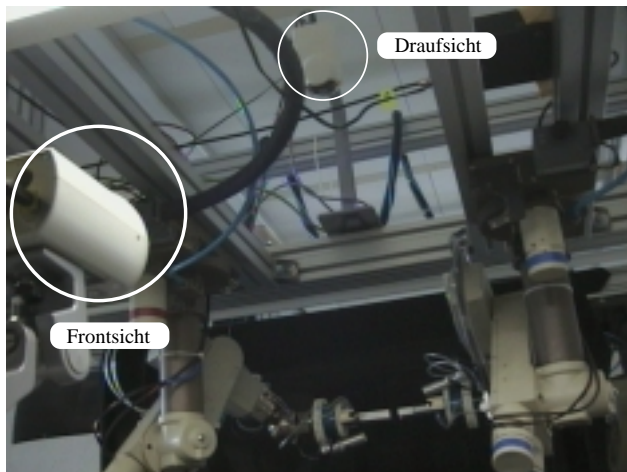
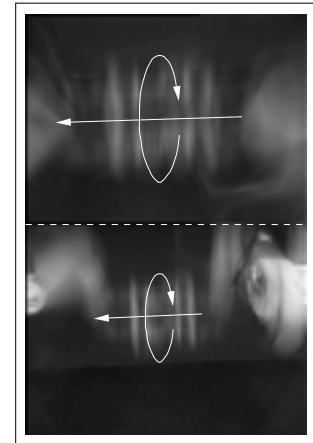


Abbildung 10: Beispiel einer verfolgten Roboter-Trajektorie. Das Rechteck um die Greiferspitze beschreibt die aktuelle Suchregion.

- **Agent zur Verfolgung von Objekten in Bildsequenzen:** Basis verschiedener möglicher Dienste, etwa der implementierten Mehrkamera-Verfolgung eines Roboters (siehe unten) ist die Fähigkeit, den Roboter in einer Bildsequenz zu erkennen (Abbildung 10). Diese Agenten verwenden ein regionenbasiertes *Template-Matching* unter Berechnung der sog. *sum of squared differences* (SSD)[Hag95]. Um die Laufzeit-Effizienz des Verfahrens zu erhöhen, wird nach dem Roboter in einem Folgebild statt im ganzen Bild lediglich in einer Suchregion um die zuletzt ermittelte Bildposition herum gesucht. Diese Suchregion verringert sich mit jedem Folgebild. Da nicht sicher ist, daß der Roboter immer korrekt verfolgt wird, beurteilt der Agent die Güte seines eigenen Meßresultats mit den in [SK98d] beschriebenen Maßen. Mit Hilfe dieser Maße können solche Agenten, die die Meßergebnisse nutzen wollen, entscheiden, ob die angebotene Dienstleistung mit der notwendigen Genauigkeit angeboten wird.
- **Agent zur Zwei-Arm Distanz-Schätzung:** Die Arme werden von zwei externen *unkalibrierten* Kameras beobachtet (Abbildung 11(a)), deren Bilder zu einem zusammengefügt werden (Abbildung 11(b)).  
Zur Distanz-Schätzung wird ein speziell trainiertes Neuro-Fuzzy-System verwendet. Dessen Eingabe während des Trainings und zur Laufzeit bilden zuvor konkatenierte Kamera-Bilder. Die Distanz wird dann durch *Fusion aller Teilresultate* ermittelt [SZK99a]. So wird erreicht, daß nicht nur die Daten verteilt akquiriert werden, sondern auch, daß die Verarbeitung parallelisiert werden kann. Zur Zeit ist die Arm-Distanz-Schätzung bei verschiedenen einfachen Bewegungsmustern möglich [SZK99b].
- **Agent zur Winkelbestimmung einer Leiste:** Die bislang auf Leisten definierte Operation *Ausrichten* verlangt die präzise Feststellung der Richtung der Hauptachse einer Leiste gegenüber einer Bezugsebene. Die Erkennung der Leistenwinkel basiert auf einem Mustervergleich, bei dem nach markanten Strukturen gesucht wird, in diesem Fall den Leistenlöchern (Abbildung 12). Da der Roboter beim Ausrichtungsvorgang eine vorgegebene Position einnimmt, kann der Suchraum stark eingeschränkt werden, wobei allerdings auch störenden Einflüssen Rechnung getragen werden muß.



(a)



(b)

Abbildung 11: (a) Zwei Kameras beobachten die Roboter. Der Abstand der Kameras sowohl zueinander als auch zu den Armen beträgt etwa 1m. (b) Konkateniertes Mittelwertsbild bei gegenseitiger Umkreisung der Arme. Die Pfeile deuten die Bewegungsrichtung an.



Abbildung 12: Kamerasicht bei der Bestimmung des Leistenwinkels inklusive Suchraster und ermitteltem Winkel.

### 0.2.2.3 Motorische und Sensomotorische Fähigkeiten

Die einfachsten verfügbaren Montagefertigkeiten beziehen sich nur auf die Aktorik. Sie existieren für jeden der verwendeten Roboter und dienen als Basis für alle weiteren komplexen Fertigkeiten. Sie werden hier nur der Vollständigkeit halber erwähnt, Voraussetzung ist jedoch in jedem Fall eine Robotersteuerung zur schnellen und feinfühligten Kraftregelung (wie sie bislang industriell kaum erhältlich ist).

- **Basis- und Greifertransformationen festlegen:** Dies sind Transformationen, die die Lage des Roboters zu einem globalen Bezugssystem bzw. die Lage der Manipulatorspitze in bezug auf den Roboter beschreiben. Dies wird benötigt, um dynamisch auf unterschiedliche Roboterstandpunkte und unterschiedliche Bauteile im Greifer der Roboter eingehen zu können.
- **Hand öffnen und schließen:** Wird benötigt, um Kontakt zu Bauteilen herzustellen bzw. ihn wieder aufzugeben.
- **Relativ- und Absolutbewegungen:** Es sind eine Reihe von unterschiedlichen Bewegungen für die Roboter verfügbar. Sie unterscheiden sich hauptsächlich in:

- dem Bezugssystem, in dem sie agieren. Bewegungen sind in bezug auf das globale Weltkoordinatensystem, auf das Roboterkoordinatensystem oder in bezug auf das Manipulatorkoordinatensystem beschrieben.
  - dem Interpolationmodus. Dieser Modus gibt an, ob der Roboter im kartesischen Koordinatensystem (Weltkoordinatensystem) oder im Gelenkwinkelraum eine geradlinige Bewegung durchführen soll. Die Möglichkeit, zwischen diesen Bewegungsarten umzuschalten ist unabdingbar, um in der Umgebung *singulärer Konfigurationen* des Roboters mit diesem interagieren zu können.
- **Festlegen der Manipulatorgeschwindigkeit.** Je nach dem Ort, an welchem sich der Greifer befindet, kann die Geschwindigkeit vorgewählt werden, mit der er sich bewegt.
  - **Unmittelbares Anhalten des Manipulators:** Weil der Instrukteur auch in bereits laufende Aktionen der Roboter eingreifen können soll, ist es notwendig, die Roboter zu jeder Zeit (in jedem beliebigen Zyklus der Steuerung) sofort stoppen zu können. Hierzu ist eine weitere Schnittstelle zur Robotersteuerung geschaffen worden, um am normalen Kontrollfluß vorbei in die Bewegungsplanung eingreifen zu können.

Die *sensomotorischen Fertigkeiten* erlauben es dem Roboter, sich über Bauteilen in beliebiger Lage optimal zu positionieren, Kontakt herzustellen, sie zusammenzustecken, sie zu schrauben und sie kraftgeregelt abzulegen. Komplexere Fertigkeiten können sich aus diesen zusammensetzen. Um z.B. ein Bauteil zu greifen, wird eine mehrstufige Strategie verfolgt: Grobpositionierung über dem Objekt, visuell geregelte Feinpositionierung, Kontaktherstellung (kraftüberwachter Griff) und ggf. Umgreifen. Im folgenden wird eine Auswahl der implementierten sensomotorischen Fertigkeiten beschrieben. Dabei soll auch deutlich werden, welche erheblicher Anzahl von Einzelaspekten Rechnung zu tragen ist, um auch scheinbar einfache Montagefolgen sicher und flexibel steuern zu können.

1. **Gefügte Bewegungen:** Sobald an Bauteilen mit geringer Maßhaltigkeit manipuliert wird, kann nicht mehr mit fest vorgegebenen Bewegungen gearbeitet werden. Es sind gefügte Bewegung notwendig, die auf außen anliegende Kräfte reagieren (z.B. Bewegungsabbruch bei definierter Kraftschwelle und/oder Aufrechterhaltung einer bestimmten Kraft). Die Kraft-Momentenregelung wird mit einem Neuro-Fuzzy-System realisiert, das mit einem on-line Lernverfahren schnell auf die jeweiligen Gegebenheiten und unterschiedlichen Roboterkonfigurationen angepaßt wird [ZvCK97].
2. **Lochsuche:** Zur Lochsuche vor dem Durchstecken eines länglichen Bauteils (Schraube) durch eine Leiste existieren zwei Methoden:
  - Ausführung einer spiralförmigen Bewegung auf der Oberfläche, in der ein Loch vorhanden sein soll. Dabei wird in Annäherungsrichtung eine definierte Kraft aufrechterhalten. Werden durch das Einschneiden in das Loch entlang der Normalen- und Schließrichtung definierte Schwellkräfte überschritten, ist damit das Loch detektiert [vCZK97].
  - Beim zweiten Verfahren werden die Bauteile mit den Handkameras der Roboter beobachtet und die Schraube so zielgerichtet über das Loch geführt [vCZK98]. Durch gleichzeitige Beobachtung der auftretenden Kräfte erlangt man eine hohe Robustheit (Abbildung 13).

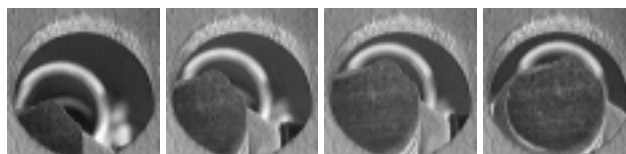


Abbildung 13: Korrekturschritte unter Verwendung der Handkamera.



3. **Visuelle Grobpositionierung:** Da es aufgrund diverser Ungenauigkeiten unmöglich ist, nach geometrischer Planung Objekte „blind“ zu greifen, wird der Roboter zuerst lediglich in die Nähe der interessierenden Objektlage bewegt. Dann sorgt visuell geregelte Feinpositionierung unter Verwendung der Handkamera für eine optimale Greifpostur. In Ergänzung hierzu wurde ein selbst-kalibrierendes Positionierungsverfahren entwickelt, welches *keine* a-priori 3D-Rekonstruktion der Szene benötigt [SK98a]. Es ist in der Lage, über die Fusion mehrerer ( $n \geq 2$ ) unkalibrierter 2D-Ansichten der Szene ebenfalls eine Positionierung über einem Objekt durchzuführen [SK98b]. Dazu benötigt es pro Sicht die 2D-Differenz zwischen aktueller Roboterposition (geliefert von Agenten zur Verfolgung von Objekten aus 0.2.2.2) und dem Zielobjekt. Zur Fusion der Sichten wird ein Kalman-Filter eingesetzt. Abbildung 14 zeigt, daß durch Redundanzausnutzung selbst dann noch ein robustes Positionieren möglich ist, wenn einzelne Track-Agenten den Roboter aus ihrer lokalen Sicht verlieren [SK98d].

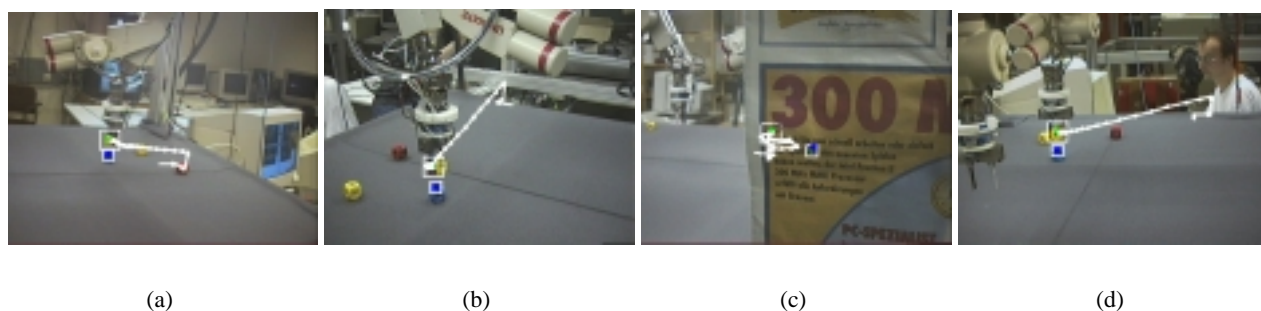


Abbildung 14: Trotz Sichtversperrung in Kamera (c) ist aufgrund der (redundanten) weiteren Sichten eine Positionierung möglich.

4. **Visuelle Feinpositionierung:** Um ein Objekt mit einer beliebigen Orientierung und Lage auf dem Tisch zu greifen, werden die Handkameras benutzt. Die visuell geführte Feinpositionierung wurde basierend auf dem in [ZKS99] vorgestellten Modell implementiert. Es wird pro Freiheitsgrad ein eigener Regler trainiert, deren Resultate zu einer Korrekturbewegung *fusioniert* werden.



Abbildung 15: Übersicht greifbarer Objekte in ihren unterschiedlichen Lagen.

5. **Greifen mit Umgreifen:** Die vorstehenden Fertigkeiten werden alle benötigt, um das Greifen durchführen zu können. Abbildung 15 zeigt alle greifbaren Objekte. Ist der Roboter optimal positioniert, wird das kraftgesteuerte Greifen ausgelöst. Dabei nimmt der Roboter die nach Art und Lage des Objekts notwendige Greifhöhe ein und schließt den Greifer. Bei liegenden Schrauben ist ein Umgreifen mit dem anderen Arm notwendig (Abbildung 16), um die Schraube in ihre „kanonische Lage“ zu bringen.

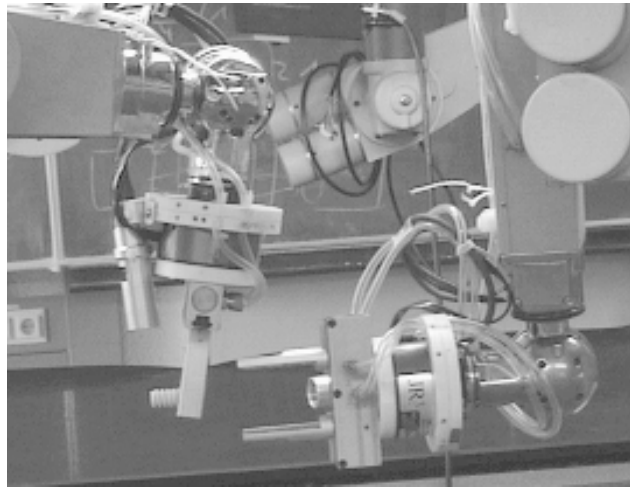


Abbildung 16: Umgreifen einer liegend gegriffenen Schraube.

6. **Stecken:** Leisten, Hülsen und Räder müssen auf eine Schraube aufgesteckt werden, bevor sie festgeschraubt werden können. Da vor Schraubbeginn nicht davon ausgegangen werden kann, daß die Schraube sich über dem Loch des jeweiligen Bauteils befindet, muß das Loch mit den oben beschriebenen Verfahren gesucht werden. Befindet sich die Schraube über dem Loch, kann das entsprechende Bauteil aufgesteckt werden, wobei eine gleichzeitige Drehbewegung der Schraube ein Verkanten verhindert. Durch anschließende Prüfbewegungen wird festgestellt, ob die gesamte Operation erfolgreich war oder ob ein Fehler aufgetreten ist. Im Fehlerfall wird der Instrukteur benachrichtigt, der anschließend entscheiden kann, ob er die gesamte Operation wiederholen oder manuell berichtigen möchte.
7. **Schrauben:** Die wichtigste und mit Abstand komplexeste Operation zum Verbinden der Bauteile ist das Schrauben mit beiden Armen. Nach unserer Kenntnis wurde sie bislang mit kooperierenden Robotern überhaupt noch nicht realisiert. Sie besteht aus folgende Teilschritten: Auffahren der Schraube auf die Mutter oder die Raute; Lochsuche mit Verifikation; Finden des Einschnittpunktes im Gewinde; Hineindreihen der Schraube ins Gewinde; Prüfen, ob in das Gewinde eingeschnitten wurde; Festziehen der Schraube bis zum Erreichen eines bestimmten Schraubmoments.

Nachdem der Kontakt zwischen der Schraube und dem Würfel hergestellt ist, kann noch nicht davon ausgegangen werden, daß die Schraube sich im Gewindeloch des Würfels befindet. Das Loch muß, wie beim Aufstecken von Bauteilen, gesucht werden. Für die Suche kann nur die spiralförmige Bewegung der Schraube auf der Oberfläche der Mutter oder der Raute verwendet werden, da die Sicht für die gegenüberliegende Handkamera versperrt ist. Ist durch Drehen der Schraube entgegen der Einschraubrichtung der Einschnittpunkt gefunden worden, so wird die Schraube in das Gewinde gedreht. Sowohl beim Suchen des Einschnittpunktes als auch beim eigentlichen Schrauben müssen die anliegenden Kräfte geregelt werden. Ist dies nicht der Fall, verschiebt sich unter Umständen die Schraube im Greifer oder schneidet nicht ins Gewinde ein. Wurden die Bauteile mit einem starken Versatz entlang des Normalenvektors gegriffen, so daß die Drehachse der Schraube deutlich von der des Greifers abweicht, ist ebenfalls mit einem Fehler zu rechnen. Daher müssen diese Abweichungen während des Schraubens durch geeignete Ausgleichsbewegungen kompensiert werden. Eine Regelung der Querkräfte ist notwendig. Um das Schrauben unter diversen Unsicherheiten robust zu machen, wird eine fuzzy-basierte on-line Lernmethode ([ZvCK97], [ZF98]) verwendet. Die fuzzy-linguistischen Regeln für die Regelung in jede Richtung können nach dem Lernen extrahiert werden [FZK99].

Zusätzliche Robustheit wird erreicht, wenn der Winkel betrachtet wird, mit dem die Schraube im Greifer sitzt und den Manipulator entsprechend dreht, so daß die Drehachse der Schraube mit der Symmetrieachse

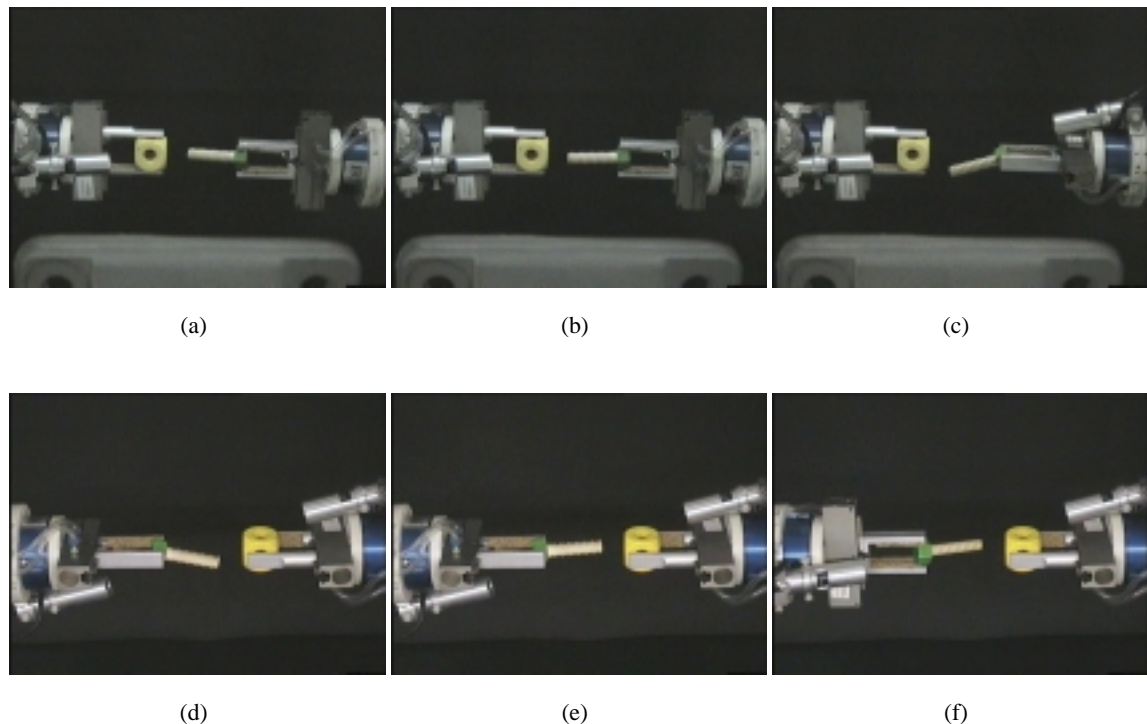


Abbildung 17: Typische Ansichten der beiden externen Kameras ((a)-(c) Sicht von oben, (d)-(f) Seitenansicht)

des Loches zusammenfällt [vCSZK99]. Diese Achse und nicht der Annäherungsvektor des Manipulators wird als Drehachse verwendet. Die Schraube wird aus zwei unterschiedlichen Perspektiven betrachtet (Abbildung 17) und separate Ausschnittsbilder zu einem fusioniert (Abbildung 18).

8. **Visuell gesteuertes Leistenausrichten:** Für eine ganze Reihe von Aggregaten werden exakt ausgerichtete Leisten verlangt (z.B. beim Bau des Leitwerks oder des Rumpfes). Ist die Position der Leiste bekannt, so kann sie durch den Roboter ausgerichtet werden, indem dieser rechts (oder links) an dem zusammengebauten Aggregat vorbeifährt und die Leiste dadurch in die richtige Position schiebt (Abbildung 19(a)). Anschließend wird die Leiste durch den haltenden Roboter fixiert und die Schraube nachträglich festgezogen. Diese Operation setzt sich aus den folgenden Teiloperationen zusammen: Anfahren beider Roboter in eine definierte Position, Bestimmung des Leistenwinkels, Ausrichtbewegungen in Abhängigkeit vom bestimmten Winkel.
9. **Aggregatbau:** Es können unter Verwendung der beschriebenen Fertigkeiten eine ganze Reihe von Aggregaten weitgehend autonom zusammengebaut werden; (Abbildung 20). Ein komplexes Beispiel dafür ist der Bau des Flugzeugrumpfes. Dazu müssen beispielsweise zwei Leisten parallel miteinander verschraubt werden (Abbildung 19(b)) – alles durch Instruktion im Dialog und bei weitgehender Freiheit der Lage der Objekte (Ausgangssituation).

#### 0.2.2.4 Steuerung durch ein Multi-Agenten-System

Um die Zuverlässigkeit sensorbasierter Systeme zu erhöhen, wurde schon vor längerem der Einsatz von Multisensor-Systemen vorgeschlagen [DW88]. Dabei werden Mechanismen zur Sensorselektion [GJ94], zur Kommunikation zwischen Sensoreinheiten [ISK92], zur Fusion der Information verschiedener Sensoren [EHA93] und der Sensor-System-Modellierung [DH98] benötigt. Von der vorhandenen Redundanz erhofft man sich eine größere Robustheit, Schnelligkeit und Flexibilität des Gesamtsystems im Vergleich zum Einzelsensor. So soll z.B. der Ausfall eines Sensors in einem redundanten Netzwerk lediglich zu *graceful degradation* der

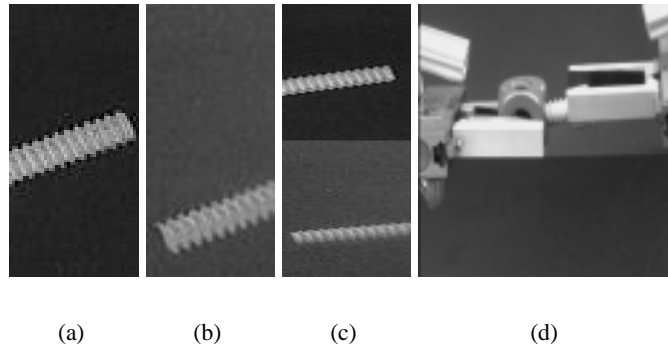


Abbildung 18: Schrauben unter Fusion von Kraft und zweier externer Kameransichten. Ausschnittbild von Kamera 1 (a) und Kamera 2 (b) und das zusammengefügte Gesamtbild (c). Trotz Abweichung der Schraube von ihrer kanonischen Greiflage ist Schrauben möglich (d).

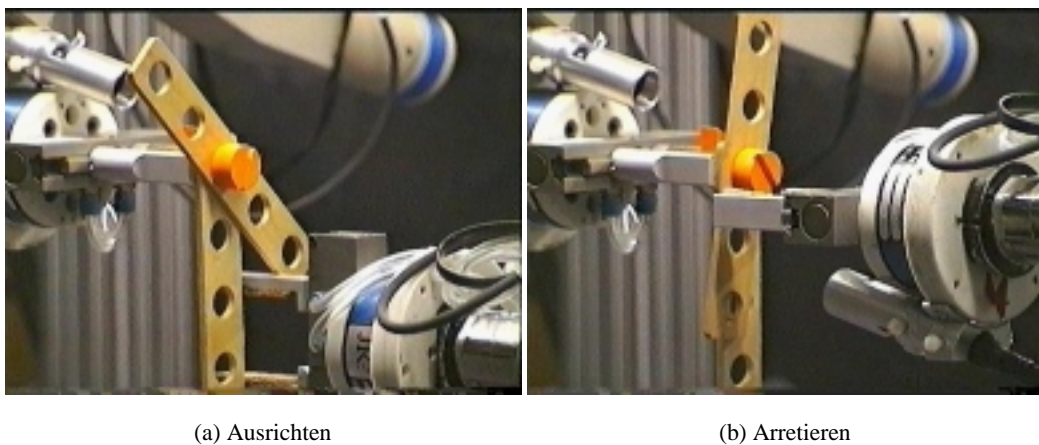


Abbildung 19: Ausrichten einer Leiste kollinear zu einer schon gegriffenen.

Leistungsfähigkeit statt zu einem Totalausfall führen. Durch Vergabe von Subaufgaben auf mehrere gleichartige Komponenten soll der mittlere Durchsatz gesteigert werden und neu hinzugekommene Komponenten sollen nahtlos integriert werden können oder alte ersetzen.

Das Hauptproblem liegt dabei in der Organisation und Koordination des Sensoreinsatzes. **MagiC** (**M**ulti **a**gent **g**eneration **i**n **C++**) wurde basierend auf der früheren Arbeit [MK92] entwickelt [SK98c]. Damit ist die Entwicklung kooperierender Agenten möglich, wobei deren Einsatzgebiet nicht notwendigerweise auf die Modellierung von Sensoragenten beschränkt ist. Es wird damit relativ einfach möglich, bereits vorhandene Funktionalität anderer C/C++-Bibliotheken leicht in das Agenten-Framework durch „wrapper“ zu integrieren [Bur98]. Die wichtigsten Merkmale von Agenten in MagiC sind:

- Agenten können sich reaktiv, proaktiv oder gemischt verhalten.
- Alle Agenten bilden einen *virtuellen Agentenraum*, d.h. es findet eine vollständige Abstraktion von der tatsächlichen physikalischen Abbildung auf Threads, Prozesse und Rechner statt.
- Kooperierende Agenten können sich durch aufgabenorientierte Kommunikation dynamisch konfigurieren. Dadurch wird die selbstgesteuerte Ausnutzung funktionaler Redundanz möglich. Außerdem können Agenten zeitlich persistente Teams bilden, die sich bei Bedarf dynamisch rekonfigurieren.
- Verschiedene Aufgaben können parallel bearbeitet werden.

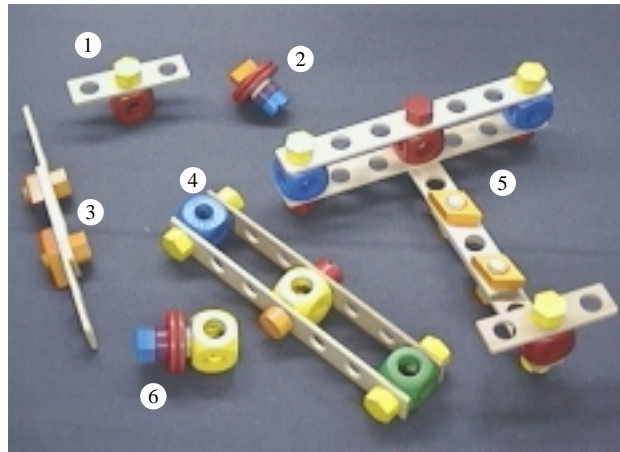


Abbildung 20: Bau- bzw. handhabbare Aggregate. (1) Leitwerk, (2) Teil des Fahrgestells, (3) Rumpf, (4) Doppelflügel (handhabbar zum Bau von (5)), (5) Flugzeug montiert aus (1), (3) und (4), (6) Teil des Fahrgestells mit Würfel.

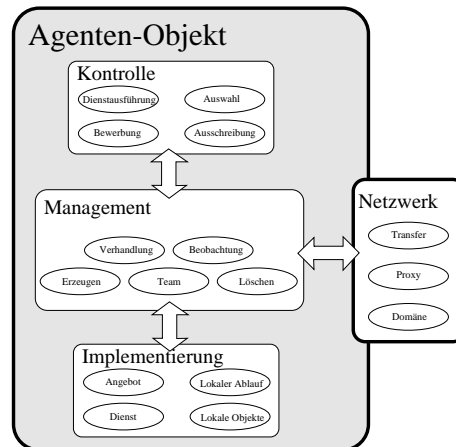


Abbildung 21: Agenten-Modell in MagiC.

Agenten in MagiC lassen sich also als *Agenten-Objekte* der Struktur nach Abbildung 21 auffassen. Ein Agenten-Designer leitet sie von einer bestimmten Basisklasse ab, um die gewünschte Funktionalität in ihnen zu implementieren, wobei verschiedene Basisklassen für unterschiedliche Komplexität zur Verfügung stehen. Agenten bieten daraufhin Dienste an, die von anderen Agenten im Netzwerk genutzt werden können. Diese Dienste können selbst wieder aus Subdiensten anderer Agenten zusammengefügt sein. Die Kooperation der Agenten ergibt sich anhand von Aufgabenbeschreibungen und Verhandlungen nach einem erweiterten Vertragsnetzprotokoll [Smi81]. Dabei schreibt ein *manager* Aufgabenbeschreibungen (*tasks*) aus, mit denen er den gewünschten *service* beschreibt. Aus den Bewerbern wird (je nach Bedarf) eine Untermenge beauftragt, die Dienstleistung zu erbringen. Alternativ kann ein Agent auch direkt angesprochen werden, um seine Dienstleistung in Anspruch zu nehmen.

Aus Sicht des Anwenders realisiert ein Agent zum einen den proaktiven *lokalen Ablauf*, der selbständig Ziele verfolgt und Aufgaben ausschreibt. Zum anderen implementiert er den reaktiven *Dienst*-Teil, der mittels einer Aufgabenausschreibung von außen angefordert werden kann. Jeder Dienst enthält eine Signatur, die mit der der ausgeschriebenen Aufgabe übereinstimmen muß, um ein *Angebot* für ihre Bearbeitung abzugeben. So ergibt sich eine aufgabenabhängige *Manager – Contractor*-Konfiguration. Der große Vorteil für den Anwender (insbesondere bei einer redundanten Erfüllungsmöglichkeit seiner Aufgabe) ist, daß das System *dynamisch*

seine Vertragspartner findet, ohne daß sie a-priori bekannt sein müssen. Das bedeutet, daß eine automatische Aufgaben- und Lastverteilung nach dem aktuellen Verfügbarkeitszustand des Gesamtsystems stattfindet, die durch verschiedene *Kontrollstrategien* in ihrem Verhalten beeinflusst werden kann. Der gesamte Ablauf der Aufgabenvermittlung samt Parameterübergabe, der Contractorauswahl, der Neuausschreibung und der Ergebnisübermittlung geschieht für den Agentennutzer unsichtbar im *Management-Teil* des Agenten. In *MagiC* gibt es verschiedene Kontrollstrategien, mit denen man das Verhalten von Agenten während einer Aufgabenbearbeitung definieren kann. Mit einer von fünf möglichen *Auswahlstrategien* (*early bird*, *all*, *best subset*, *requested quality* und *selection from all*) legt der Manager dynamisch fest, wie er aus den Bewerbern seine *Contractors* auswählt. Auf Wunsch kann die *Ausschreibung* einer Aufgabe *synchron* (der Manager wartet, bis die Aufgabe entweder erfüllt oder durch Zeitunterschreitung beendet wurde) oder *asynchron* (der Manager wartet erst an einem Rendezvous-Punkt auf das Resultat) erfolgen. Damit ein Agent eine *Bewerbung* auf eine Aufgabe abgeben kann, vergleicht sein Managementteil autonom seine Ein/Ausgabe-Signatur mit der ausgeschriebenen Aufgabe, um zu entscheiden, ob er prinzipiell ein Angebot abliefern kann. Man kann die Menge potentieller *contractors* dadurch vergrößern, daß bei der Aufgaben-Ausschreibung generische Signaturelemente verwendet werden. Die *Dienstausführung* in einem *contractor* kann nach dem *fifo*-Prinzip oder nach der *immediate service*-Strategie erfolgen, bei der für jeden neuen Auftrag dynamisch eine neue Instanz des Agenten erzeugt wird. Neben der internen Abwicklung der *Verhandlung* zur dynamischen Manager-Contractor-Zuordnung und dem *Erzeugen* und *Löschen* von Agenten, ermöglicht das Agenten-Management auf Wunsch die *Beobachtung* der verschiedenen Phasen einer Auftragsbearbeitung durch Propagierung der internen Zustände an einen speziellen Visualisierungs-Agenten. Zusätzlich ist dieser Teil auch für die Verwaltung von *Teams* zuständig, mit denen Agenten über einen längeren Zeitraum einem Auftraggeber exklusiv zur Verfügung stehen.

In *MagiC* ist durch Umsetzung des *Proxy*-Konzepts [Sha86] ein sog. virtueller Agenten-Raum realisiert. Das bedeutet, daß alle Agenten gleich erreichbar scheinen, unabhängig davon, wie sie real instantiiert sind (als Thread, als lokaler Prozeß oder als Prozeß auf einem anderen Rechner). Weitere Aufgaben der Netzwerk-Komponente sind die *Domänen*-Verwaltung, um disjunkte Bereiche *nicht interagierender Agenturen* zu schaffen und der *Transfer* von C++-Objekten.

### 0.2.3 Ausblick auf zukünftige Forschungslinien

Bei der Montage größerer Einheiten (Aggregate) ist es weder wünschenswert noch möglich, daß ein menschlicher Instrukteur sämtliche sensorischen und motorischen Einzeloperationen eines Robotersystems durch natürlichsprachliche Anweisungen einzeln steuert. Statt einer objektreferenzierenden natürlichsprachlichen Anweisungsfolge wie „Nimm die Dreilochleiste“, „Greif die blaue Schraube mit dem anderen Arm“, „Greif jetzt um“, „Steck sie jetzt durch das mittlere Loch“, „Zieh sie fest an“ . . . sollte also einfach eine aggregatreferenzierende Instruktion der Form „Bau mir ein *Leitwerk* mit blauer Schraube und gelbem Würfel“ gegeben werden können. Die Ausführung dieser komplexen Anweisung sollte ohne Rückgriff auf bereits bestehende oder im voraus zu entwerfende Modelle der Aggregate möglich sein. Dazu muß das (Teil-)Aggregat im Dialog von Mensch und Konstrukteur zu einem früheren Zeitpunkt gebaut und durch den Menschen benannt worden sein. Eine veränderte Ausgangssituation (Lage der Teile, sensorische Ausstattung, zur Verfügung stehende Aktoren) soll zu keiner Einschränkung der Autonomie führen. Sehr wünschenswert ist ferner die Fähigkeit, Bedeutungen neuer Handlungsverben und Objekte zu lernen. Der Roboter soll also durch Erfahrung bei der interaktiven Montage sukzessive lernen und auf immer abstraktere Anweisungen situativ reagieren können. Aus unserer Sicht ergeben sich damit vier wesentliche Bereiche für weitere Untersuchungen:

- *Autonomie bei Abfolgeplanung und -ausführung*: Bereitstellung und Integration von Planungsverfahren zur Montage unter dem Gesichtspunkt der Echtzeitfähigkeit; Planung und Ausführung von Demontageoperationen,
- *Autonomie bei sensorischer Erfassung*: Verschränkte Planung von Sensor- und Aktoreinsatz unter Unsicherheit; Integration von Datenfusionsverfahren in diesen Prozeß,



- *Autonomie bei Wissenserwerb und -anwendung*: Interaktives Lernen neuer Montagesequenzen und dabei Anpassung an veränderte Ausgangssituationen,
- *Autonomie bei der Dialogführung*: Intervenierbarkeit; Abstraktion und Transformation von Fertigkeiten; interaktives Erlernen von Begriffen und Stellen von Rückfragen.

Wir beschränken uns bei deren Erläuterungen angesichts des beschränkten Platzes in diesem Band auf einige kurze Betrachtungen.

### **0.2.3.1 Planung und Ausführung von Demontageoperationen**

Es ist nicht damit zu rechnen, daß Instrukteure von vornherein über das notwendige Vorstellungsvermögen verfügen, um die Montagefolge komplexer oder auch einfacherer Aggregate mental zu planen und „aus dem Kopf“ in eine korrekte Anweisungsfolge umsetzen zu können. Wenn nicht vorher mittels Simulation und ggf. einem alle Beschränkungen einbeziehenden Probehandeln zwischen Konstrukteur und Instrukteur eine Übereinkunft über die vorzunehmenden Montageschritte erzielt wurde, dann wird es nicht ausbleiben, daß sich die beiden in Sackgassen bewegen. Mit anderen Worten: Der Instrukteur erkennt erst während oder nach der Ausführung der Aktion, daß sie nicht auf das vom ihm verfolgte Ziel hinführt. In einem solchen Fall muß nicht nur Intervenierbarkeit (siehe unten) sichergestellt sein, sondern es müssen auch die Primitiva für die Demontage zur Verfügung stehen (Schraube lösen, Schraube herausdrehen, Abziehen, Abnehmen, . . . ). Es muß ferner dafür gesorgt sein, daß während der Montageplanung Wiederaufsetzpunkte definiert werden, die bei einer Demontageplanung Zielpunkte sein können.

### **0.2.3.2 Verschränkte Planung von Sensor- und Aktoreinsatz unter Unsicherheit**

Die beweglichen bzw. mitbewegten Sensoren des Robotersystems sind so einzusetzen, daß sie die unter einer gegebenen Erfassungsaufgabe notwendige Information aus einer sich ständig verändernden Umwelt extrahieren. Erwünscht ist dabei die nahtlose Integration in die Planung der Aktorbewegungen, insbesondere natürlich derjenigen Aktoren, die einen mitbewegten Sensor (optisch, taktil) tragen. Die Folge nach Abbildung 22 illustriert das Problem für verschiedene Folgezustände einer Ausgangsszene, die durch die Ausführung bestimmter Aktionen durch den Roboter entstehen: Objekte ändern während des Ablaufs der einzelnen Manipulationen ihre Lage, das Erkennungssystem detektiert jeweils unterschiedliche Objekte (schlimmstenfalls werden solche, die vorher erkannt wurden, im weiteren Verlauf nicht mehr erkannt), und je nach Objektlage liefert der Erkenner unterschiedliche Genauigkeiten bei der Lagebestimmung. Bei räumlich verteilter Erfassung durch Kameras ergibt sich darüber hinaus das Problem, daß korrespondierende Punkte eines Objekts gefunden werden müssen, obwohl sich die Ansichten von Sicht zu Sicht fundamental unterscheiden (Leiste mit Normalansicht vs. Leiste aus Seitenansicht) – verstärkt durch die Schwierigkeiten bei der Teilüberdeckung. Ziel muß es hier also sein, Sichten zu erzeugen, die es erlauben, von möglichst vielen Objekten korrespondierende Teilansichten zu erzeugen, die dann von der Datenfusionskomponente zu einem Gesamtbild zusammengefügt werden können. Der Einsatz der Sensoren erfolgt gemischt parallel (für gleichzeitig arbeitende Sensoren) und sequentiell (für mitbewegte Sensoren). Für ihre zeit- und kostenoptimale Verwendung ist es also erforderlich, daß das Planungssystem folgenden Randbedingungen Rechnung trägt:

- Kenntnis des prinzipiell möglichen und jeweils aktuellen Erfassungsvermögens (z.B. Sichtwinkel, Standorte) der einzelnen Sensoren ,
- Kenntnis der Erfassungsaufgabe bzw. der aus der Handhabungsaufgabe folgenden Erkennungsaufgaben (z.B. genaue Bestimmung von Greifpunkten),
- Kenntnis der aktuellen Absichten des Einsatzes der Aktoren,
- Kenntnis der Anforderungen der Fusionskomponente,



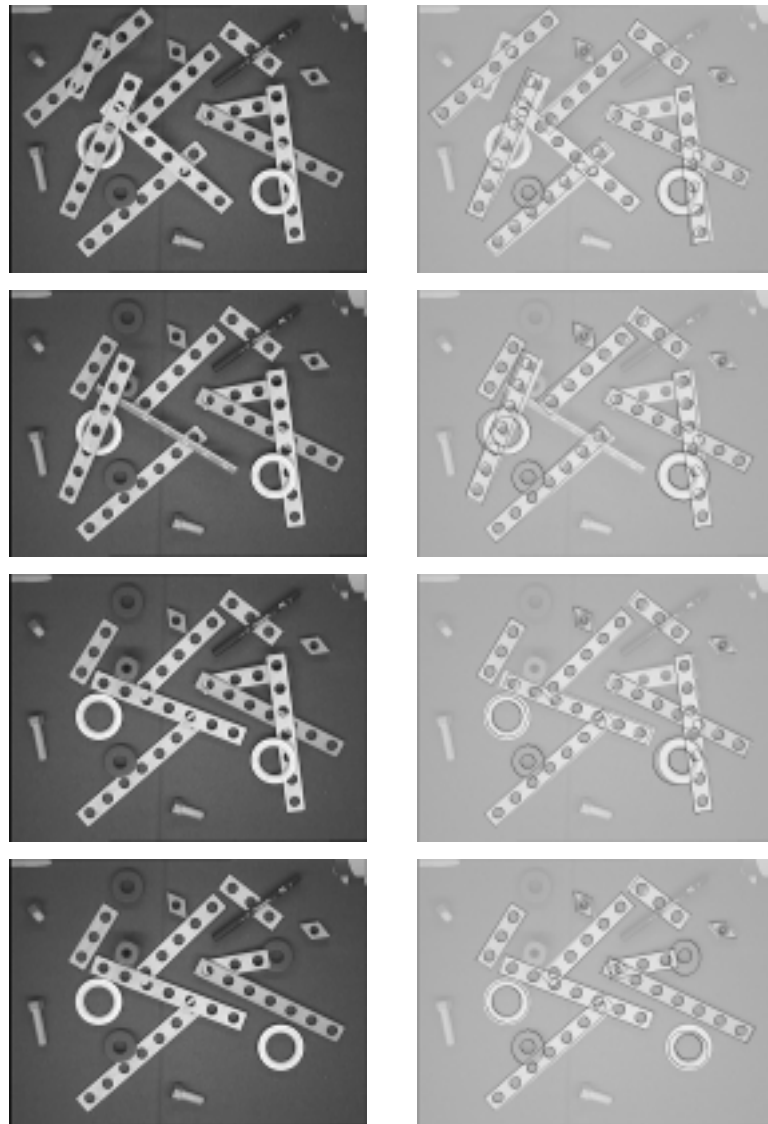


Abbildung 22: Folge von Szenen nach Manipulation durch den Roboter (von oben nach unten). Links Ausgangsbild, rechts visuelle Erkennung mit Kennzeichnung der rekonstruierten Lage. Nicht nur die Resultate der Bildverarbeitung ändern sich (Genauigkeit, Menge der erkannten Objekte), sondern es ändert sich auch die Objektlage: die Dreilochleiste rechts in Bildmitte wird erst nach dem Abheben der Siebenlochleiste (rechts, senkrecht) erkannt (Zeile 3 nach Zeile 4); die Siebenlochleiste in der Bildmitte fällt während des Abhebens der Fünflochleiste vom Würfel auf ihre schmale Seite.

- Bewertung des von einem bestimmten Sensor gelieferten bzw. des von ihm in einer bestimmten Situation zu erwartenden Ergebnisses.

Perspektivisch geht es hier darum, Erfassungsstrategien zum „Nachschauen“ zu entwickeln, die die eigene „Körperlichkeit“ des Robotersystems mit in Rechnung stellen. Es ist eine Umgebung zu schaffen, die es dem Robotersystem erlaubt, eine Selbstrepräsentation über seine sensorischen Fähigkeiten und ihre Verwendung zur zielgerichteten Durchführung der Montageoperationen zu entwickeln (Aufbau eigener Repräsentationsschemata).

### 0.2.3.3 Datenfusion

Nachdem über eine leistungsfähige Sensoreinsatzplanung die zur Lösung einer Aufgabe erforderliche Infor-

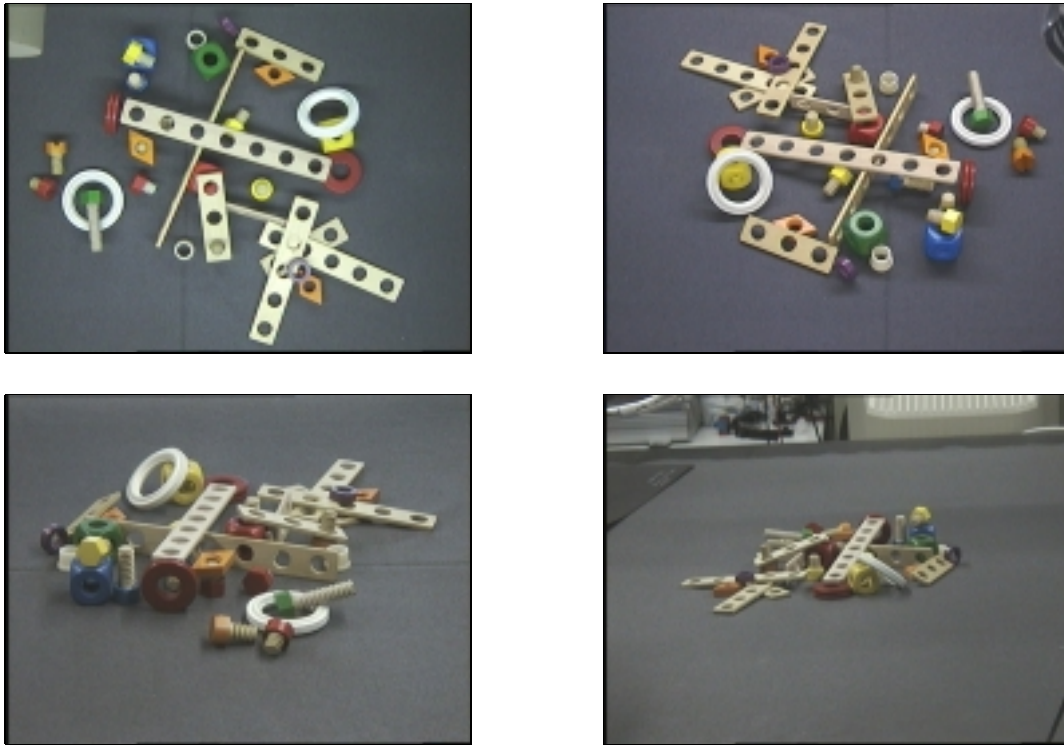


Abbildung 23: Unterschiedliche Ansichten derselben Szene, wie sie sich aus verschiedenen Blickpunkten mit unterschiedlichen Erfassungswinkeln ergeben. Hier hat auch ein menschlicher Betrachter Schwierigkeiten, korrespondierende Objekte sofort zu identifizieren.

mationsbestandteile extrahiert wurden, müssen diese unter Umständen sehr verschiedenen Bestandteile nun zu einer plausiblen Gesamtsicht integriert werden (siehe Abbildung 23). Wie bereits ausgeführt, sind die statistischen Verfahren zur Datenfusion bereits gut entwickelt. Normalverteiltes Sensorrauschen ist damit zwar gut behandelbar, es führt jedoch bei optischen Sensoren gegenüber den nichtlinearen Verzerrungen (Linsenfehler, Perspektive, Farbwerte) nur zu einem vernachlässigbaren Fehler. Um mit diesen Sensoren in vertretbarer Zeit zu einem möglichst exakten Abbild der wahren Objektparameter zu kommen, sind Techniken zu entwickeln zur

- Modellierung des Verhaltens der Sensoren (Abbildungsverhalten in Abhängigkeit von der Lage, von Umwelteinflüssen, von aufgewendeter Zeit) über die rein stochastische Charakterisierung hinaus unter Einbezug z.B. von Erfahrungswissen;
- Aufstellung einer Bewertungsfunktion zur Auswahl der Merkmale, deren Extraktion verspricht, für die Erkennungsaufgabe wesentliche Information zu liefern (Steuerung des Fusionsprozesses auf unterer Ebene);
- Steuerung des Fusionsprozesses auf höherer Ebene (Fusionsfolgeoptimierung), also die Auswahl von nacheinander angebotenen Sensorergebnissen zu einem Gesamtbild.

Wenn eine automatisch trainierbare Bewertungsfunktion für die spezifischen Leistungen der Einzelsensoren zur Verfügung steht, können wirklich redundante Sensoren praktisch implementiert werden. Ein solcher Sensor wird an das Sensornetzwerk angeschlossen und dann durch aktives Training unter Mitbenutzung seiner eigenen Beweglichkeit und bei Hilfestellung der Aktorik zum Vormachen vollständig autonom eingebunden. Mit anderen Worten: wenn ein komplexer Sensor hinzugefügt wird, wird er automatisch „eingelernt“, die zeitaufwendige explizite Programmierung und Kalibration entfällt vollständig.

#### 0.2.3.4 Interaktives Lernen neuer Montagesequenzen

Grundlage für die eigenständige Planung und nachfolgende Ausführung von längeren Ablauffolgen zur Konstruktion von Aggregaten ist die prototypische Durchführung einer Sequenz und die Benennung des Ergebnisses durch den Instrukteur. Während dieser Durchführung werden die Aktionsfolgen und die korrespondierenden Resultate der Umwelt- und Objekterkennung in einer geeigneten Repräsentationsform aufgezeichnet. Zunächst ist es sicher zweckmäßig, nur Lernvorgänge mit dem Ziel der selbständigen Wiederausführung auf der Ebene von Teilaggregaten, etwa des Flugzeugs, anzustreben (Leitwerk, Rumpf, Fahrwerk, Propeller, . . . ). Sobald diese erfolgreich gelernt werden können, sollen die Teilaggregate – zunächst ausgehend von einer standardisierten Konfiguration für die einzelnen Bauelemente auf dem Montagetisch – robust und völlig autonom gebaut werden.

Ein sinnvolles Ziel ist ferner die Verknüpfung mit dem Modul zur Montagefolgeplanung in der Weise, daß später auch *modifizierende Anweisungen* gegeben werden können. In einfacheren Fällen soll dies keinen weiteren Lernvorgang erfordern, bei aufwendigeren Modifikationen sollte zumindest erkannt werden können, welche der bereits gelernten alternativen Anweisungsfolge durchzuführen ist. Beispiele:

- > „Bau mir ein Leitwerk, aber mit *grünem* Würfel“ (gelernt wurde für einen blauen),
- > „Bau mir die Tragflächen, aber nimm *statt der Siebenlochleiste die fünflöchrige*“ (gelernt wurde nur für die Siebenlochleiste),
- > „Bau jetzt ein Fahrwerk, aber für das *große Flugzeug*“ (gelernt wurde für das große und das kleine Flugzeug).

Eingeführt werden sollten schließlich spezielle Anweisungen (als eine Klasse von *Metainstruktionen*), die ausschließlich für den Lerndialog Anwendung finden, sogenannte *bedingte Instruktionen*. Damit wird es möglich, Aktionsfolgen von vorhergehenden Aktionsfolgen, vom Systemzustand und vom sensorischen Input abhängig zu machen. Beispiel:

- > „*Wenn Du jetzt eine Schraube und eine Dreilochleiste siehst, kannst Du sie auf den grünen Würfel schrauben. Sonst nimmst Du eine Fünflochleiste und schraubst sie an die Seite und sagst mir Bescheid. Danach fragst Du nach der Dreilochleiste*“.

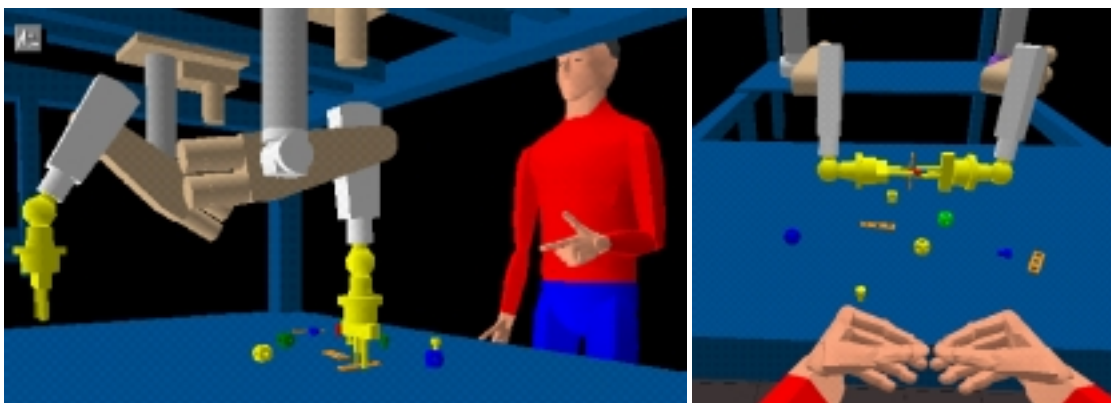


Abbildung 24: Kombinierte Interaktion über Sprache für die gewünschte Operation und Zeigen für die räumliche Relation. Links: einfaches Zeigen. Rechts: Vormachen einer komplexen Operation (Schrauben mit Richtung).

Es sind hier Ansätze für zwei Bereiche zu entwickeln und zu integrieren:

1. Lernen von diskreten Ereignisabläufen in Abhängigkeit von Sensormustern<sup>22</sup> bzw. Umweltkonstellationen unter Einbezug von automatisch zu erzeugenden Unterzielen und der Integration der reaktiven Planung,
2. Ableitung und Erfragen von *Erklärungen* über den Prozeßverlauf, um kausales Montagewissen zu erwerben und dieses Wissen bei analogen Aufgaben erneut einzusetzen,
3. Verhaltensbeobachtung eines bestimmten Instruktors während der Führung des Lerndialogs.

Bei der *Verhaltensbeobachtung* des Instruktors sind seine sprachlichen und gestischen Anweisungen zu analysieren, die dazu dienen, den Lernvorgang zu steuern. Zum einen beobachtet das Robotersystem den Instruktors und erkennt dabei dynamische Ereignisse, die wiederholt werden können. Auf der Ebene der Anweisungen zur Steuerung des Lernvorgangs sind neben der rein sprachlichen weitere Erfassungsmodalitäten zu kombinieren. Sie sollen integriert werden, sobald sie robust einsetzbar sind. Wir geben sie dennoch hier an, um die Richtung anzudeuten:

- Zeigeposition. Über die relativ einfache Auswertung von Zeigegesten mit gestreckten Zeigefinger oder mit einem speziellen Stift können in Verbindung mit der Spracheingabe nicht nur Objekte disambiguiert, sondern der Roboter kann auch zu einem bestimmten Objekt dirigiert werden (siehe Abbildung 24).
- Instruktionsqualifikation. Über komplexere Gesten können Anweisungen zur Modifikation bzw. exakterer Spezifikation sprachlicher Instruktionen gegeben werden: Reichen, Schrauben, Führen, Zusammen-Tragen; siehe Abbildung 24 rechts.
- Sprecherlokalisierung. Eine wesentliche Erleichterung kann es für den Prozeß der Intentionserkennung sein, wenn der Ort des Sprechers bekannt ist. Dies gilt speziell für die Einbeziehung räumlicher Relationen („Links von *mir*“; „Links von *dir*“).
- Blickrichtung. Die ungefähre Abschätzung des Blickwinkels des Instruktors kann ganz offensichtlich ebenfalls die Erkennung seiner Intention wesentlich erleichtern.

Für alle diese Modalitäten sind Erfassungstechniken zu entwickeln bzw. anzupassen. Es ist außerdem zu studieren, welche Zeigegesten, Erläuterungsgesten und Blickwinkel in einem Montagedialog eine Rolle spielen, weil der Instruktors damit seine Absichten unterlegt. Es soll in diesem Zusammenhang nicht unerwähnt bleiben, daß es sehr reizvoll wäre, wenn der Instruktors mit dem Robotersystem in Verbindung mit einer geeigneten Visualisierung über den Bauplan des Flugzeugs zunächst rein simulativ eine Operationenfolge aushandeln könnte; damit ließe sich die Ausführung des Dialogs während der realen Montage sicher deutlich abkürzen. Das gleiche gilt, wenn während der Ausführung des Dialogs eine Prädiktion des Verhaltens des Instruktors erreicht werden könnte.

### 0.2.3.5 Abstraktion und Transformation von Fertigkeiten

Hier geht es um die Extraktion der für die generischen Klassen von Handhabungsvorgängen wesentlichen Kenngrößen; langfristig kann auch die Abbildung des Vorgangs auf bestimmte physikalische Grundgesetze (Hooksches Gesetz, Reibungskegel, . . . ) ins Auge gefaßt werden. Die zentrale wissenschaftliche Fragestellung ist die Aufstellung minimaler expliziter Repräsentationen für die wichtigen Parameter der Handhabungsoperationen durch das Robotersystem selbst. Es steht dies in unmittelbarer Analogie zum *Begriffslernen*: so wie dabei angestrebt wird, das Konzept „Schraube“ unabhängig davon, ob es sich um eine Baufix-Holzschraube oder eine kleine Metallschraube handelt, mit einer Semantik zu füllen (die ohne weitere Erklärungen auch auf neue Schraubentypen paßt), ist es hier das Ziel, Handlungen bei neuen Objekten unter Kenntnis der wesentlichen

<sup>22</sup>Hierbei sind Sensormuster antizipatorisch einzubeziehen; das dazu notwendige Wissen soll aus dem Vergleich von Sensormustern und bekannter, zugrundeliegender Objektanstellung abgebildet werden (Ableitung der Abbildungsfunktion des Sensors; zur Sensormodellierung siehe Abschnitt 0.2.3.3).

Objektcharakteristika *ausführen* zu können, ohne daß vorher ein erneuter Trainingsvorgang erforderlich wird, mit ihnen also eine allgemeine Handlungssemantik und -pragmatik zu assoziieren (und nach Möglichkeit auch, ohne dabei vorher eine große Menge unterschiedlicher Fälle bearbeitet zu haben [LKW97] bzw. eine ständige Rückkopplung mit dem Menschen zu benötigen [NYM99]).<sup>23</sup>

Mit dieser Abstraktion soll zweierlei erreicht werden: zum einen soll die Anpassungsfähigkeit an strukturell neue Konstellationen<sup>24</sup> erhöht und der zeitliche Aufwand bis zu ihrer sicheren Handhabung verkürzt werden, zum anderen soll aber – längerfristig – ein grundsätzliches *Verständnis* des Robotersystems über sein eigenes Vermögen, Montagehandlungen vornehmen zu können, erreicht werden. Mit diesem *Eigenmodell* und darauf zu definierenden Emulationshandlungen (Selbstrepräsentation) kann der Konstrukteur dann auch bei neu auftretenden Situationen sofort über die wesentlichen Kenngrößen einer intendierten Operation in Abhängigkeit von den Parametern des Objekts bestimmen und über ihre Ausführbarkeit entscheiden („So etwas kann ich nicht schrauben, dazu wird meine Kraft nicht reichen“).

### 0.3 Steuerung und Architekturdesign verhaltensbasierter Roboter

#### 0.3.1 Verhaltensbasierte KI

Mitte der achtziger Jahre wendeten sich eine Reihe von KI-Wissenschaftlern, u.a. Rodney Brooks, MIT, Luc Steels, Vrije Universität Brüssel, und Rolf Pfeifer, Univ. Zürich, radikal von dem symbolorientierten Ansatz ab und formulierten einen Gegenentwurf eines KI-Forschungsprogramms: Die verhaltensbasierte KI. Brooks formulierte es besonders prägnant:

This suggests that problem solving behavior, language, expert knowledge and application, and reason, are all pretty simple once the essence of being and reacting are available. That essence is the ability to move around in a dynamic environment, sensing the surroundings to a degree sufficient to achieve the necessary maintenance of life and reproduction. This part of intelligence is where evolution has concentrated its time—it is much harder. [Bro91]

Da Intelligenz in der Natur nie ohne einen Körper auftritt und existieren kann, wird deren Einbettung in ihn als grundlegend betrachtet (engl. Embodiment). Diese Einbettung geschieht durch aufeinander aufbauende sensorimotorische Rückkopplungsschleifen. Will man versuchen, ein künstlich intelligentes System zu konstruieren, so geht dies nur mithilfe und auf der Basis von Robotern[Bro86b].

Im Unterschied zur sonst üblichen Robotik wird darüber hinaus verlangt, die Umgebung nicht künstlich so zu verarmen, daß sie für den Roboter präzise vorhersagbar wird. Diese Situietheit oder strukturelle Kopplung, wie es Humberto Maturana nannte, erfordert eine beständige interaktive Anpassung zwischen Umgebung und System (engl. Situatedness). Sowohl das System als auch die Umgebung verändern sich dadurch in systematischer Weise.

Als Bestandteil dieser Umgebung werden Systeme derselben Art explizit zugelassen und von einigen auch als Grundvoraussetzung gefordert. So hat Kerstin Dautenhahn schon sehr früh dafür plädiert, die “soziale Intelligenz” Hypothese aus der Primatenforschung ernst zu nehmen. Sie behandelte in ihren Experimenten zum Imitationslernen zwischen Robotern einige Aspekte dazu[Dau95].

Einher gehen diese Überlegungen mit dem bewußten Verzicht darauf, die Welt objektiv, vollständig und widerspruchsfrei zu modellieren. Begründet wird das durch Arbeiten aus der Gehirnforschung, in denen gerade immer deutlicher wird, daß jedes Gehirn sich seine eigene subjektive Wirklichkeit konstruiert[Rot94]. Gerade dies steht im Widerspruch mit den traditionellen logikorientierten Ansätzen der Kognitionswissenschaften.

---

<sup>23</sup>Zu einer kompletten Handlungssemantik gehört natürlich noch die Angabe von Vorbedingungen und Resultaten in einer geeigneten Darstellungsform.

<sup>24</sup>Schrauben in langes Gewinde des Klotzes vs. Schrauben in kurzes Gewinde der Rautenmutter oder Schrauben einer Leiste auf einen Würfel (ein Gewinde) vs. Schrauben eines Würfels auf einen Würfel (zwei hintereinanderliegende Gewinde).

In den letzten zehn Jahren sind eine Reihe von Roboterexperimenten durchgeführt worden, in denen diese Prinzipien zugrunde gelegt wurden. Die Roboter zeichnen sich alle dadurch aus, daß sie ihr beobachtbares Verhalten mithilfe mehrerer oft gleichzeitig aktiver sehr einfacher sensomotorischer Rückkopplungen erreichen. So wurden mithilfe verschiedener methodischer Ansätze sehr natürlich wirkende Bewegungsmuster für sechsbeinige Laufmaschinen realisiert.

Doch wurde auch immer klarer, worin die Schwierigkeiten dieses Ansatzes bestehen. Die einfachen (internen) Verhaltensweisen interagieren in hohem Maße miteinander. Bislang ist es möglich, Verhaltenssysteme, die zwischen zwanzig und vierzig derartiger Verhaltensweisen enthalten, zu konstruieren. Doch dann steigt die Komplexität für den menschlichen Konstrukteur so sehr, daß jede Veränderung unkontrollierbar wird bzgl. der schon vorhandenen Verhaltensweisen.

Dieser Schwierigkeit kann man versuchen mit allen Formen des (maschinellen) Lernens zu begegnen. Und tatsächlich werden viele der Kontrollprogramme als künstliche neuronale Netze realisiert. Aber auch damit oder ähnlichen Verfahren wie den genetischen Algorithmen gelang der Durchbruch zur größeren Komplexität bislang nicht.

Wie so oft in der KI, gelang es mit einem sehr einfachen Prinzip, rasche Anfangserfolge zu erzielen. Doch jetzt stellt sich die Frage, wie komplexere Systeme entwickelt werden können: Wie kommt man von den heute üblichen 20-40 Verhaltensweisen zu tausend, millionen und noch mehr Verhaltensweisen. Dies ist das Scaling-Up Problem, zu dem es von niemandem bislang ein überzeugendes Konzept gibt. Und es ist mehr als blauäugig anzunehmen, daß man von einer Bewegungssteuerung für eine sechsbeinige Laufmaschine—sei sie noch so dicht an einer Heuschrecke modelliert—oder von Teelichtern einsammelnden Robotergruppen schnurstracks die Komplexität der Gehirne von Primaten erreichen kann oder daß damit die entscheidende Einsicht gewonnen wurde, wie ein solches Gehirn funktioniert. Der Abstand von den heutigen Robotern zu den “einfachsten” Lebewesen ist noch so gewaltig, daß selbst Daniel Dennett von “Roboacteria” sprach, um den Stand der Technik zu charakterisieren.

Am Anfang der verhaltensbasierten KI stand neben dem Körperbezug die grundlegende These, daß Intelligenz nur in einem *autonomen System* entstehen könne:

„An agent is autonomous if it is able to cope with all the consequences of its actions to which it is subjected while remaining viable as a task-achieving agent in the world it operates in.“ [Smi95]

Dabei argumentiert Smithers, daß Autonomie wie andere biologische Phänomene auch nicht eine schwarz-weiß Unterscheidung sondern sie tritt in graduellen Abstufungen auf. Smithers fährt fort:

„It is, however, a necessary prerequisite for intelligent behaviour: The more or less autonomous an agent, the more or less its potential for intelligent behaviour.“ [Smi95]

Weiterhin ist die Orientierung an biologischen Vorbildern wichtig[PS99]. Dies erfolgt teilweise durch “biometrische” Roboter, d.h. die Morphologie der Roboterkörper und ihrer Bewegungsmöglichkeiten ist an biologischen Vorbildern orientiert, z.B. bei Laufmaschinen oder kriechenden Roboterschlängen. Teilweise wird die “neuronale Architektur” nachgebildet, z.B. das Navigationsverhalten von Heuschrecken oder von Wüstenameisen ([Web93b, Web93a, MLRPW98]).

Soziales Verhalten wird unter anderem von [Dau95, DC97, Mat95] untersucht. Bei den dort beschriebenen Interaktionen können zwei grundlegende Klassen unterschieden werden, die anonyme ([Mat95]) und die individuelle ([Dau95]) Interaktion. Bei letzterer treten neben die bloße Wahrnehmung anderer Agenten auch Konzepte von Selbstwahrnehmung und Fremdwahrnehmung.

Brooks vertritt außerdem die stark umstrittene These, daß für intelligentes Verhalten keine interne Repräsentation von Wissen im Agenten notwendig ist [A.B91, Bro95].

Oft wird Intelligenz als ein Mittel von Lebewesen angesehen, um die Überlebensfähigkeit zu maximieren. Doch das ist in einem trivialen Sinne natürlich auch wahr für andere Organe und deren Funktionalität als das

Gehirn und seine Intelligenz. Auch die physikalische Umgebung ist für alle Lebewesen gleich und deshalb kann alleine daraus nicht die Notwendigkeit erwachsen sein, Intelligenz zu entwickeln. Eine Hypothese ist aber, dass die steigende Komplexität von Sozialbeziehungen und von Verhaltensalternativen in ähnlichen Situationen den notwendigen evolutionären Druck erzeugt:

„If an individual can only survive in a social group with individualized social relationships and the variety of possible behaviors is beyond some threshold it must develop intelligence as a means to forecasting the behaviors of other individuals, primarily in the same group, and imagine its own behaviors as possible responses.“[Chr99]

Zum Abschluß dieser kurzen Einführung basierend auf[Ste94a] noch die Erläuterung einiger Begriffe, die in der verhaltensbasierten KI verwendet werden<sup>25</sup>:

**Funktionalität:** „A functionality is something that the agent needs to achieve, . . . Other terms used for functionality are task, goal, and competence. Functionalities belong to the descriptive vocabulary of the observer.“

**Behavior/Verhaltensweise:** „A behavior is a regularity in the interaction dynamics between the agent and the environment ( . . . ) One or more behaviors contribute to the realisation of a particular functionality. Behaviors belong to the descriptive vocabulary of the observer.“

**Mechanismen:** „A mechanism is a component of the agent that plays a role in establishing a particular behavior. Mechanisms include body parts, sensors, actuators, particular transformations of sensory signals into action parameters, internal states and processes updating them, adaptive mechanisms, etc.“

**Behavior System/Verhaltenssystem:** „A behavior system is the set of all mechanisms which play a role in a particular behavior.“

**Emergenz:** „A behavior is emergent if it can only be defined using descriptive categories which are not necessary to describe the behavior of the constituent components. An emergent behavior leads to emergent functionality if the behavior contributes to the system’s self-preservation and if the system can build further upon it.“

In der verhaltensbasierten KI wird ein Roboter dadurch programmiert, indem Mechanismen gebaut/implementiert werden, die zusammen Verhaltenssysteme bilden. Verhaltenssysteme bestehen zumeist aus mehreren sensomotorischen Rückkopplungen, die parallel aktiv sind. Durch diese Verhaltenssysteme wird bei der Interaktion von Agent und Umwelt ein Verhalten von außen sichtbar. Dieses ist zielgerichtet oder aufgabenorientiert, wenn es eine Funktionalität erfüllt. Ein Beispiel für emergentes Verhalten ist eine Wandverfolgung, bei der ein Verhaltenssysteme des Agenten eine Hindernisvermeidung und ein zweites eine Hindernissuche ist. Von außen ist (bei ausreichend guter Abstimmung) dann ein Wandverfolgungsverhalten sichtbar. Emergentes Verhalten wird als ein Schlüssel zur Intelligenz angesehen, da es qualitativ neue Verhalten erzeugt, die nicht explizit implementiert worden sind.

### 0.3.2 Steuerungsarchitekturen

Wie in vielen anderen Teilgebieten der KI so spielen auch in der verhaltensbasierten KI Architekturen eine wichtige Rolle. Und auch hier finden wir viele Variationen, um mit denselben Herausforderungen fertig zu werden:<sup>26</sup>

- Reaktivität unter (mehr oder weniger harten) Realzeitbedingungen.

<sup>25</sup>Leider ist die Begrifflichkeit nicht einheitlich auch nicht bei denselben Autoren.

<sup>26</sup>Dieser Abschnitt basiert wesentlich auf [Str96] und [SS99].



- Beherrschung der Komplexität.
- Erweiterbarkeit basierend auf Erreichtem.
- Nachvollziehbarkeit der Experimente.
- Korrespondenz zu theoretischen oder methodologischen Grundannahmen.

Viele dieser Architekturen sind eher ad hoc entstanden, zugeschnitten auf ganz bestimmte Fragestellungen oder spezielle Experimente bzw. Robotertypen. Dabei kann man für dasselbe System auch unterschiedliche Perspektiven einnehmen, um eine Architektur zu definieren. Wir hier beschränken uns ausschliesslich auf die Steuerungsarchitektur und lassen dabei z.B. Systemimplementierungsaspekte weitgehend aussen vor.

Die Architekturansätze lassen sich nach [Ste94a] basierend auf dem verwendeten Programmieransatz in folgende vier Gruppen einordnen:

- Künstliche neuronale Netze.
- Schaltkreisentwurf.
- “Algorithmischer” Ansatz.
- Dynamische Systeme.

Im folgenden werden exemplarisch als “algorithmischer” Ansatz die Subsumptionsarchitektur von Rodney Brooks und als Ansatz basierend auf dynamischen Systemen die PDL-Architektur von Luc Steels dargestellt. Zu beiden gibt es eine vielfältige Literatur, dokumentierte Programmiersprachen und sie wurden in verschiedenen Labors für unterschiedliche Roboter eingesetzt.

### 0.3.3 Subsumptionsarchitektur

Die Subsumptionsarchitektur war einer der ersten Ansätze in der verhaltensbasierten KI, um mobile Roboter zu programmieren. Dabei wird hier unter Subsumption eine Methodik verstanden, nach der aufgabenorientierte Verhaltenssysteme oder Kompetenzen als Bausteine für das Gesamtprogramm definiert werden. Sie werden in aufeinander aufbauenden Ebenen angeordnet. Das Entscheidungskriterium dafür ist, daß tiefere Ebenen (fast) nicht verändert werden sollen, wenn eine höhere Ebene (nachträglich) hinzugefügt wird. Jede Ebene kann direkt auf die aktuellen Sensorwerte zugreifen und Aktuatorwerte erzeugen, die dann in Motorsignale umgesetzt werden. Dies erlaubt eine (quasi-)parallele Architektur.

Jede Ebene selber wird durch eine Menge von Modulen implementiert. Jedes der Module wiederum wird als ein erweiterter endlicher Automat (engl. Augmented Finite State Machine AFSM) realisiert. Die möglichen Erweiterungen bestehen in einer internen Uhr und zusätzlichen internen Variablen. Diese Automaten werden untereinander durch ihre Ein- bzw. Ausgangskanäle oder mit den Sensoren und Aktuatoren verbunden. Eine spezielle Programmiersprache, Behavioral Language, erlaubt eine abstraktere Definition dieser Automaten [Bro90].

Die Idee ist nun, dass mit der Spezifikation und Implementierung der niedrigsten Ebene begonnen wird. Die nächsthöhere Ebene wird dann durch “verhindernde” (engl. Inhibited Wire) bzw. “überschreibende” (engl. Suppressed Wire) Kanäle angebunden. Die verhindernden Kanäle blockieren für eine festgelegte Zeit den Ausgangskanal eines Moduls während ein überschreibender Kanal den Wert auf einem Eingangskanal eines Moduls ersetzt. Module einer höheren Ebene können also solche einer niedrigeren Ebene subsumieren, daher die Bezeichnung Subsumptionsarchitektur.

Eine der eindrucksvollsten ersten Roboter, die so programmiert wurden, war Ghengis[Bro]. Es handelt sich hierbei um eine sechsbeinige Laufmaschine, mit zwei Bewegungsfreiheitsgraden je Bein, einer verteilten Rechnerarchitektur (basierend auf vier Mikrokontrollern), diversen Sensoren und vollständiger Energieversorgung an

Bord. Das Steuerungsprogramm bestand aus acht Ebenen und insgesamt 57 Modulen, d.h. endlichen Automaten. Der Roboter wurde entworfen, um in unebenem Gelände sich sicher fort zu bewegen. Dies demonstrierte er auch auf eindrucksvolle und überzeugende Art und Weise.

Der anspruchvollste Roboter, der nach dieser Methodik seit 1993 fortlaufend weiter entwickelt wird, ist der Humanoide „Cog“ (siehe Abbildung 4). Leider lässt sich aufgrund der publizierten Literatur nicht viel über die Erfahrungen sagen, die mit der Subsumptionsarchitektur hierbei gemacht wurden. Mit die spannendsten Fragen sind, inwiefern Lernverfahren eingesetzt werden können, um einzelne Module oder sogar Ebenen durch Erfahrungen des Roboters wenigstens teilweise erwerben zu lassen, und wie gut sich die Prinzipien bei der steigenden Komplexität haben durchhalten lassen<sup>27</sup>.

### 0.3.4 Dynamische Systeme als Architekturprinzip: PDL

PDL bedeutet “Process Description Language” und ist an der Universität Brüssel(VUB) von Luc Steels entwickelt worden. Die Sprache ist in die Programmiersprache C eingebettet und wurde in zahlreichen Experimenten auch mit verschiedenen (mobilen) Robotern erprobt [SSH97, Ste94b]. Die Einbettung erlaubt neben den wenigen PDL-spezifischen Sprachkonstrukten beliebigen C-Code.

Ein PDL-Programm besteht aus sogenannten Prozessen (prozeduraler Anteil) und Quantitäten (Datenanteil). Das System läuft in Zyklen mit einer konstanten Periode (z.B. 40 Zyklen pro Sekunde). Die Grundidee steckt in der inkrementellen Berechnung des Gesamtverhaltens durch die jeweiligen Beiträge der parallel ablaufenden Prozesse in jedem Zyklusschritt. Damit erhält man u.a. eine implizite Synchronisation. Es gibt drei Sorten von Daten: Sensordaten, interne Daten und Aktuatordaten. Diese *Quantitäten* (engl. Quantity) nehmen (ganzzahlige) Werte aus einem vorgegebenem Intervall an, z.B. zwischen 0 und 255.

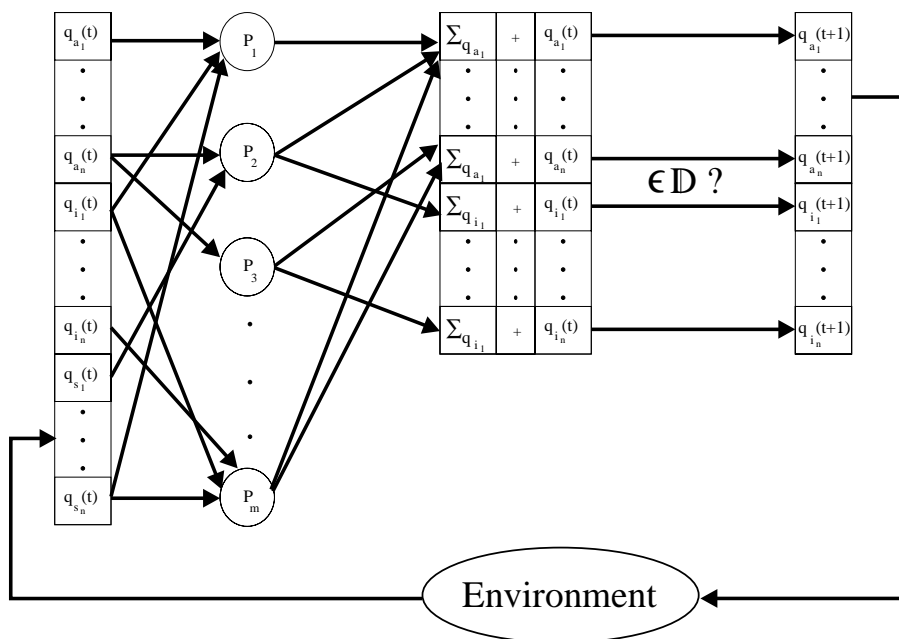


Abbildung 25: Ein kompletter PDL-Zyklus.

Die Abbildung 25 zeigt einen PDL-Zyklus in seinem zeitlichen Verlauf von links nach rechts. Die  $q_{x_i}(t) | x \in \{a, i, s\}$  sind die aktuellen Quantitätenwerte, wobei die Werte der Sensorquantitäten  $q_{s_i}$  jeweils zum Anfang eines Zyklus anhand der aktuellen Sensorenwerte aktualisiert werden.

<sup>27</sup>Es ist natürlich klar, dass jede genügend ausdrucksstarke Programmiersprache letztlich alles erlaubt zu programmieren. Die Frage richtet sich also dann darauf, wie “natürlich” sich mit der Behavior Language Cog programmieren liess.

Nun gibt es sogenannte *Prozesse*, die auf diesen Quantitäten operieren. Die Prozesse werden (quasi-)parallel einmal pro Zyklus abgearbeitet. Ein Prozeß ist ähnlich aufgebaut wie eine einfache Produktionsregel. Im linken Teil erfolgt eine oder mehrere Abfragen von Quantitäten  $\text{Value}(\text{Quantity})$ , im einfachsten Falle von Sensorquantitäten. Auf der rechten Seite stehen Inkremente oder Dekremente von (anderen) Quantitäten  $\text{Add-Value}(\text{Quantity}, \text{Additionsterm})$ , im einfachsten Falle von Aktuatorquantitäten. verfügt über zwei Möglichkeiten. Erstens kann er den Wert einer Quantity mit dem Befehl auslesen. Hierbei wird die aktuelle Quantität aber nicht sofort verändert, sondern das System speichert diesen additiven Einfluß in einer "Einflußsumme" zwischen.

Jeder Prozeß kann auf beliebig viele Quantitäten zugreifen und auch beliebig viele inkrementelle positive oder negative Einflüsse nehmen, die erst am Ende eines Zyklus addiert und wirksam werden. Quantitäten dienen also der indirekten Kommunikation zwischen den Prozessen und sind technisch globale Variablen für das gesamte Kontrollprogramm.

Am Ende eines Zyklus wird zu jeder Quantität die jeweilige "Einflußsumme" addiert und der resultierende Wert modulo des Werteintervalls steht im  $(t+1)$ sten Zyklus zur Verfügung. Einzige Ausnahme hierbei sind die Sensorquantitäten, da sie am Anfang eines Zyklus jeweils durch den aktuellen Sensorwert überschrieben werden. Bevor nun ein neuer Zyklus initiiert wird, aktualisiert das System die Aktuatoren mit den neuen Aktuatorwerten, die dann in entsprechende Veränderungen der Motorgeschwindigkeiten umgesetzt werden.

Solange man sich dieser wenigen und einfachen Sprachkonstrukte bedient und vorsichtig die Möglichkeit nutzt, beliebigen C-Code zu verwenden, erfüllt man die Grundforderungen der verhaltensorientierten KI. Die Prozesse als Abbildung zwischen Sensoren und Aktuatoren via der Quantitäten erlaubt die sensomotorische (Rück-)Kopplung. Die inkrementelle Veränderungen der Quantitätswerte und eine geeignete schnelle Zykluszeit (bei Geschwindigkeiten von 20-40 cm pro Sekunde sind 40Hz ausreichend) sichern eine hohe Reaktivität auf Unvorhergesehenes<sup>28</sup>. Und die wichtigste Forderung ist ebenfalls erfüllt: Es gibt keine zentrale Kontrolle und keine exklusive Auswahl der auszuführenden Aktion (hier: des Prozesses).

Daneben gibt es auch keine explizite Sensorvorverarbeitung oder eine Sensorfusion. Einerseits lassen sich hiermit viele Probleme sehr elegant lösen. Andererseits ergibt sich daraus eine hohe Herausforderung an den Entwickler, unerwünschte Interferenzen zwischen verschiedenen Prozessen unterschiedlicher Verhaltenssysteme zu verhindern. Das grösste Problem stellt aber die fehlende syntaktische Unterscheidung in verschiedenen abstrakte Prozesse dar. Es stellen sich ähnliche Probleme wie schon bei den ersten Produktionsregelsystemen für Expertensysteme.

Abstrakt kann ein PDL-Programm als eine lineare Approximation an ein (u.U. nicht-lineares) Differentialgleichungssystem interpretiert werden. Der PDL-Interpreter löst dann dieses (Differenzen-)Gleichungssystem anhand der jeweils aktuellen Sensordaten und der Vergangenheit des Systems (festgehalten in den Quantitäten). Deshalb kann man auch den umgekehrten Weg gehen, und direkt das Gesamtverhalten mithilfe eines formalen Gleichungssystems spezifizieren. Dies führt dann auch zu anderen Möglichkeiten, um die Schwierigkeiten sowohl bei PDL als auch der Behavior Language der Subsumptionsarchitektur zu überwinden. Ein konkreter Vorschlag dazu ist die "Dual Dynamics" Methode von Herbert Jaeger [Jae96].

### 0.3.5 Hindernisvermeidung mit Hilfe von Infrarotsensoren

Im folgenden wird an einem konkreten Beispiel mithilfe von PDL dargestellt, wie ein bestimmtes Verhalten entworfen und implementiert werden kann. Es ist fast unverändert aus [SS99] entnommen worden. Dort findet sich auch der vollständige Programmcode dafür und ausserdem ein komplexes Kontrollprogramm, das an die Grenzen der Leistungsfähigkeit des Ansatzes der verhaltensbasierten KI geht.

Die für dieses Beispiel verwendeten Roboter sind mechanisch mithilfe von **fischer technik** © gebaut worden. Sie haben—wie viele der experimentellen mobilen Roboter in den Labors—zwei Antriebsräder, ein Stützrad, einen

---

<sup>28</sup>Eigentlich gibt es hier sowieso nichts Unvorhergesehenes. Nur einmal eingeleitete Bewegungen lassen sich schon aus physikalischen bzw. Selbstschutzgründen nicht abrupt stoppen oder in ihr Gegenteil verkehren. Insofern ist jede Veränderung der Aktuatorwerte eher eine Modulation der aktuellen Bewegungsabläufe. Es muss also sichergestellt sein, dass diese Modulation rasch genug in bezug auf die Eigengeschwindigkeit und die der Situationsveränderungen erfolgt.

Ring von Infrarotsensoren und -sendern, Stosskontaktsensoren, Enkoder zur Messung der Radumdrehungen, usw.

Die Funktionalität der Infrarot-Hindernsvermeidung basiert auf den drei am Roboter vorne links, rechts und in der Mitte angebrachten Infrarotempfängern und den dazugehörigen rund um den Roboter verteilten Infrarot-sendern. Dabei soll die Stärke und der Zeitpunkt des Ausweichens parametrisierbar sein. Außerdem soll es eine Möglichkeit geben, dieses Verhalten zu inhibieren, falls es sich störend auf andere Verhalten auswirken könnte.

**0.3.5.1 Die Idee** In Prinzip ist diese Infrarothindernsvermeidung einfach dadurch realisiert, daß, wenn der rechte Sensor unter einem signifikanten Schwellwert der Roboter den linken Motor bremst und somit nach links ausweicht. Für den linken Sensor gilt die selbe sensomotorische mit dem rechten Motor. Sollte nun auch noch der mittlere Sensor unter dieser Schwelle liegen, wird dafür gesorgt, das der Motor, auf dessen Seite weniger Infrarot wahrgenommen wird nun stärker als der andere gebremst wird, bzw. der andere Motor sogar wieder beschleunigt wird, damit der Roboter sich nun möglichst schnell wegdreht. Ansonsten mag es passieren, daß der Roboter mit einem Hindernis kollidiert, weil sich die beiden sensomotorischen Rückkoppelungen gegenseitig aufheben. Wie die einzelnen Feinheiten dieses Behaviors in der Realisation aussehen folgt nun.

**0.3.5.2 Die Realisation** Mit der Quantität `IR_on` kann das Verhalten inhibiert werden. Dann wird überprüft, ob eine der drei Infrarotsensorquantitäten `LIR(links)`, `RIR(rechts)`, `MIR(Mitte)` einen vom Benutzer eingestellten Schwellwert unterschreitet<sup>29</sup>. Sind beide bedingungen erfüllt, so wird die Quantität `IR_active` auf eins gesetzt. Mithilfe der Variablen `weight` wird berechnet, ob der Roboter mehr Infrarot rechts (`weight=1`) oder links (`weight=0`) oder gleich viel (`weight=0`) registriert. Der Einfluß z.B. des mittleren Sensors berechnet sich dann wie folgt:

$$(1) \quad MEinfluß = (Schwellwert - Value(MIR)) / (Schwellwert / MFaktor) * weight$$

Die Konstante `MFaktor` dient zur Gewichtung der Einflußnahme<sup>30</sup>.

Wie sieht nun der Einfluß auf die Motoren aus? Mit dem `AddValue`-Befehl wird folgender Einfluß auf den linken Motor genommen:

$$AddValue(MotorL, \quad Drehe + \\ ((SFaktor * (Value(RIR) - Grundwert)) + MEinfluß) \\ *((100 - abs(Value(MotorL) - Value(MotorR))) / 100))$$

Dabei sorgt der Term  $((100 - abs(Value(MotorL) - Value(MotorR))) / 100)$ <sup>31</sup> dafür, daß die Drehung nicht zu hart ausfällt, sondern der Roboter nur noch geringeren zyklischen Einfluß auf den Motor nimmt, wenn der entsprechende Motor sowieso schon stark gebremst ist.

Da der Einfluß von der Stärke der empfangenen Infrarotsignale abhängen soll, wird der Term  $(SFaktor * (Value(RIR) - Grundwert)) + MEinfluß$  eingefügt. Der Grundwert bestimmt sich aus der Empfindlichkeit der Sensoren, wenn keine Reflexionen empfangen werden<sup>32</sup>. Mithilfe der Variablen `Drehe` wird eine schnelle Drehung auf der Stelle angestossen, falls er auf allen 3 Sensoren Hindernisse registriert. Ansonsten liegt `Drehe` auf Null:

$$(2) \quad Drehe = 5 * weight.$$

<sup>29</sup>Konstruktionsbedingt fällt der Wert aus der Sensorvorverarbeitung bei steigender Infrarotreflexion.

<sup>30</sup>MFaktor liegt normalerweise bei 3

<sup>31</sup>Die Motorwerte liegen im Intervall  $[-100, 100]$ .

<sup>32</sup>Dieser Wert liegt üblicherweise bei 180-190

Für den rechten Motor werden nur die beiden Vorzeichen bei *MEinfluss* und *Drehe* vertauscht:

$$\begin{aligned} \text{AddValue}(\text{MotorR}, & \quad -\text{Drehe} \\ & +((\text{SFaktor} * (\text{Value}(\text{RIR}) - \text{Grundwert})) - \text{MEinfluss}) \\ & *((100 - \text{abs}(\text{Value}(\text{MotorL}) - \text{Value}(\text{MotorR}))) / 100)) \end{aligned}$$

Insgesamt erhalten wir damit eine auch in Sackgassen gut funktionierende Hindernisvermeidung, die sich mittels des Reaktionsschwellwertes, sowie mit *SFaktor* und *RFaktor* sehr leicht an die entsprechenden Bedürfnisse anpassen läßt.

## 0.4 Navigationsverfahren für autonome mobile Roboter

Ziel der Forschung in dem Bereich autonomer mobiler Roboter ist die Entwicklung von Verfahren, die es mobilen Robotern erlauben, selbständig und zuverlässig über längere Zeiträume hinweg in ihrer Umgebung zu operieren. Dazu benötigen solche Systeme verschiedene Fähigkeiten. Zunächst müssen sie in der Lage sein, selbständig ein Modell der Umgebung zu lernen oder gegebenenfalls auch zu adaptieren. Sofern einmal ein Modell gegeben ist, sollen sie möglichst effiziente Wege zu ihren Zielpunkten berechnen können. Dabei müssen sie sich stets auch lokalisieren, d.h. ihre Position innerhalb ihrer Umgebung berechnen.

Dabei stellt sich das Problem, daß nicht nur die Informationen, die der Roboter über seine Sensoren erhält, sondern auch seine Aktionen stets mit Fehlern behaftet sind. Beispielsweise liefern Ultraschallsensoren aufgrund ihres breiten Öffnungswinkels häufig nur sehr ungenaue Informationen über den Abstand zu Hindernissen [BEF96, Eve95]. Darüber hinaus sind aber auch die Bewegungsaktionen eines mobilen Roboters mit Unsicherheit behaftet. Dies liegt zum einen daran, daß für die Ermittlung der zurückgelegten Strecke bzw. der durchgeführten Drehung ebenfalls wieder Sensoren verwendet werden. Andererseits führen aber auch andere Effekte wie z.B. Rutschen und Drift zu Meßfehlern. Verfahren zur Bestimmung von Umgebungsmodellen, zur Pfadplanung und zur Lokalisierung müssen daher stets berücksichtigen, daß sowohl die Sensorinformationen als auch die Aktionen eines mobilen Roboters mit Unsicherheiten behaftet sind.

Bei der Umgebungsmodellierung und der Lokalisierung handelt es sich um Probleme der Zustandsschätzung. Während bei der ersten Aufgabe der Zustand der Umgebung bestimmt werden soll, liegt bei dem zweiten Problem die Aufgabe darin, die Position des Roboters zu ermitteln. Im folgenden wollen wir nun Techniken vorstellen, welche es erlauben, diese Zustände robust schätzen zu können. Um dabei geeignet mit den unsicheren Informationen umgehen zu können, verwenden wir probabilistische Methoden.

### 0.4.1 Umgebungsmodellierung

Aufgabe bei der Umgebungsmodellierung ist die Erstellung oder Adaption eines Modells der Umgebung auf der Basis der Informationen, die der Roboter über seine Sensoren von seiner Umgebung erhält. Dieses Problem ist schematisch in Abbildung 26 dargestellt. Aufgezeigt sind hier alle Ultraschallmessungen des Roboters entlang einer Trajektorie durch seine Umgebung. Aufgrund dieser Abbildung wird bereits deutlich, wie schwierig es ist, auf der Basis solch ungenauer Sensordaten, geeignete Informationen über die Umgebung des Roboters abzuleiten.

Bevor wir genauer auf ein Verfahren eingehen werden, wollen kurz noch einmal unterschiedliche Arten von Umgebungsmodellen diskutieren. Hierbei wollen wir unterscheiden zwischen

- topologischen,
- geometrischen und
- ikonischen bzw. rasterbasierten Modellen.

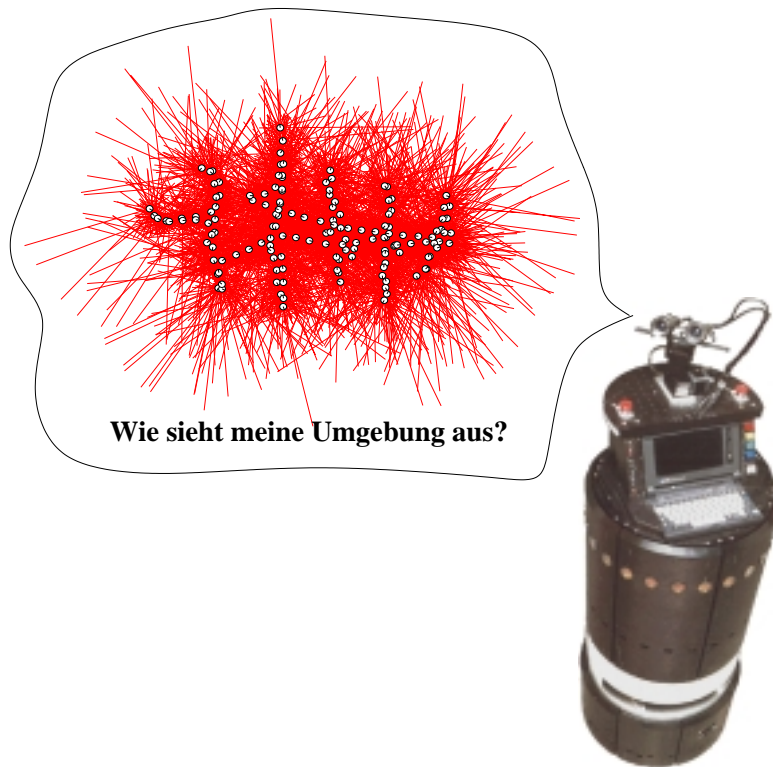


Abbildung 26: Das Problem der Umgebungsmodellierung

Unter einer topologischen Karte der Umgebung versteht man üblicherweise einen annotierten Graphen, dessen Knoten für den Roboter relevante Orte der Umgebung darstellen. Diese Knoten sind genau dann durch eine Kante miteinander verbunden, wenn sie unmittelbar voneinander erreichbar sind. Im Vergleich zu den anderen Modellierungsmethoden liegt der Vorteil dieser Repräsentation in der Kompaktheit. Darüber hinaus sind diese Karten sehr gut für die Pfadplanung geeignet, da sich dieses Problem dann auf die Erreichbarkeit in Graphen reduzieren läßt. Ein Nachteil topologischer Karten ist jedoch die große Abstraktion von den Sensorinformationen. Die Akquisition von Karten dieser Form oder die Navigation mit Hilfe solcher Karten setzt daher entsprechende Abstraktionsmethoden für die Identifikation von Knoten voraus. Trotz dieser Schwierigkeiten, gibt es Systeme, die topologische Modelle lernen [CKB96, TB96, SK97c, Thr98a] oder für die Navigation nutzen [NPB95, KCK96, HK96, KS98].

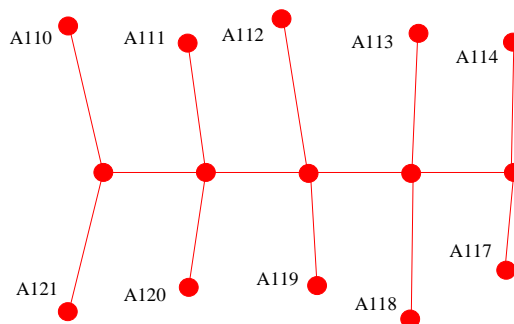


Abbildung 27: Topologische Karte einer typischen Büroumgebung

Die detaillierteste Modellierungsform stellen 2D- und 3D-Modelle dar. Typische Varianten solcher Modelle

sind in Abbildung 28 dargestellt. Wenn die verwendeten graphischen Primitive darüber hinaus noch z.B. in hierarchischen Modellen zu komplexen Einheiten zusammengefaßt werden, so lassen sich Umgebungen damit geometrisch sehr exakt beschreiben. Nachteile dieser Modelle liegen jedoch in ihrer schwierigen Lern- und Aktualisierbarkeit, denn sie erfordern geeignete Verfahren zur Detektion und Lokalisierung der einzelnen Objekte.

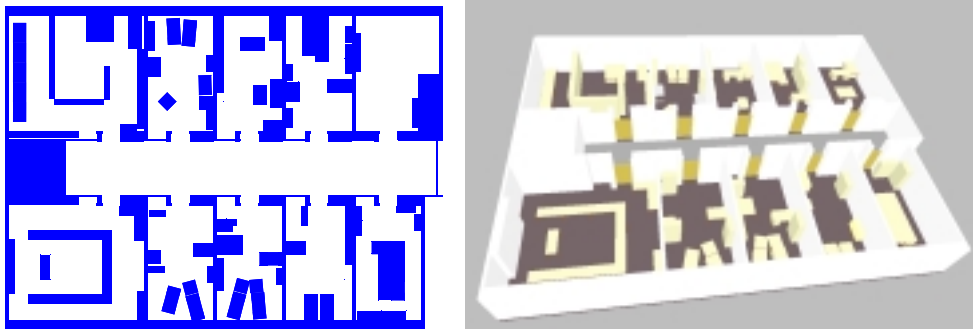


Abbildung 28: Geometrische 2D- und 3D-Karten einer Büroumgebung

Eine sehr populäre Repräsentationsform für autonome mobile Roboter stellen rasterbasierte oder Occupancy probability grid maps dar [ME85, Mor88]. Eine solche Occupancy probability grid map besteht aus einer Diskretisierung der Umgebung des Roboters in üblicherweise kleine Quadrate. Jede Zelle einer solchen Grid-Karte enthält die Wahrscheinlichkeit, daß die entsprechende Stelle der Umgebung durch ein Hindernis belegt ist. Gesucht ist demnach für jede Zelle  $\langle x, y \rangle$  die Wahrscheinlichkeit  $P(O_{x,y} | s_0, \dots, s_T)$ , wobei  $s_0, \dots, s_T$  den Observations des Roboters mit seinen Sensoren entspricht und die Variable  $O_{x,y}$  eine binäre Variable ist, die beschreibt, ob die Zelle  $\langle x, y \rangle$  belegt ist oder nicht. Da der Roboter mit seinen Sensoren permanent neue Informationen über die Umgebung aufnimmt, benötigt man nun eine Formel, die es erlaubt, diese Information in eine möglicherweise bestehende Karte zu integrieren. Unter verschiedenen Unabhängigkeitsannahmen läßt sich die folgende rekursive Update-Formel herleiten [Mor88, Thr98b]:

$$(3) \quad \frac{P(O_{x,y} | s_0, \dots, s_T)}{P(\neg O_{x,y} | s_0, \dots, s_T)} = \frac{P(O_{x,y} | s_T)}{P(\neg O_{x,y} | s_T)} \cdot \frac{P(O_{x,y})}{P(\neg O_{x,y})} \cdot \frac{P(O_{x,y} | s_0, \dots, s_{T-1})}{P(\neg O_{x,y} | s_0, \dots, s_{T-1})}$$

Der Quotient auf der linken Seite wird häufig auch bezeichnet als die Odds, daß die Zelle  $\langle x, y \rangle$  belegt ist [Mor88]. Nimmt man nun für die a priori Belegtheitswahrscheinlichkeit  $P(O_{x,y}) = 0.5$  an, so der mittlere Faktor vernachlässigt werden. Wegen  $P(X) = 1 - P(\neg X)$  kann man Gleichung 3 weiter umformen zu:

$$(4) \quad P(O_{x,y} | s_0, \dots, s_T) = \left( 1 + \frac{1 - P(O_{x,y} | s_T)}{P(O_{x,y} | s_T)} \cdot \frac{1 - P(O_{x,y} | s_0, \dots, s_{T-1})}{P(O_{x,y} | s_0, \dots, s_{T-1})} \right)^{-1}$$

Damit erhält man eine Update-Gleichung, für die lediglich der Term  $P(O_{x,y} | s_T)$  benötigt wird, d.h. die Wahrscheinlichkeit der Belegtheit einer Zelle gegeben eine einzelne Messung. Hierfür wurden in der Praxis unterschiedliche Verfahren angewandt. Während in [Mor88] ein probabilistisches Modell entwickelt wird, verwendet [Thr98b] Neuronale Netze. Abbildung 29 zeigt eine Occupancy-Grid-Map zu den in Abbildung 26 dargestellten Daten mit Hilfe der in [TBB<sup>+</sup>98, Thr98b] dargestellten Verfahren.

#### 0.4.2 Lokalisierung

Das Problem der Lokalisierung oder Positionsbestimmung stellt ein weiteres grundlegendes Problem der mobilen Robotik dar. Ohne Wissen über ihre Position können mobile Roboter typische Aufgaben wie z.B. Transportaufgaben oder die Überwachung von Gebäuden nicht effizient durchführen. Unter der Annahme, daß ein Roboter über ein Modell der Umgebung verfügt, läßt sich das Lokalisierungsproblem als die Aufgabe beschreiben, die wahrscheinlichste Position des Roboters gegeben alle Sensorinformationen und alle durchgeführten Navigationsoperationen zu berechnen (siehe auch Abbildung 30).



Abbildung 29: Occupancy-Grid-Karte zu der in Abbildung 28 dargestellten geometrischen Karte

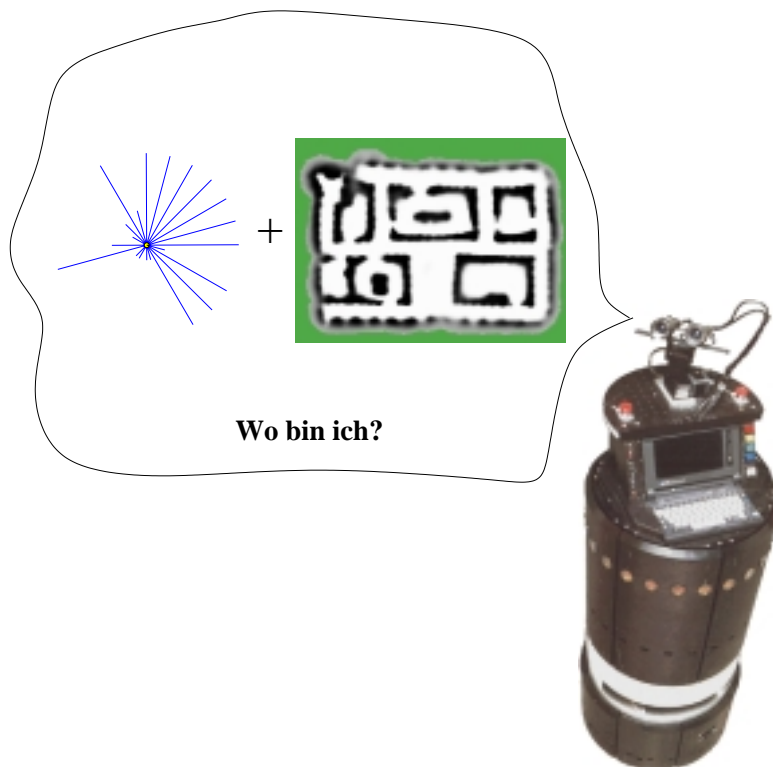


Abbildung 30: Das Problem der Positionsschätzung



Ein sehr allgemeiner Ansatz zur Schätzung der Position eines mobilen Roboters ist Markov Lokalisierung [FBT99]. Grundidee von Markov-Lokalisierung ist es, eine Wahrscheinlichkeitsverteilung  $P(L = l)$  über dem Zustandsraum des Roboters in seiner Umgebung schätzen. Hierbei ist der Zustandsraum üblicherweise durch alle möglichen Positionen und Orientierungen des Vehikels gegeben.

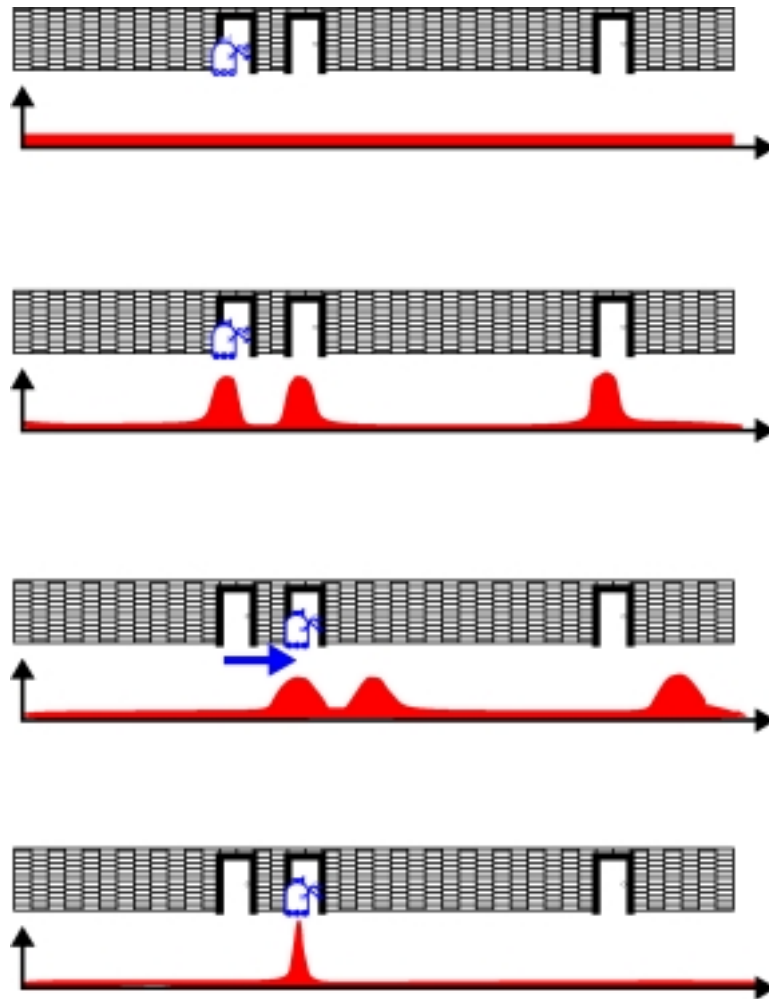


Abbildung 31: Grundidee der Markov-Lokalisierung: Globale Positionierung in einer eindimensionalen Welt

Bevor wir die mathematischen Grundlagen der Markov-Lokalisierung beschreiben, wollen wir zuvor kurz die Idee dieses Verfahrens an einem einfachen eindimensionalen Beispiel vorstellen. Hierzu betrachten wir die in Abbildung 31 dargestellte Umgebung und nehmen an, daß der Roboter sich nur entlang des Korridors bewegen und dabei nicht rotieren kann. Wir nehmen weiter an, daß der Roboter zu Beginn über keinerlei Vorwissen über seine Position verfügt. Markov Lokalisierung repräsentiert diesen Zustand der Unsicherheit durch eine Gleichverteilung, die jeder Position in der Umgebung die gleiche Wahrscheinlichkeit zuordnet. Nehmen wir jetzt an, daß der Roboter links von sich eine Tür detektiert. Markov-Lokalisierung ändert nun die Verteilung so, daß sich die Wahrscheinlichkeit erhöht, vor einer Tür zu stehen. Analog dazu werden die Wahrscheinlichkeiten, sich an anderen Stellen zu befinden, entsprechend verringert. Dies ist in der zweiten Zeile von Abbildung 31 dargestellt. Wenn der Roboter anschließend weiterfährt, verändert Markov-Lokalisierung wegen der möglichen Fehler bei der Ausführung dieser Aktion die Verteilung erneut. Dabei wird die Verteilung entsprechend der Aktion des Roboters verschoben. Gleichzeitig wird die gesamte Verteilung aber auch geglättet, um der möglichen Fehler bei der Ausführung der Aktion Rechnung zu tragen. Dies ist in der dritten Zeile von Abbildung 31 dargestellt. Wenn

wir nun annehmen, daß der Roboter wiederum eine Tür detektiert, und so wird die Wahrscheinlichkeit vor der mittleren Tür zu stehen erneut erhöht und an allen anderen Positionen weiter verringert. Damit ergibt sich eine Verteilung, die durch einen deutlichen Peak vor der mittleren Tür dominiert wird. Die Wahrscheinlichkeiten, sich an anderen Positionen zu befinden sind sehr gering, wenngleich sie nicht verschwinden.

Formal besteht das Problem der Positionsschätzung daraus, eine Wahrscheinlichkeitsverteilung  $P(L_t)$  über einer Zufallsvariable  $L_t$  zu schätzen. Die einzelnen Werte  $l$  der Variablen  $L_t$  sind die 3-Tupel der Form  $l = \langle x, y, \theta \rangle$ , wobei  $x$  und  $y$  die Position und  $\theta$  die Orientierung des Roboters angeben. Die Verteilung  $P(L_t)$  beschreibt somit für jeden möglichen Zustand des Roboters die Wahrscheinlichkeit, daß sich der Roboter genau in diesem Zustand befindet. Wie in dem Beispiel oben bereits angedeutet wird die Wahrscheinlichkeitsverteilung  $P(L_t)$  auf zwei verschiedene Arten aktualisiert, nämlich wann immer der Roboter Informationen über sein Umgebung aufnimmt bzw. wann immer er sich bewegt. Allgemein läßt sich das so formulieren, daß der Roboter eine Sequenz von Sensorinformationen erhält, die entweder die Umgebung wahrnehmen und durch  $s$  bezeichnet werden oder Aktionen beschreiben und durch  $a$  bezeichnet werden. Insgesamt sei der gesamte Datenstrom von Sensorinformationen durch  $d$  bezeichnet. Somit schätzt Markov-Lokalisierung eine Posterior-Verteilung über  $L_T$  bedingt über allen erhaltenen Information, d.h.

$$(5) \quad P(L_T = l | d) = P(L_T = l | d_0, \dots, d_T).$$

Wenn man verschiedene Unabhängigkeitsannahmen zuläßt wie z.B., daß aufeinanderfolgende Messungen voneinander unabhängig sind, sofern die Position des Roboters bekannt ist, kann man für die unterschiedliche Fälle, d.h. Perzeption und Aktion zwei verschiedene Update-Gleichungen herleiten [RN95]. Für den Fall einer Sensormessung, d.h.  $d_T = s_T$  verwenden wir die folgende rekursive, Bayes'sche Update-Gleichung:

$$(6) \quad P(L_T = l | d) = \alpha_T P(s_T | L_T = l) P(L_T = l | d_0, \dots, d_{T-1}).$$

Hierbei stellt der Term  $\alpha_T$  eine Normalisierungskonstante dar, die dafür sorgt, daß die Summe von  $P(L_T = l | d)$  über alle  $l$  eins ergibt [Pea88].

Nach einer Bewegung des Roboters aktualisieren wir  $P(L_T = l | d)$  auf der Basis einer aus dem Bereich der Markov-Ketten kommenden rekursiven Update-Gleichung:

$$(7) \quad P(L_T = l | d) = \int P(L_T = l | a_T, L_{T-1} = l') P(L_{T-1} = l' | d_0, \dots, d_{T-1}) dl'.$$

Dabei gehen wir davon aus, daß aufeinanderfolgende Aktionen des Roboters unabhängig voneinander sind [FBT99]. Beide Gleichungen zusammen beschreiben demnach ein rekursive Update-Schema zur Aktualisierung der Wahrscheinlichkeitsverteilung nach Aktionen und Perzeptionen.

Die interessanten Terme in den Gleichungen 6 und 7 sind  $P(s_T | L_T = l)$  und  $P(L_T = l | a_T, L_{T-1} = l')$ . Erstere entspricht im Prinzip einem Sensormodell und beschreibt die Wahrscheinlichkeit einer Messung  $s_T$  gegeben der Roboter befindet sich an Position  $l$ . Die konkrete Form dieser Verteilung hängt daher stark von den verwendeten Sensoren ab. Unterschiedliche Versionen solcher Sensormodelle findet man z.B. in [SK95, KCK96, NPB95, HK96, Kon99, FBT99, DBFT99]. In Gleichung 7 beschreibt  $P(L_T = l | a_T, L_{T-1} = l')$  das Aktionsmodell des Roboters, d.h. die Wahrscheinlichkeit, daß der Roboter, sofern er an Position  $l'$  startet und Aktion  $a_T$  ausführt an Position  $l$  landet. Häufig verwendet man zur Spezifikation dieses Aktionsmodells zwei unabhängige Normalverteilung für den Orientierungs- und Translationsfehler. Abbildung 32 zeigt typische resultierende Verteilungen über die Position des Roboters, gegeben der Roboter kennt seine Ausgangsposition und hat längere Strecken von über 20m zurückgelegt.

Markov Lokalisierung wurde in verschiedenen Varianten in der Vergangenheit erfolgreich angewendet. Die einzelnen Anwendungen unterscheiden sich im wesentlichen durch die Art der Repräsentation der Wahrscheinlichkeitsverteilung  $P(L_T)$ . Eine sehr populäre Technik ist die Verwendung von Kalman-Filtern [Kal60, May90, SSC90].  $P(L_T)$  wird in diesem Fall durch eine unimodale Gaußverteilung über dem Zustandsraum des Roboters repräsentiert. Diese Technik wird beispielsweise in [LDW91, Cox91, SC94, SGS92, GS96, LM94, AV98,

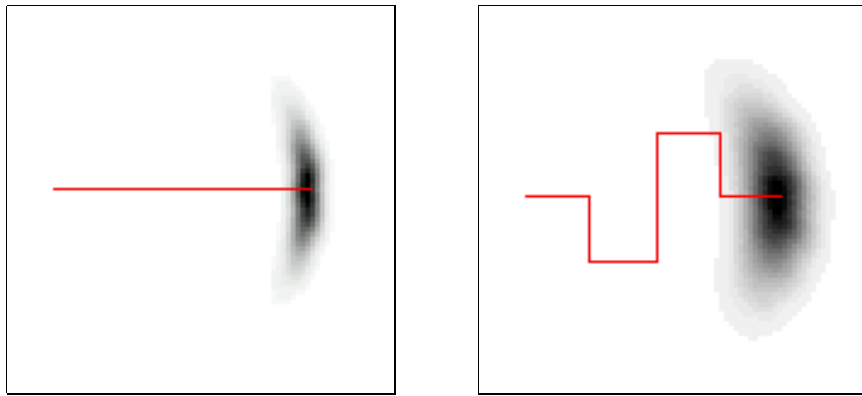


Abbildung 32: Typische Verteilung über die Roboterposition nach verschiedenen Aktionen des Roboters

EFL98, GWN99] verwendet. Alternativ zu Kalmanfiltern werden an verschiedenen Stellen auch stückweise konstante Funktionen mit unterschiedlichen Graden der Granularität verwendet [SK95, KCK96, NPB95, HK96, BDFC98, Kon99, FBT99]. Vorteile dieser Varianten ist, daß sie multimodale Verteilungen repräsentieren können. Sie sind somit in der Lage, den Roboter global zu lokalisieren, nachdem er z.B. eingeschaltet wurde, oder gar zu re-lokalisieren, falls ein Lokalisierungsfehler aufgetreten ist. Allerdings erkaufen sie sich diese Robustheit durch höheren Rechen- und Speicherplatzbedarf und auch auf Kosten der Genauigkeit [GBFK98]. Eine alternative Variante der Markov-Lokalisierung [DFBT99, FBDT99], die sowohl sehr genau als auch in Hinblick auf Speicherplatz und Rechenzeit sehr effizient ist, beruht auf den sogenannten Monte-Carlo-Filtern [Han70] bzw. dem im Computer-Vision-Bereich sehr erfolgreichen Condensation Algorithmus [IB98].

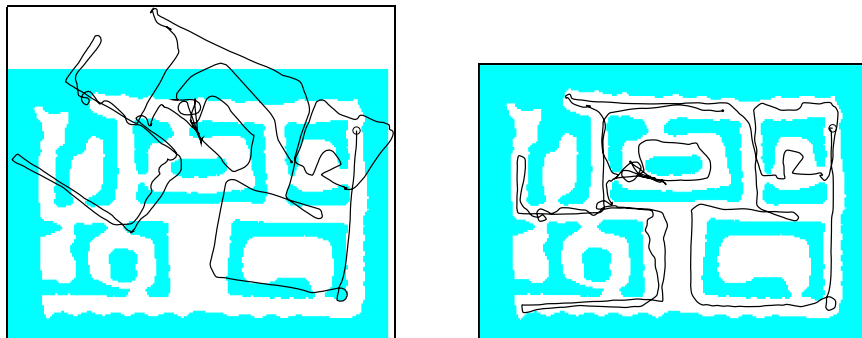


Abbildung 33: Anwendung der Markov-Lokalisierung zur Schätzung der Position eines Roboters in einer Grid-Karte und auf der Basis von Ultraschallsensoren

Die Anwendung und der Nutzen eines solchen Lokalisierungsverfahrens wird besonders durch die Abbildung 33 deutlich. Sie zeigt eine Karte der Umgebung der Mobile robot competition während der AAI 1994 [Sim95]. Diese Karte wurde mit den in [TBB<sup>+</sup>98] beschriebenen Verfahren generiert. Hierbei startet der Roboter in der rechten oberen Ecke der Karte. Auf der linken Seite zeigt Abbildung 33 den mit Hilfe der Odometriesensoren gemessenen Pfad 240m langen Pfad des Roboters. Die rechte Seite enthält den mit Hilfe der Markov-Lokalisierung auf der Basis der Ultraschallmessungen korrigierten Pfad des Roboters. Wie aus der Abbildung ersichtlich wird, wäre die Karte der Umgebung ohne das Lokalisierungsverfahren bereits nach wenigen Metern nutzlos. Am Ende der Trajektorie ist der Orientierungsfehler bereits auf über 40 Grad angewachsen. Die auf der rechten Seite gezeigte Trajektorie verdeutlicht hingegen, daß der Roboter aufgrund des Lokalisierungsverfahrens in der Lage ist, seine Position trotz relativ ungenauer Sensoren genau zu schätzen.

### 0.4.3 Pfadplanung

Die Pfadplanung ist eines der klassischen Probleme der (mobilen) Robotik. Dies wird z.B. dadurch deutlich, daß dem Pfadplanungsproblem ein komplettes Buch gewidmet ist [Lat91b]. Ziel der Pfadplanung ist die Berechnung eines möglichst kurzen Weges von der aktuellen Position des Roboters zu einem gegebenen Zielpunkt. In der Vergangenheit wurden zahlreiche Verfahren für die Berechnung solcher Trajektorien entwickelt. Eine häufige Voraussetzung dabei ist, daß die Umgebung vollständig bekannt ist und sich während der Fahrt des Systems auch nicht verändert. Techniken dieser Art sind beispielsweise Sichtbarkeitsgraphen oder Voronoi-Diagramme [Lat91b]. Ein Unterschied zwischen diesen beiden Techniken ist, daß Sichtbarkeitsgraphen die Weglänge minimieren, während Voronoi-Diagramme versuchen, den Abstand zu Hindernissen zu maximieren. Ein Verfahren für dynamische, d.h. sich evtl. verändernden Umgebungen, stellt die Potentialfeldmethode [Kha86] dar. Hierbei betrachtet man den Roboter als einen Massepunkt unter dem Einfluß eines künstlichen Potentialfeldes. Dieses Potentialfeld wiederum ist die Summe aus zwei verschiedenen Potentialfeldern  $U_r$  und  $U_a$ . Während  $U_r$  eine Kraft induziert, die den Roboter von Hindernissen abstößt, verursacht  $U_a$  eine Kraft, die den Roboter zum Zielpunkt hinzieht. Anders als bei Sichtbarkeitsgraphen oder Voronoi-Diagrammen versucht ein von dieser Methode gesteuerter Roboter stets gleichzeitig, seinen Abstand zu Hindernissen zu maximieren und dabei die Entfernung zum Ziel zu minimieren. Wenngleich ihre Beschreibung elegant ist, haben Potentialfeldmethoden verschiedene Nachteile. Beispielsweise müssen die Potentialfunktionen  $U_r$  und  $U_a$  sehr sorgfältig ausgewählt werden, um ein gutes Verhalten des Roboters zu erzeugen [Lat91b, KB91]. Das gravierendste Problem der Potentialfeldmethode entsteht durch lokale Minima in der Potentialfunktion, z.B. wenn der Roboter in ein U-förmiges Hindernis einfährt [Lat91b, KB91]. Gerät der Roboter in ein solches lokales Minimum, so bleibt er stecken und kann seinen Zielpunkt nicht mehr erreichen. Möglichkeiten, dieses Problem zu lösen, liegen in der Verwendung alternativer Potentialfunktionen, wie sie z.B. in [KC95] auch für bestimmte Verhalten des Roboters vorgestellt werden, oder in aufwendigeren Suchstrategien innerhalb des Potentialfeldes [Lat91b]. Eine speziell für Occupancy grid maps geeignetes Pfadplanungsproblem stellt die Value iteration [Bel57, How60, RN95, Sut92, KLM96] dar, ein Verfahren aus dem Bereich der Dynamischen Programmierung. Value iteration berechnet für jede Zelle einer Grid-Karte den jeweiligen Nutzen  $U(x, y)$  auf der Basis einer Kostenfunktion  $K(x, y)$ , die jeder Zelle  $\langle x, y \rangle$  Kosten für ihre Überquerung zuordnet. Bei Occupancy-grid-Karten wählt man diese Kosten üblicherweise als eine Funktion, die streng monoton steigend in der Belegtheitswahrscheinlichkeit  $P(O_{x,y})$  ist. Darüber hinaus berücksichtigt Value iteration auch mögliche Ausführungsfehler. Dies geschieht dadurch, daß man wie bei der Markov-Lokalisierung eine Transitionswahrscheinlichkeit  $P(\langle x', y' \rangle | a, \langle x, y \rangle)$  definiert, die festlegt, mit welcher Wahrscheinlichkeit der Roboter bei Ausführung einer Aktion  $a$  von Feld  $\langle x, y \rangle$  in das Feld  $\langle x', y' \rangle$  gelangt. Der Pfad mit den minimalen Kosten von der aktuellen Position des Roboters zum Zielpunkt wird dann auf der Basis der folgenden beiden Schritte bestimmt:

#### 1. Initialisierung:

$$(8) \quad U(x, y) \leftarrow \begin{cases} 0, & \text{falls } \langle x, y \rangle \text{ den Zielpunkt enthält} \\ \infty, & \text{sonst} \end{cases}$$

#### 2. Aktualisierung: Für jede Grid-Zelle $\langle x, y \rangle$ wiederhole:

$$(9) \quad U(x, y) \leftarrow \min_a \sum_{\langle x', y' \rangle} U(x', y') P(\langle x', y' \rangle | a, \langle x, y \rangle) + K(x, y)$$

bis  $U(x, y)$  konvergiert ist.

Nach der Terminierung kann dann die für jede Zelle optimal Aktion  $a^*(x, y)$  mit Hilfe der folgenden Formel bestimmt werden:

$$(10) \quad a^*(x, y) = \operatorname{argmin}_a \sum_{\langle x', y' \rangle} U(x', y') P(\langle x', y' \rangle | a, \langle x, y \rangle)$$

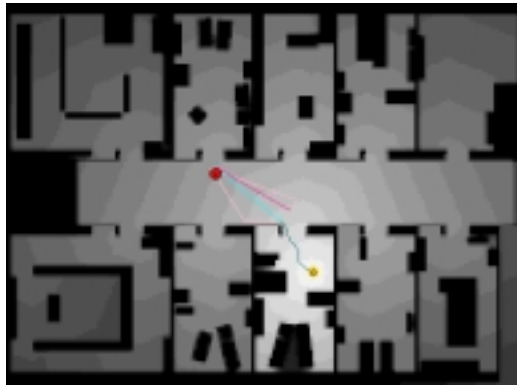


Abbildung 34: Typische, mit Value iteration berechnete Utility-Funktion bezüglich eines Zielpunktes.

Abbildung 34 zeigt eine mit diesem Verfahren erzeugte Utility-Funktion  $U(x, y)$  für den Fall von deterministischen Aktionen [TBB<sup>+</sup>98]. Der Vorteil dieses Verfahrens gegenüber den anderen bisher vorgestellten Methoden liegt in der expliziten Einbeziehung der Unsicherheit des Roboters bei der Ausführung von Aktionen. Darüber hinaus berechnet dieses Verfahren den optimalen Pfad von jedem Punkt aus zum Zielpunkt, so daß das System bei Abweichungen des Roboters vom ursprünglich festgelegten Pfad stets unmittelbar reagieren kann. Allerdings wird auch hier die Umgebung als statisch vorausgesetzt. Dieses Problem kann z.B. dadurch gelöst werden, daß man ein zusätzliches Modul verwendet, welches unmittelbar auf unvorhergesehene Hindernisse reagiert. Typische Techniken hierfür sind reaktive Kollisionsvermeidungen wie sie beispielsweise in [Sim96, FBT97] vorgestellt werden. Speziell diese beiden Techniken haben noch den zusätzlichen Vorteil, daß sie auch die Dynamik des Roboters berücksichtigen. Dabei werden nur solche Steuerungskommandos generiert, die ein sicheres Abbremsen vor dem nächsten Hindernis oder auch ein notwendiges Ausweichen zulassen. Aspekte dieser Art sind in den oben beschriebenen Pfadplanungsverfahren nicht berücksichtigt worden.

#### 0.4.4 Systeme und zukünftige Forschungen

In der jüngeren Vergangenheit gab es verschiedene Systeme, die mit den hier beschriebenen Techniken für die Navigation autonomer mobiler Roboter kontrolliert wurden. Als Beispiele hierfür sind unter anderem die Roboter Xavier [SGH<sup>+</sup>97] (Abbildung 35 links), Kurt [HK96] (Abbildung 35 rechts), Rhino [BCF<sup>+</sup>98, BCF<sup>+</sup>00] (Abbildung 36 links) und Minerva [TBB<sup>+</sup>99] (Abbildung 36 rechts) zu nennen. Während Xavier zur Navigation in einem Bürogebäude eingesetzt wird, soll Kurt Kanalsysteme autonom inspizieren. Eine typische Anwendung



Abbildung 35: Die mobilen Roboter Xavier (links) und Kurt (rechts).



Abbildung 36: Die Roboter Rhino (links) und Minerva (rechts) als interaktive Museumsführer.

von mobilen Robotern als interaktive und mobile Informationsterminals wurde mit den Systemen Rhino und Minerva realisiert. Beide Roboter wurden in den Jahren 1997 und 1998 im Deutschen Museum Bonn sowie im National Museum of American History in Washington DC über mehrere Tage als interaktive Museumsführer eingesetzt. Während ihres Einsatzes führten sie tausende von Besuchern durch die Ausstellung, erklärten Exponate über ihre Displays und durch die Ausgabe von Audiosequenzen und interagierten auf verschiedene Arten mit den Besuchern der Museen. Dabei waren sie gleichzeitig über das Internet ansteuerbar [SBF<sup>+</sup>00]. Abbildungen 37 zeigen typische Trajektorien der beiden Roboter in den Museen. Beide Systeme zeichneten sich durch eine sehr große Zuverlässigkeit aus. Beispielsweise hat Rhino 47 Stunden operiert und dabei eine Distanz von 18,6km zurückgelegt [BCF<sup>+</sup>00]. Nur in 6 von 2.400 Fällen hat er das angesteuerte Exponat nicht erreicht, was einer Erfolgsquote von 99.7% entspricht.

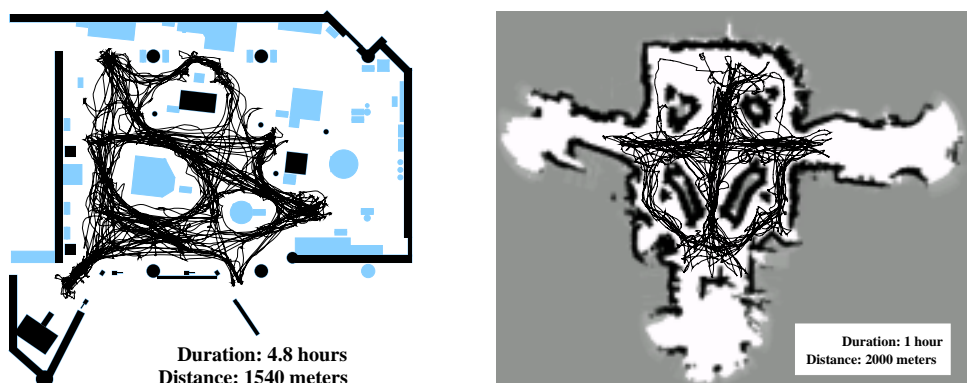


Abbildung 37: Typische Trajektorien der Roboter Rhino im Deutschen Museum Bonn (links) und Minerva in National Museum of American history in Washington DC (rechts).

Wenngleich sich die hier beschriebenen System durch eine große Zuverlässigkeit auszeichnen, bleibt dennoch eine Vielzahl von offenen Fragestellungen. Eine interessante Problematik stellt beispielsweise die Aktionsplanung für mobile Roboter und die Anbindung von Planungssystemen and Roboter dar. Vielversprechende Ansatzpunkte hierfür finden sich beispielsweise in [McD92, HV97, LRL<sup>+</sup>97, HBL98, Bee99]. Ein sehr allgemeines Verfahren zur Aktionsplanung unter Unsicherheit stellen die sogenannten Paritally observable Markov decision processes (POMDPs) dar [KLC95]. Aufgrund der explodierenden Suchräume sind diese Techniken bisher jedoch nur eingeschränkt nutzbar. Dennoch gibt es bereits Robotersysteme, die Techniken dieser Art nutzen, um Unsicherheiten bereits bei der Planung berücksichtigen zu können [KS98, KCK96, FBT98, RBFT99].

Interessante Forschungsansätze ergeben sich, wenn man von einzelnen Robotern zu Mehrrobotersystemen über-



Abbildung 38: Fußballspielende Roboter des CS-Freiburg Teams

geht. Unter anderem um die Forschungen auf dem Gebiet von Mehrrobotersystemen voranzutreiben wurde mit RobotSoccer eine internationale Initiative gestartet. Die erste Weltmeisterschaft fußballspielender Roboter wurde 1997 in Nagoya, Japan, während der IJCAI-97 ausgetragen. Deutsche Teams waren bisher sowohl in der Simulationsliga als auch in der Mid-size league sehr erfolgreich [GHH<sup>+</sup>99]. Hierbei müssen sowohl Probleme der koordinierten Navigation, koordinierten Zustandsschätzung als auch koordinierten Aktionsplanung effizient gelöst werden.

Schließlich ergeben sich auch stets neue Forschungsaufgaben durch den praktischen Einsatz von Systemen. Gerade durch den Schritt heraus aus typischen Laborumgebungen in reale Welt ergeben sich häufig völlig neue Problemstellungen. Ein weiteres interessantes Problem ist die Gestaltung von Benutzerschnittstellen für autonome mobile Systeme, die es ungeübten Benutzern erlauben, auf einfache Weise mit den Systemen zu interagieren. Techniken, wie sie z.B. in [BCF<sup>+</sup>00, TBB<sup>+</sup>99, HSH<sup>+</sup>97] beschrieben werden, stellen erste Schritte in diese Richtung dar.

## Literatur

- [AAT99] S. Abrams, P. Allen, and K. Tarabanis. Computing camera viewpoints in an active robot work cell. *Int. J. of Robotics Research*, 18(3):267 – 285, 1999.
- [A.B91] R. A. Brooks. Intelligence without representation. *Artificial Intelligence Journal*, 47:139–159, 1991.
- [AB98] R.C. Arkin and T. Balch. Cooperative multiagent robotic systems. In Kortenkamp et al. [KBM98].
- [Ada98] M. Adams. *Sensor Modelling, Design and Data Processing for Autonomous Navigation*. Number 13 in World Scientific Series in Robotics and Intelligent Systems. World Scientific, 1998.
- [AF88] N. Ayache and O. Faugeras. Building, registering and fusing noisy visual maps. *Int. J. of Robotics Research*, 7(6), 1988.
- [AG92] M. Abidi and R. Gonzalez, editors. *Data Fusion in Robotics and Machine Intelligence*. Academic Press, 1992.
- [Ang97] J. Angeles. *Fundamentals of Robotic Mechanical Systems : Theory, Methods, and Algorithms*. Springer-Verlag, 1997.
- [Ast65] K. Astrom. Optimal control of Markov decision processes with incomplete state estimation. *J. Math. Analysis and Appl.*, 10, 1965.
- [ATT99] N. Abe, K. Tanaka, and H. Taki. Understanding of mechanical assembly instruction manual by integrating vision and language processing and simulation. In *Proc. 1999 IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, 1999. IEEE-Press.
- [AV98] K.O. Arras and S.J. Vestli. Hybrid, high-precision localization for the mail distributing mobile robot system MOPS. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1998.



- [Aya88] N. Ayache. *Construction et fusion de représentations visuelles tridimensionnelles – applications à la robotique mobile*. PhD thesis, INRIA, 1988.
- [BA99] S. Botelho and R. Alami. M+: A scheme for multi-robot cooperation through negotiated task allocation and achievement. In *Proc. 1999 IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, 1999. IEEE-Press.
- [Bak88] A. Baker. Complete manufacturing control using a contract net: A simulation study. In *Proc. Int. Conf. on Comp. Int. Manufacturing*, pages 100 – 109, 1988.
- [BB97] C. Boutilier and R. Brafman. Planning with concurrent interacting actions. In *Proc. 14th Nat. Conf. on Artif. Intell. AAAI-97*. AAAI Press / The MIT Press, 1997.
- [BBW<sup>+</sup>97] T. Boulton, R. Blum, R. Wallace, G. Zhang, S. Nayar, P. Allen, and J. Kender. Advanced visual sensor systems. In T. Strat, editor, *Proc. of the 1997 Image Understanding Workshop*, New Orleans, 1997. Morgan Kaufmann.
- [BCF<sup>+</sup>98] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The interactive museum tour-guide robot. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1998.
- [BCF<sup>+</sup>00] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1-2), 2000. To appear.
- [BCG<sup>+</sup>99] A. Bredenfeld, T. Christaller, W. Göhring, H. Günther, H. Jaeger, H. Kobialka, P. Plöger, and P. Schöll. Behavior engineering with “dual dynamics” models and design tools. In *Proc. 19th Joint Conf. Artif. Intell. IJCAI-99*. Morgan Kaufmann, 1999.
- [BDFC98] W. Burgard, A. Derr, D. Fox, and A.B. Cremers. Integrating global position estimation and position tracking for mobile robots: the Dynamic Markov Localization approach. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1998.
- [BDH99] C. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *J. Art. Intell. Res.*, 11:1 – 93, 1999.
- [Bee99] M. Beetz. Structured reactive controllers — a computational model of everyday activity. In *Proceedings of the Third International Conference on Autonomous Agents*, 1999.
- [BEF96] J. Borenstein, B. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques*. A. K. Peters, Ltd., Wellesley, MA, 1996.
- [Bel57] R.E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.
- [Ber85] J. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer Series in Statistics. Springer-Verlag, 1985.
- [BFH99] C. Borst, M. Fischer, and G. Hirzinger. A fast and robust grasp planner for arbitrary 3D objects. In *Proc. 1999 IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, 1999. IEEE-Press.
- [BI97] R. Brooks and S. Iyengar. *Multi-Sensor Fusion: Fundamentals and Applications with Software*. Prentice-Hall, 1997.
- [BKM<sup>+</sup>98] M. Becker, E. Kefalea, E. Makl, C. v. d. Malsburg, M. Pagel, J. Triesch, J. C. Vorbrüggen, and S. Zadel. Gripsee: A robot for visually-guided grasping. In *Proceedings of the Int. Conf. Artif. Neural Netw. 1998*, Sweden, 1998.
- [BLG97] B. Bonet, G. Loerincs, and H. Geffner. A robust and fast action selection mechanism for planning. In *Proc. 14th Nat. Conf. on Artif. Intell. AAAI-97*. AAAI Press / The MIT Press, 1997.
- [BM98] I. Bloch and H. Maître. Fusion of image information under imprecision. In B. Bouchon-Meunier, editor, *Aggregation and Fusion of imperfect information*, Studies in Fuzziness and Soft Computing. Physica-Verlag, 1998.
- [Bog87] P. Bogler. Shafer-Dempster reasoning with applications to multisensor target identification systems. *IEEE Trans. Systems, Man, and Cybernetics*, 17, 1987.
- [Bor96] J. Borenstein. *Navigating Mobile Robots*. A. K. Peters Ltd., 1996.

- [BP97] S. Baluja and D. Pomerleau. Dynamic relevance: Vision-based focus of attention using artificial neural networks. *Artificial Intelligence*, pages 381 – 395, 1997.
- [Bro] Ronald A. Brooks. A robot that walks: Emergent behaviors from a carefully evolved network. *Neural Computation*, 1:153.
- [Bro86a] R. Brooks. Achieving artificial intelligence through building robots. Technical Report AI memo 899, MIT, May 1986.
- [Bro86b] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, RA-2:14–23, April 1986.
- [Bro90] Rodney A. Brooks. The behavior language. users’s guide. Technical report, MIT AI laboratory, 1990.
- [Bro91] R. A. Brooks. Intelligence without reason. *International Joint Conference on Artificial Intelligence*, pages 569–595, 1991.
- [Bro95] R. A. Brooks. Intelligence without reason. In Luc Steels and Rodney Brooks, editors, *The artificial life route to artificial intelligence: building embodied, situated agents*, pages 25–81, Hillsdale, New Jersey, 1995. Lawrence Erlbaum Associates.
- [BS94] Rodney A. Brooks and Lynn A. Stein. Building brains for bodies. *Autonomous Robots*, 1(1):7–25, 1994. nachgedruckt in pfeifer+95 monte verita proceedings.
- [BSF88] Y. Bar-Shalom and T. Fortmann. *Tracking and Data Association*. Academic Press, 1988.
- [BU97] I. Bratko and T. Urbančič. Transfer of control skill by machine learning. *Eng. Appl. of Artif. Intell.*, 10(1):63 – 71, 1997.
- [Bur96] E. Burdet. *Algorithms of Human Motor Control and their Implementation in Robotics*. PhD thesis, ETH-Zuerich, 1996.
- [Bur98] Hans-Dieter Burkhard. Einführung in die Agenten-Technologie. *Informationstechnik und Technische Informatik*, (4):6–11, August 1998.
- [CA99] C. Colombo and B. Allotta. Image-based robot task planning and control using a compact visual representation. *IEEE Trans. on Systems, Man, and Cybernetics (A)*, 29(1), 1999.
- [Can88] J. Canny. *The complexity of robot motion planning*. MIT-Press, 1988.
- [CGS97] V. Cantoni, V. Di Gesu, and A. Setti, editors. *Human and Machine Perception: Information Fusion*. Plenum Press, 1997.
- [Chr99] Thomas Christaller. Cognitive robotics: A new approach to artificial intelligence. *Artificial Life and Robotics*, (3), 1999. <http://ais.gmd.de/people/Thomas.Christaller/>.
- [CKB96] H. Choset, I. Konuksven, and J.W. Burdick. Sensor Based Planning for a Planar Rod Robot. In *Proc. IEEE/SICE/RSJ Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, Washington, DC, 1996.
- [Cla97] Andy Clark. *Being There. Putting brain, body, and the world together again*. MIT Press, 1997.
- [CLZ96] K. Chang, J. Liu, and J. Zhou. Bayesian probabilistic inference for target recognition. In *Signal Processing, Sensor Fusion, and Target Recognition V*, volume 2755 of *SPIE Proceedings*. SPIE – Society of Photo-Optical Instrumentation Engineers, 1996.
- [CM99a] T. Celinski and B. McCarrager. Achieving efficient data fusion through integration of sensory perception control and sensor fusion. In *Proc. 1999 IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, 1999. IEEE-Press.
- [CM99b] T. Celinski and B. McCarrager. Improving sensory perception through predictive correction of monitoring errors. In *Proc. 1999 IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, 1999. IEEE-Press.
- [Cox91] I.J. Cox. Blanche—an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation*, 7(2):193–204, 1991.
- [Cra89] J. Craig. *Introduction to Robotics: Mechanics and Control by John J. Craig*. Addison-Wesley, 1989.

- [CSS94] William J. Clancey, Stephen W. Smoliar, and Mark J. Stefik, editors. *Contemplating minds. A forum for artificial intelligence*. Artificial Intelligence. MIT Press, Cambridge (Mass), 1994. buchbesprechungen aus dem AI journal. und reaktionen der jeweiligen autoren.
- [CW90] I. Cox and G. Wilfong. *Autonomous Robot Vehicles*. Springer-Verlag, 1990.
- [CW93] C. Chun and W. Wolfe. *Mobile Robots*. SPIE, 1993.
- [CY90] J. Clark and A. Yuille. *Data Fusion for Sensory Information Processing Systems*. Kluwer Academic, 1990.
- [Das93] B. Dasarathy. *Decision Fusion*. IEEE Press, 1993.
- [Dau95] Kerstin Dautenhahn. Getting to know each other—artificial social intelligence for autonomous robots. *Robotics and Autonomous Systems*, 16:333–356, 1995.
- [DBFT99] F. Dellaert, W. Burgard, D. Fox, and S. Thrun. Using the condensation algorithm for robust, vision-based mobile robot localization. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999.
- [DC97] Kerstin Dautenhahn and Thomas Christaller. Remembering, rehearsal and empathy - towards a social and embodied cognitive psychology for artifacts. In Sean O’Nuallian, Paul Mc Kevitt, and Eoghan Mac Aogain, editors, *Two science of the mind. Readings in cognitive science and consciousness*, Advances in Consciousness Research, pages 257–282. John Benjamins, Amsterdam, 1997.
- [DFBT99] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo localization for mobile robots. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1999.
- [DH98] M. Dekhil and T. C. Henderson. Instrumented sensor system architecture. *International Journal of Robotics Research*, 17(4):402–417, April 1998.
- [DHW93] D. Draper, S. Hanks, and D. Weld. Probabilistic planning with information gathering and contingent execution. Technical Report 93-12-04, University of Washington Department of Computer Science and Engineering, 1993.
- [DL95] T. Dean and S. Lin. Decomposition techniques for planning in stochastic domains. In *Proc. 14th Joint Conf. Artif. Intell. IJCAI-95*, pages 1121 – 1127. Morgan Kaufmann, 1995.
- [DT99] B. Dasarathy and S. Townsend. FUSE – fusion utility sequence estimator. In *Proc. 2nd Int. Conf. on Information Fusion*, Sunnyvale, CA, 1999. Int. Society of Information Fusion.
- [DW87] H. Durrant-Whyte. Sensor models and multi-sensor integration. *Int. J. Robotics Research*, 6(3), 1987.
- [DW88] Hugh F. Durrant-Whyte. *Integration, coordination and control of multi-sensor robot systems*. Kluwer Academic Publishers, 1988.
- [DWJM98] C. Dorai, G. Wang, A. Jain, and C. Mercer. Registration and integration of multiple object views for 3D model construction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(1), 1998.
- [EFL98] H. Endres, W. Feiten, and G. Lawitzky. Field test of a navigation system: Autonomous cleaning in supermarkets. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1998.
- [EHA93] S. P. Engelson, G. D. Hager, and A. Atiya. On comparing statistical and set-based methods in sensor data fusion. In *Proc. Int. Conf. on Robotics and Automation*, 1993.
- [EHW<sup>+</sup>92] O. Etzioni, S. Hanks, D. Weld, D. Draper, N. Lesh, and M. Williamson. An approach to planning with incomplete information. In *Proc. 3rd Int. Conf. on Principles of Knowledge Representation and Reasoning KR-1992*, pages 115 – 125. Morgan Kaufmann, 1992.
- [Eng89] J. Engelberger. *Robotics in Service*. MIT-Press, 1989.
- [Erd95] D. Erdmann. Understanding action and sensing by designing action-based sensors. *Int. Journal of Robotics Research*, 14(5), 1995.
- [Eve95] H.R. Everett. *Sensors for Mobile Robots - Theory & Applications*. AK Peters, 1995.
- [FBDT99] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte Carlo localization: Efficient position estimation for mobile robots. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1999.

- [FBT97] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1), March 1997.
- [FBT98] D. Fox, W. Burgard, and S. Thrun. Active Markov localization for mobile robots. *Robotics and Autonomous Systems*, 25:195–207, 1998.
- [FBT99] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11, 1999.
- [FGL87] K. Fu, R. Gonzalez, and C. Lee. *Robotics – Control, Sensing, Vision and Intelligence*. McGraw-Hill, 1987.
- [FH86] O. Faugeras and M. Hebert. The representation, recognition, and locating of 3-d objects. *Int. J. of Robotics Research*, 5(3), 1986.
- [FN71] R. Fikes and N. Nilsson. STRIPS: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3/4), 1971.
- [Fox81] M. Fox. An organizational view of distributed systems. *IEEE Trans. on Systems, Man and Cybernetics*, SMC-11, 1981.
- [FPWK99] J. Fraile, C. Paredis, C. Wang, and P. Khosla. Agent-based planning and control of a multi-manipulator assembly system. In *Proc. 1999 IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, 1999. IEEE-Press.
- [Fuj91] K. Fujimura. *Motion Planning in Dynamic environments*. Springer-Verlag, 1991.
- [Ful98] J. Fuller. *Robotics: Introduction, Programming, and Projects*. Prentice Hall, 1998.
- [FZK99] M. Ferch, J. Zhang, and A. Knoll. Robot skill transfer based on B-spline fuzzy controller for force-control tasks. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1999.
- [GBFK98] J.-S. Gutmann, W. Burgard, D. Fox, and K. Konolige. An experimental comparison of localization methods. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1998.
- [GHH<sup>+</sup>99] J.-S. Gutmann, W. Hatzack, I. Herrmann, B. Nebel, F. Rittinger, A. Topor, T. Weigel, and B. Welsch. The CS Freiburg Robotic Soccer Team: Reliable self-localization, multirobot sensor integration, and basic soccer skills. In M. Asada, editor, *RoboCup-98: Robot Soccer World Cup II*. Springer-Verlag, 1999.
- [GI94] K. Gremban and K. Ikeuchi. Planning multiple observations for object recognition. *Int. J. Comp. Vision*, 12(2):137 – 172, 1994.
- [GJ94] C. Giraud and B. Jouvencel. Sensor selection in a fusion process: a fuzzy approach. In *First IEEE Conf. on Multisensor Fusion and Integration for Intelligent Systems, MFI-94*. IEEE Press, 1994.
- [GMC99] X. Gu, M. Marefat, and F. Ciarallo. A robust approach for sensor placement in automated vision dimensional inspection. In *Proc. 1999 IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, 1999. IEEE-Press.
- [GMKB97] R. Goldman, D. Musliner, K. Krebsbach, and M. Boddy. Dynamic abstraction planning. In *Proc. 14th Nat. Conf. on Artif. Intell. AAAI-97*. AAAI Press / The MIT Press, 1997.
- [GMN97] I. Goodman, R. Mahler, and H. Nguyen. Mathematics of data fusion. In *Theory and Decision Library. Series B, Mathematical and Statistical Methods*, volume 37. Kluwer Academic, 1997.
- [GS96] J.-S. Gutmann and C. Schlegel. AMOS: Comparison of scan matching approaches for self-localization in indoor environments. In *Proc. of the 1st Euromicro Workshop on Advanced Mobile Robots*. IEEE Computer Society Press, 1996.
- [GWN99] J.-S. Gutmann, T. Weigel, and B. Nebel. Fast, accurate, and robust self-localization in polygonal environments. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1999.
- [Hag95] G. D. Hager. The X-vision system: A general purpose substrate for real-time vision-based robotics. *Workshop on Vision for Robots*, pages 56–63, 1995.
- [Hal92] D. Hall. *Mathematical Techniques in Multisensor Data Fusion*. Artech House, 1992.

- [Han70] J.E. Handschin. Monte Carlo techniques for prediction and filtering of non-linear stochastic processes. *Automatica*, 6, 1970.
- [HBL98] D. Hähnel, W. Burgard, and G. Lakemeyer. GOLEX — bridging the gap between logic (GOLOG) and a real robot. In *Proc. of the 22nd German Conference on Artificial Intelligence (KI'98)*, LNCS. Springer Verlag, 1998.
- [HE99] B. Hildebrandt and H.J. Eikmeyer. Effizientes sprachverstehen mit combinatory categorial grammar. Technical report, SFB360, Univ. Bielefeld, 1999.
- [HK89] S. Hutchinson and A. Kak. Planning sensing strategies in a robot work cell with multi-sensor capabilities. *IEEE Trans. on Robotics and Automation*, 5(6), 1989.
- [HK96] J. Hertzberg and F. Kirchner. Landmark-based autonomous navigation in sewerage pipes. In *Proc. of the First Euromicro Workshop on Advanced Mobile Robots*. IEEE Computer Society Press, 1996.
- [HLSvS98] U. Handmann, G. Lorenz, T. Schnitger, and W. v. Seelen. Fusion of different sensors and algorithms for segmentation. In *Proceedings of the IEEE Conference on Intelligent Vehicles*, 1998.
- [HM91] G. Hager and M. Mintz. Task-directed sensor data fusion and sensor planning. *Int. J. of Robotics Research*, 10(4), 1991.
- [Ho92] Y. Ho, editor. *Discrete Event Dynamic Systems*. IEEE Press, 1992.
- [Hol90] J. Hollerbach. Planning of arm movements. In D. Osherson, S. Kosslyn, and J. Hollerbach, editors, *Visual Cognition and Action*, volume 2. MIT Press, 1990.
- [How60] R. A. Howard. *Dynamic Programming and Markov Processes*. MIT Press and Wiley, 1960.
- [HR97] B. Hildebrandt and G. Rickheit. Verarbeitung von präpositionalphrasen in der combinatory categorial grammar. Technical Report 97/6, SFB360, Univ. Bielefeld, 1997.
- [HSFS99] A. Hauck, M. Sorg, G. Färber, and T. Schenk. What can be learned from human reach-to-grasp movements for the design of robotic hand-eye systems. In *Proc. 1999 IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, 1999. IEEE-Press.
- [HSH<sup>+</sup>97] A. Hideki, H. Satoru, I. Hara, M. Yoichi, A. Shotaro, and Toshihiro M. Socially embedded learning of the office-conversant mobile robot, Jijo-2. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1997.
- [HV97] K.Z. Haigh and M.M. Veloso. High-level planning and low-level execution: Towards a complete robotic agent. In *Proc. of the International Conference on Autonomous Agents (AA)*, 1997.
- [IA97] M. Irani and P. Anandan. Robust multi-sensor image alignment. In T. Strat, editor, *Proc. of the 1997 Image Understanding Workshop*, New Orleans, 1997. Morgan Kaufmann.
- [IB98] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.
- [IS94] K. Ikeuchi and T. Suehiro. Toward an assembly plan from observation. Part I: Task recognition with polyhedral objects. *IEEE Trans. on Robotics and Automation*, 10(3), 1994.
- [ISK92] S. Iyengar, M. Sharma, and R. Kashyap. Information routing and reliability issues in distributed sensor networks. *IEEE Trans. on Signal Processing*, 40(12), December 1992.
- [IV97] K. Ikeuchi and M. Veloso, editors. *Symbolic Visual Learning*. Oxford University Press, 1997.
- [Jae96] Herbert Jaeger. The dual dynamics design scheme for behaviour-based robots: A tutorial. Arbeitspapiere der GMD 966, German National Research Institute for Information Technology (GMD), St. Augustin, 1 1996.
- [JS99] R. Joshi and A. Sanderson. Minimal representation multisensor fusion using differential evolution. *IEEE Trans. on Systems, Man, and Cybernetics (A)*, 29(1), 1999.
- [JWC98] R. Jones, R. Wilson, and T. Calton. On constraints in assembly planning. *IEEE Trans. on Robotics and Automation*, 14(46):849 – 863, 1998.

- [KAF99] F. Kobayashi, F. Arai, and T. Fukuda. Sensor selection by reliability based on possibility measure. In *Proc. 1999 IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, 1999. IEEE-Press.
- [Kai97] M. Kaiser. Transfer of elementary skills via human-robot interaction. *Adaptive Behavior*, 5(3/4):249–280, 1997.
- [Kal60] R.E. Kalman. A new approach to linear filtering and prediction problems. *Trans. of the ASME, Journal of basic engineering*, 82:35–45, March 1960.
- [KB91] Y. Koren and J. Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, April 1991.
- [KBM98] D. Kortenkamp, R.P. Bonasso, and R. Murphy, editors. *Artificial Intelligence and Mobile Robots*. MIT/AAAI Press, Cambridge, MA, 1998.
- [KC95] M. Khatib and R. Chatila. An extended potential field approach for mobile robot sensor-based motions. In *Proc. International Conference on Intelligent Autonomous Systems (IAS'4)*, 1995.
- [KCK96] L.P. Kaelbling, A.R. Cassandra, and J.A. Kurien. Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1996.
- [Kha86] O. Khatib. Real-time obstacle avoidance for robot manipulator and mobile robots. *The International Journal of Robotics Research*, 5(1), 1986.
- [KI95] S. Kang and K. Ikeuchi. Toward automatic robot instruction from perception – temporal segmentation of tasks from human hand motion. *IEEE Trans. on Robotics and Automation*, 11(5), 1995.
- [KII92] Y. Kuniyoshi, M. Inaba, and H. Inoue. Seeing, understanding and doing human task. In *Proc. 1992 IEEE Int. Conf. on Robotics and Automation*, Nice, 1992. IEEE-Press.
- [KLC95] L.P. Kaelbling, M.L. Littman, and A.R. Cassandra. Planning and acting in partially observable stochastic domains. Technical report, Brown University, 1995.
- [KLM96] L.P. Kaelbling, M.L. Littman, and A.W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4, 1996.
- [KM] K. Konolige and K. Myers. The saphira architecture for autonomous mobile robots. to appear in a forthcoming book at MIT press, <http://www.ai.sri.com/konolige/saphira/arch.pdf>.
- [KMM96] G. Kamberova, R. Mandelbaum, and M. Mintz. Statistical decision theory for mobile robotics: theory and applications. In *Proc. 1996 IEEE Int. Conf. Multisensor Fusion and Integration for Intelligent Systems (MFI 96)*, Washinton, D.C., Dec. 1996.
- [Kni91] T. Knieriem. *Autonome Mobile Roboter*. BI-Sissenschaftsverlag, 1991.
- [Kon99] K. Konolige. Markov localization using correlation. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1999.
- [Kor98] D. Kortenkamp. *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*. MIT-Press, 1998.
- [KQW97] O. Khatib, S. Quinlan, and D. Williams. Robot planning and control. *Robotics and Autonomous Systems*, 21:249 – 261, 1997.
- [KS98] S. Koenig and R. Simmons. A robot navigation architecture based on partially observable Markov decision process models. In Kortenkamp et al. [KBM98].
- [Lat91a] J. Latombe. *Robot Motion Planning*. Kluwer Academic, 1991.
- [Lat91b] J. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA, 1991. ISBN 0-7923-9206-X.
- [LB91] E. Lebsaft and F. Braun. Ein Expertensystem zur Auftragsreihenfolgeplanung für eine roboterisierte Packstraße in der Lebensmittelindustrie. In H. Bullinger, editor, *Expertensysteme in Produktion und Engineering*. Springer-Verlag, 1991.

- [LC97] D. Leng and M. Chen. Robot trajectory planning using simulation. *Robotics & Computer-Integrated Manufacturing*, 13(2):121 – 129, 1997.
- [LDW91] J.J. Leonard and H.F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, 1991.
- [Lee90] M. Lee. *Intelligente Roboter*. VCH Verlag, 1990.
- [LKW97] D. Leake, A. Kinley, and D. Wilson. Case-based similarity assessment: Estimating adaptability from experience. In *Proc. 14th Nat. Conf. on Artif. Intell. AAAI-97*. AAAI Press / The MIT Press, 1997.
- [LL98] F. Lin and H. Levesque. What robots can do: robot programs and effective achieveability. *Artificial Intelligence*, 101:201 – 206, 1998.
- [Llo98] J. Lloyd. Removing the singularities of serial manipulators by transforming the workspace. In *Proc. 1998 IEEE Int. Conf. on Robotics and Automation*, Leuven, Belgium, 1998. IEEE-Press.
- [LM94] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2D range scans. In *IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, 1994.
- [LR95] M. Lee and J. Rowland. *Intelligent Assembly Systems*. World Scientific, 1995.
- [LRL<sup>+</sup>97] H.J. Levesque, R. Reiter, Y. Lespérance, F. Lin, and R. Scherl. GOLOG: A logic programming language for dynamic domains. *Journal of Logic Programming*, 31:59–84, 1997.
- [LWG97] M. Leventon, W. Wells, and W. Grimson. Multiple 2D-3D mutual information registration. In T. Strat, editor, *Proc. of the 1997 Image Understanding Workshop*, New Orleans, 1997. Morgan Kaufmann.
- [Mae90] P. Maes. Situated agents can have goals. *Robotics and Autonomous Systems*, 6(1):49 – 70, 1990.
- [Man99] G. Mantriota. Optimal grip points for contact stability. *Int. J. of Robotics Research*, 18(5):502 – 513, 1999.
- [Mat95] Maja J Mataric. Issues and approaches in the design of collective autonomous agents. *Robotics and Autonomous Systems*, 16((2-4)):321–331, Dec 1995.
- [May90] P.S. Maybeck. The Kalman filter: An introduction to concepts. In I.J. Cox and G.T. Wilfong, editors, *Autonomous Robot Vehicles*. Springer Verlag, 1990.
- [MB93] David McFarland and Thomas Bösser. *Intelligent behavior in animals and robots*. MIT Press, 1993.
- [MBF98] G. Mauris, E. Benoit, and L. Foulloy. Fuzzy linguistic methods for aggregation of complementary sensor information. In B. Bouchon-Meunier, editor, *Aggregation and Fusion of Imperfect Information*, Studies in Fuzziness and Soft Computing. Physica-Verlag, 1998.
- [McD92] D. McDermott. Transformational planning of reactive behavior. Research Report YALEU/DCS/RR-941, Yale University, 1992.
- [MDW95] J. Manyika and H. Durrant-Whyte. *Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach*. Ellis Horwood, 1995.
- [ME85] H.P. Moravec and A.E. Elfes. High resolution maps from wide angle sonar. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1985.
- [MH98] E. Marchand and G. Hager. Dynamic sensor planning in visual servoing. In *Proc. 1998 IEEE Int. Conf. on Robotics and Automation*, Leuven, Belgium, 1998. IEEE-Press.
- [MI98] J. Miura and K. Ikeuchi. Task-oriented generation of visual sensing strategies in assembly tasks. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(2), 1998.
- [Mir97] B. Mirtich. *V-Clip: Fast and Robust Polyhedral Collision Detection*. TR 97-05, Mitsubishi Electric, Cambridge, MA, June 1997. <http://www.merl.com/reports/TR97-05/index.html>.
- [MK92] J. Meinkoehn and A. Knoll. A model of non-hierarchical control in distributed sensor-networks. In *Proceedings of SPIE/OE Technology 1992, Sensor Fusion V*, volume 1828, Boston, November 1992. The International Society for Optical Engineering.
- [MLRPW98] Ralf Möller, Dimitrios Lambrinos, Thomas Labhart Rolf Pfeifer, and Rüdiger Wehner. Modeling ant navigation with an autonomous agent. In *From Animals to Animats 5 (SAB98)*, Proc. Fifth International Conference of the Society of Adaptive Behavior, pages 185–194. MIT Press, 1998.



- [MLS94] R. Murray, Z. Li, and S. Sastry. *Robotic Manipulation*. CRC Press, 1194.
- [Mor88] H.P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, Summer 1988.
- [Mor94] S. Morriss. *Automated Manufacturing Systems : Actuators, Controls, Sensors, and Robotics*. Glencoe McGraw Hill, 1994.
- [MR91] D. McAllester and D. Rosenblitt. Systematic nonlinear planning. In *Proc. 9th Nat. Conf. on Artif. Intell. AAAI-91*, pages 634 – 639. AAAI Press, 1991.
- [MS97] J. Miura and Y. Shirai. Vision-motion planning of a mobile robot considering vision uncertainty and planning cost. In *Proc. 15th Joint Conf. Artif. Intell. IJCAI-97*, pages 1194 – 1200. Morgan Kaufmann, 1997.
- [MTF96] J. Masterson, R. Towers, and S. Fardo. *Robotics Technology*. Goodheart-Willcox, 1996.
- [MTYN99] M. Mitsuishi, K. Tanaka, Y. Yokokohji, and T. Nagao. Remote rapid manufacturing with “action media” as an advanced user interface. In *Proc. 1999 IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, 1999. IEEE-Press.
- [Mur98] R. Murphy. Dempster-shafer theory for sensor fusion in autonomous mobile robots. *IEEE Trans. on Robotics and Automation*, 14(2), 1998.
- [Mut98] A. Mutambara. *Decentralized Estimation and Control for Multisensor Systems*. CRC Press, 1998.
- [Mye99] D. Myers. An approach to automated programming of industrial robots. In *Proc. 1999 IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, 1999. IEEE-Press.
- [Neh99] U. Nehmzow. *Mobile Robotics: A Practical Introduction*. Springer-Verlag, 1999.
- [Nil94] N. Nilsson. Teleo-reactive programs for agent control. *J. Art. Intell. Res.*, 1:139 – 158, 1994.
- [Nof99] S. Nof. *Handbook of Industrial Robotics*. John Wiley & Sons, 1999.
- [NoR97] United Nations and International Federation of Robotics. *World Industrial Robots*. United Nations, 1997.
- [NPB95] I. Nourbakhsh, R. Powers, and S. Birchfield. DERVISH an office-navigating robot. *AI Magazine*, 16(2), Summer 1995.
- [NYM99] Y. Nakamura, T. Yamazaki, and N. Mizushima. Synthesis, learning and abstraction of skills through parameterized smooth map from sensors and behaviors. In *Proc. 1999 IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, 1999. IEEE-Press.
- [OHB<sup>+</sup>99] D. Ouelhadj, C. Hanachi, B. Bouzouia, A. Moualek, and A. Farhi. A multi-contract net protocol for dynamic scheduling in flexible manufacturing systems (FMS). In *Proc. 1999 IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, 1999. IEEE-Press.
- [Oja83] E. Oja. *Subspace methods of pattern recognition*. Research Studies Press, Hertfordshire, 1983.
- [Pan91] V. Pantelidis. *Robotics in Education: an information guide*. The Scarecrow Press, 1991.
- [Par89] H. Parunak. Manufacturing experience with the contract net. In M. Huhns, editor, *Distributed Artificial Intelligence*. Pitman, 1989.
- [PB94] P. Pook and D. Ballard. Deictic teleassistance. In *Proc. IROS '94 – IEEE/RSJ/GI Int. Conf. on Intell. Robots and Systems*. IEEE Press, 1994.
- [PC96] L. Pryor and G. Collins. Planning for contingencies: A decision-based approach. *J. Art. Intell. Res.*, 4:287 – 339, 1996.
- [PDK98] E. Prassler, R. Dillmann, and H. Kuntze. *Robotik in Deutschland – Lehre, Forschung und Entwicklung*. Shaker-Verlag, 1998.
- [Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., 1988.
- [Pea97] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Academic Press / Morgan Kaufmann, 1997.
- [PL97] A. Pentland and A. Liu. Modeling and prediction of human behavior. In T. Strat, editor, *Proc. of the 1997 Image Understanding Workshop*, New Orleans, 1997. Morgan Kaufmann.

- [PO98] M. Parnichkun and S. Ozono. GSGM movement model for cooperative robots system. *Mechatronics*, 8:905 – 925, 1998.
- [Pom93] D. Pomerlau. *Neural network perception for mobile robot guidance*. Kluwer Academic, 1993.
- [Por88] J. Porrill. Optimal combination and constraints for geometrical sensor data. *Int. J. of Robotics Research*, 7(6), 1988.
- [PS99] Rolf Pfeifer and Christian Scheier. *Understanding intelligence*. MIT Press, 1999.
- [Rap76] Bertram Raphael. *The thinking computer: Mind inside matter*. W.H. Freeman, 1976.
- [RBFT99] N. Roy, W. Burgard, D. Fox, and S. Thrun. Coastal navigation: Mobile robot navigation with uncertainty in dynamic environments. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1999.
- [RDWS93] B. Rao, H. Durrant-Whyte, and J. Sheen. A fully decentralized multi-sensor system for tracking and surveillance. *The Int. J. on Robotics Research*, 12(1):20 – 44, Feb. 1993.
- [Reh99] J. Rehg. *Introduction to Robotics in CIM Systems*. Prentice Hall, 1999.
- [Rev99] S. Reveliotis. Production planning and control in flexibly automated manufacturing systems: Current status and future requirements. In *Proc. 1999 IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, 1999. IEEE-Press.
- [RH97] G. Rickheit and B. Hildebrandt. Der sonderforschungsbereich "situierete künstliche kommunikatoren". *Linguistik. Die Bielefelder Sicht*, pages 145–148, 1997. Bielefeld: Aisthesis Verlag.
- [RHZ99] M. Reinsch, B. Hildebrandt, and J. Zhang. RCRC – Ein flexibles System zur Spracherkennung. Technical report, SFB360, Univ. Bielefeld, 1999.
- [RN95] S.J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.
- [Rot94] Gerhard Roth. *Das Gehirn und seine Wirklichkeit. Kognitive Neurobiologie und ihre philosophischen Konsequenzen*. Suhrkamp, Frankfurt am Main, 1994.
- [RW89] P. Ramadge and W. Wonham. The control of discrete event systems. *Proc. IEEE*, 77(1):81 – 98, 1989.
- [Sac74] E. Sacerdoti. Planning in a hierarchy of abstraction spaces. *Artificial Intelligence*, 5(2):115 – 135, 1974.
- [Sad96] F. Sadjadi. Selected papers on sensor and data fusion. In *SPIE Milestone Series*, volume MS 124. SPIE - Society of Photo-Optical Instrumentation Engineers, 1996.
- [SB96] H. Siegert and S. Bocionek. *Robotik: Programmierung intelligenter Roboter*. Springer-Verlag, 1996.
- [SBF<sup>+</sup>00] D. Schulz, W. Burgard, D. Fox, S. Thrun, and A.B. Cremers. Web interfaces for mobile robots in public places. *IEEE-Magazine on Robotics and Automation*, 2000. To appear.
- [SC94] B. Schiele and J.L. Crowley. A comparison of position estimation techniques using occupancy grids. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1994.
- [Sch97] C. Scheering. Domain-knowledge incorporation for fast colour image segmentation using a pre-clustered chromaticity-plane. In D. Paulus and Th. Wagner, editors, *Tagungsband zum 3. Workshop Farbbildverarbeitung, Erlangen*, pages 51–56, Stuttgart, September 1997. IRB Verlag.
- [SD81] R. Smith and R. Davis. Frameworks for cooperation in a distributed problem solver. *IEEE Trans. on Systems, Man and Cybernetics*, SMC-11, 1981.
- [Sel96] J. Selig. *Geometrical Methods in Robotics*. Springer-Verlag, 1996.
- [SGH<sup>+</sup>97] R. Simmons, R. Goodwin, K. Haigh, S. Koenig, and J. O'Sullivan. A layered architecture for office delivery robots. In *Proc. of the First International Conference on Autonomous Agents*, Marina del Rey, CA, 1997.
- [SGS92] G. Shaffer, J. Gonzalez, and A. Stentz. Comparison of two range-based estimators for a mobile robot. In *SPIE Conf. on Mobile Robots VII*, pages 661–667, 1992.
- [Sha86] M. Shapiro. Structure and encapsulation in distributed systems: The proxy principle. In *6th International Conference on Distributed Computer Systems*, May 1986.

- [She92] T. Sheridan. *Telerobotics, Automation, and Human Supervisory Control*. MIT-Press, Boston, 1992.
- [Sim95] R. Simmons. The 1994 AAAI robot competition and exhibition. *AI Magazine*, 16(2), Summer 1995.
- [Sim96] R. Simmons. The curvature-velocity method for local obstacle avoidance. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1996.
- [SK95] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *Proc. of the International Joint Conference on Artificial Intelligence*, 1995.
- [SK97a] H. Sawhney and R. Kumar. Multi-image alignment. In T. Strat, editor, *Proc. of the 1997 Image Understanding Workshop*, New Orleans, 1997. Morgan Kaufmann.
- [SK97b] C. Scheering and A. Knoll. Fast colour image segmentation using a pre-clustered chromaticity-plane. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP-97), Munich*, volume 4, pages 3145–3147, 4 1997.
- [SK97c] H. Shatkey and L.P. Kaelbling. Learning topological maps with weak local odometric information. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1997.
- [SK98a] C. Scheering and B. Kersting. Uncalibrated hand-eye coordination with a redundant camera system. In *Proc. 1998 IEEE Int. Conf. on Robotics and Automation*, Leuven, Belgium, 1998. IEEE-Press.
- [SK98b] C. Scheering and B. Kersting. Using distributed sensing and sensor fusion for uncalibrated visual manipulator guidance. In *Proc. 4th International Symposium on Distributed Autonomous Robotic Systems (DARS-98)*, pages 267–276, Karlsruhe, May 1998.
- [SK98c] C. Scheering and A. Knoll. A framework for implementing self-organising task-oriented multi-sensor networks. In *Proceedings of SPIE Vol. 3523, Sensor Fusion and Decentralized Control in Robotic Systems*, pages 18–29, Boston, November 1998.
- [SK98d] C. Scheering and A. Knoll. Local failure detection in a redundant camera system for visual manipulator guidance. In *Proceedings of the 1st international Conference on Data Fusion, EuroFusion98*, pages 63–68, Great Malvern, UK, October 1998.
- [Smi81] Reid G. Smith. *Distributed Problem Solving*. UMI Research Press, 1981.
- [Smi92] M. Smith. An environment for more easily programming a robot. In *Proc. 1992 IEEE Int. Conf. on Robotics and Automation*, Nice, 1992. IEEE-Press.
- [Smi95] Tim Smithers. Are autonomous agents information processing systems? In Luc Steels and Rodney Brooks, editors, *The artificial life route to artificial intelligence: building embodied, situated agents*, pages 123–162, Hillsdale, New Jersey, 1995. Lawrence Erlbaum Associates.
- [SNI<sup>+</sup>94] T. Sato, Y. Nishida, Y. Ichikawa, Y. Hatamura, and H. Mizoguchi. Active understanding of human intention by a robot through monitoring of human behavior. In *Proc. IROS '94 – IEEE/RSJ/GI Int. Conf. on Intell. Robots and Systems*. IEEE Press, 1994.
- [ŠO98] P. Švestka and M. Overmars. Coordinated path planning for multiple robots. *Robotics and Autonomous Systems*, 23:125 – 152, 1998.
- [Spi90] C. Spiteri. *Robotics Technology*. Saunders College Publishing, 1990.
- [SS98] R.D. Schraft and G. Schmierer. *Serviceroboter*. Springer-Verlag, 1998. In German.
- [SS99] Eckart Schlottmann and Dirk Spenneberg. Exploration autonomer verhaltensbasierter roboter mit pfadunterstützung. Master's thesis, Universität Bielefeld, 1999.
- [SSC90] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*. Springer Verlag, 1990.
- [SSHC97] Dirk Spenneberg, Eckart Schlottmann, Timo Höpfner, and Thomas Christaller. Pdl programming manual. Arbeitspapiere der GMD 1082, GMD – Forschungszentrum Informationstechnik GmbH, St. Augustin, 1997.
- [Ste81] M. Stefik. Planning with constraints. *Artificial Intelligence*, 16(2):111 – 169, 1981.

- [Ste87] M. Steedman. Combinatory grammars and human language processing. In J. L. Garfield, editor, *Modularity in knowledge representation and natural-language understanding*, pages 187–205. MIT-Press, 1987.
- [Ste93a] M. Steedman. Categorical grammar. *Lingua*, 90:221–258, 1993.
- [Ste93b] W. Richard Stevens. *Unix network programming*. Prentice-Hall, 1993.
- [Ste94a] Luc Steels. The artificial life roots of artificial intelligence. *Artificial Life Journal*, 1(1), 1994.
- [Ste94b] Luc Steels. Building agents with autonomous behavior systems. In Luc Steels and Rodney Brooks, editors, *The artificial life route to artificial intelligence: building situated embodied agents*, New Haven, 1994. Lawrence Erlbaum Associates.
- [Ste96] M. Steedman. *Surface structure and interpretation*. MIT-Press, 1996.
- [Str96] Simone Strippgen. *INSIGHT: Ein virtuelles Labor für Entwurf, Test und Analyse von autonomen Agenten*. PhD thesis, Universität Bielefeld, 12 1996.
- [Sut92] R. S. Sutton. *Reinforcement Learning*. Kluwer Academic Publishers, Boston, MA, 1992.
- [SWR99] D. Schwammkrug, J. Walter, and H. Ritter. Rapid learning of robot grasping position. In *Proc. Int. Symposium on Intelligent Robotics Systems (SIRS-99)*, 1999.
- [SZK99a] C. Scheering, J. Zhang, and A. Knoll. Distributed multiple view fusion for two-arm distance estimation. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, (MFI-99)*, 1999.
- [SZK99b] C. Scheering, J. Zhang, and A. Knoll. Estimating the distance between two robot arms by fusing the views of uncalibrated cameras. In *2nd International Conference on Information Fusion, Fusion 99*, Sunnyvale, CA, USA, 1999.
- [TAT95] K. Tarabanis, P. Allen, and R. Tsai. A survey of sensor planning in computer vision. *IEEE Trans. on Robotics and Automation*, 11(1), 1995.
- [TB96] S. Thrun and A. Bücken. Integrating grid-based and topological maps for mobile robot navigation. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1996.
- [TBB<sup>+</sup>98] S. Thrun, A. Bücken, W. Burgard, D. Fox, T. Fröhlinghaus, D. Hennig, T. Hofmann, M. Krell, and T. Schimdt. Map learning and high-speed navigation in RHINO. In Kortenkamp et al. [KBM98].
- [TBB<sup>+</sup>99] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. MINERVA: A second generation mobile tour-guide robot. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1999.
- [Thr98a] S. Thrun. Bayesian landmark learning for mobile robot localization. *Machine Learning*, 33(1), 1998.
- [Thr98b] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):27–71, 1998.
- [Tod86] D. Todd. *Fundamentals of Robot Technology: An Introduction to Industrial Robots, Teleoperators and Robot Vehicles*. Wiley, 1986.
- [TPJ92] T. Lozano-Pérez and J. Jones. *Handey – a Robot Task Planner*. MIT-Press, 1992.
- [TTA95] K. Tarabanis, R. Tsai, and P. Allen. The MVP sensor planning system for robotic vision tasks. *IEEE Trans. on Robotics and Automation*, 11(1), 1995.
- [TUWR97] E. Trucco, M. Umasuthan, A. Wallace, and V. Roberto. Model-based planning of optimal sensor placements for inspection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(2), 1997.
- [VB97] P. Varshney and C. Burrus, editors. *Distributed Detection and Data Fusion*. Springer-Verlag, 1997.
- [vCSZK99] Y. v. Collani, C. Scheering, J. Zhang, and A. Knoll. A neuro-fuzzy solution for integrated visual and force control. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, (MFI-99)*, 1999.
- [vCZK97] Y. von Collani, J. Zhang, and A. Knoll. A neuro-fuzzy solution for fine-motion control based on vision and force sensors. Technical report, Universität Bielefeld, 1997.

- [vCZK98] Y. von Collani, J. Zhang, and A. Knoll. A neuro-fuzzy solution fine-motion control based on vision and force sensors. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2965–2970, Leuven, Belgium, 1998.
- [VK99] R. Voyles and P. Khosla. Gesture-based programming: A preliminary demonstration. In *Proc. 1999 IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, 1999. IEEE-Press.
- [Vol92] J. Volmer. *Industrieroboter*. Verlag Technik, 1992.
- [vR97] G. von Randow. *Roboter: Unsere Nächsten Verwandten*. Rowohlt Verlag, Reinbek, Germany, 1997.
- [Web93a] Babara Webb. Modeling biological behavior or 'dumb animals and stupid robots'. In *Proceedings of the 2nd European Conference on Artificial Life*, pages 1002–1090, 1993.
- [Web93b] Babara Webb. *Perception in real and artificial insects: A robotic investigation of cricket phonotaxis*. PhD thesis, University of Edinburgh, 1993.
- [Wel94] D. Weld. An introduction to least commitment planning. *AI Magazine*, 15(4):27 – 61, 1994.
- [WI95] M. Wheeler and K. Ikeuchi. Sensor modeling, probabilistic hypothesis generation, and robust localization for object recognition. *IEEE Trans on Patt. Analysis and Mach. Intell.*, 17, No. 3, March 1995.
- [Win72] Terry Winograd. *Understanding natural language*. Edinburgh University Press, 1972.
- [Wis99] E. Wise. *Applied Robotics*. Howard W. Sams & Co, 1999.
- [WL90] E. Waltz and J. Llinas. *Multisensor Data Fusion*. Artech House, 1990.
- [XY95] Y. Xu and J. Yang. Towards human-robot coordination: Skill-modelling and transferring via hidden-markov-model. In *Proc. IEEE Conference on Robotics and Automation*. IEEE Comp. Soc. Press, 1995.
- [YG99] X. Yuan and Y. Gu. An integration of robot programming and sequence planning. In *Proc. of the 1999 IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, 1999. IEEE-Press.
- [YMC98] C. Yang, M. Marefat, and F. Ciarallo. Error analysis and planning accuracy for dimensional measurement in active vision inspection. *IEEE Trans. on Robotics and Automation*, 14(3), 1998.
- [YUDS99] Y. Yamada, Y. Umetani, H. Daitoh, and T. Sakai. Construction of a human/robot coexistence system based on a model of human will – intention and desire. In *Proc. 1999 IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, 1999. IEEE-Press.
- [YX96] W. Yun and Y. Xi. Optimum motion planning in joint space for robots using genetic algorithms. *Robotics and Autonomous Systems*, 18:373 – 393, 1996.
- [YXC94] J. Yang, Y. Xu, and C. Chen. Hidden markov model approach to skill learning and its application to telerobotics. *IEEE Trans. on Robotics and Automation*, 10, No. 5, Oct. 1994.
- [YXC97] J. Yang, Y. Xu, and C. Chen. Human action learning via hidden markov model. *IEEE Trans. on Systems, Man, and Cybernetics (A)*, 27(1):34 – 44, 1997.
- [Zel98] A. Zelinsky. *Field and Service Robotics*. Springer Verlag, 1998.
- [ZF98] J. Zhang and M. Ferch. Rapid on-line learning of compliant motion for two-arm coordination. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1853–1858, 1998.
- [Zha95] J. Zhang. *Ein Integriertes Verfahren zur effizienten Planung und Ausführung von Roboterbewegungen in unscharfen Umgebungen*. Infix-Verlag, 1995.
- [Zha99a] X. Zha. Discrete event control with active events. In *Proc. 1999 IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, 1999. IEEE-Press.
- [Zha99b] X. Zha. An intelligent approach and system for rapid robotic assembly prototyping, planning and control. In *Proc. 1999 IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, 1999. IEEE-Press.
- [ZK99] J. Zhang and A. Knoll. Integrating deliberative and reactive strategies via fuzzy modular control. In A. Saffiotti and D. Driankov, editors, *Fuzzy logic techniques for autonomous vehicle navigation*. Springer, 1999.

- [ZKS99] J. Zhang, A. Knoll, and R. Schmidt. A neuro-fuzzy control model for fine-positioning of manipulators. *Journal of Robotics and Autonomous Systems*, 1999.
- [ZSK98] J. Zhang, V. Schwert, and A. Knoll. Visuelles Lernen zur Lokalisierung eines mobilen Roboters. In *Ta- gungsband Autonome Mobile Robotersysteme*, Karlsruhe, 1998.
- [ZSK99] J. Zhang, R. Schmidt, and A. Knoll. Appearance-based visual learning in a neuro-fuzzy model for fine- positioning of manipulators. In *Proc. 1999 IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, 1999. IEEE-Press.
- [ZvCK97] J. Zhang, Y. v. Collani, and A. Knoll. B-spline fuzzy controller to acquire sensor-based assembly skills. In *Proc. 1997 IEEE Int. Conf. on Robotics and Automation*, Albuquerque, NM, USA, 1997. IEEE-Press.